COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

11/2018 VOL.61 NO.11

Special Section on China Region

A Look at the Design of Lua

AI, Explain Yourself

Software Challenges for the Changing Storage Landscape

Association for Computing Machinery

VEE 2019

15th ACM SIGPLAN/SIGOPS international conference on

Virtual Execution Environments

Providence, RI   April 13-14, 2019   with ASPLOS

Authors are invited to submit original papers related to virtualization across all layers of the software stack, from high-level language virtual machines down to the micro-architectural level. VEE 2019 accepts both full-length and short papers.

Abstract deadline: December 7, 2018

General Chair
Jennifer Sartor
(Vrije Universiteit Brussel and Ghent University)

Program Co-chairs
Christopher J. Rossbach
(UT Austin and VMware Research)
Mayur Naik
(University of Pennsylvania)

acm

in cooperation with

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

http://conf.researchr.org/home/vee-2019

The ACM International Conference on
Interactive Experiences for Television and Online Video

tvx2019

Manchester, UK 5th-7th June 2019

Bringing together researchers and practitioners to explore the design, engineering and human experience of future online, interactive and immersive video content.

Important Dates

16th November 2018
Workshop Proposals

18th January 2019
Full & Short Papers (abstract, title & metadata)

25th January 2019
Full & Short Papers (submission of manuscript)

22nd March 2019
TVX-in-Industry, Demos, Work-in-Progress, Doctoral Consortium

- -

1st November 2018
SIGCHI Student Travel Grant

15th November 2018
SIGCHI Gary Marsden Student Development Fund

Key Topics

Artificial Intelligence // Big Data // Social Computing // Immersive Experiences // Virtual Reality // Mixed Reality // UX // Interaction Design // Content Production // Systems and Infrastructures // Devices // Interaction Techniques // Media Studies // Business Models // Marketing // Innovative Concepts // Media Art // Object Based Media // etc...

tvx.acm.org/2019

BBC   UNIVERSITY of York   SIGCHI   acm Association for Computing Machinery

COMMUNICATIONS OF THE ACM

Departments

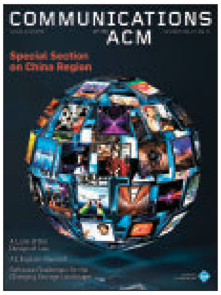
Last Byte

News

Viewpoints

Special Section: China Region

Watch the co-organizers discuss this section in the exclusive *Communications* video. https://cacm.acm.org/videos/china-region



About the Cover:
A spherical mosaic of some of the images, icons, and technologies depicted in this issue's special section on the China Region. Cover illustration by Spooky Pooka at Debut Art.

IMAGES IN COVER COLLAGE: Naomi Wu photo courtesy of Naomi Wu/Wikimedia CC-BY-SA-4.0. AliExpress photo by Piotr Swat/Shutterstock.com; PUBG player photo by Hafiez Razali/Shutterstock.com; subway photo by Chutharat Kamkhuntee/Shutterstock.com; Xiaomi store photo by THINK A/Shutterstock.com; robot waiters photo by Inspired By Maps/Shutterstock.com; smart-bike photo by tomocz/Shutterstock.com; Lei Jun photo by zhangjin_net/Shutterstock.com; Alibaba app photo by Jonathan Weiss/Shutterstock.com. Additional stock images from Shutterstock.com.

ILLUSTRATION BY SPOOKY POOKA AT DEBUT ART

11/2018
VOL. 61 NO. 11

Practice

Articles' development led by acmqueue
queue.acm.org

Contributed Articles

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/a-look-at-the-design-of-lua

Review Articles

Research Highlights

Association for Computing Machinery
Advancing Computing as a Science & Profession

(L) PHOTO BY VERSHININ89; (R) IMAGERY BY BOBNEVV AND ART-SONIK

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

CONTACT POINTS
Copyright permission
permissions@hq.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmhelp@acm.org
Letters to the Editor
letters@cacm.acm.org

WEBSITE
http://cacm.acm.org

AUTHOR GUIDELINES
http://cacm.acm.org/about-communications/author-center

ACM ADVERTISING DEPARTMENT
2 Penn Plaza, Suite 701, New York, NY 10121-0701
T (212) 626-0686
F (212) 869-0481

Advertising Sales Account Manager
Ilia Rodriguez
ilia.rodriguez@hq.acm.org

Media Kit acmmediasales@acm.org

Association for Computing Machinery (ACM)
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (212) 869-7440; F (212) 869-0481

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $269.

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current advertising rates can be found by visiting http://www.acm-media.org or by contacting ACM Media Sales at (212) 626-0686.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM
(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER
Please send address changes to
Communications of the ACM
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA

Printed in the USA.

Association for Computing Machinery

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/3281164

Vinton G. Cerf

The Upper Layers of the Internet

The Internet is a layered infrastructure and much has been made of the utility of that layering. IP layer packets are unaware of their underlying transport mechanisms.

The Internet Protocol (IP) layer does not know or care what it carries in its payloads except that they are made up of binary bits. The layer above IP, such as the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Real-Time Protocol (RTP), are equally unaware of the meaning of the bits they carry although they treat these bits differently: TCP keeps packets in order and repairs loss by retransmission. It also filters out duplicates and controls the flow to deal with congestion. The UDP transports packets quickly but without concern for ordering or recovery from loss. RTP uses time stamping and payload typing to cope with jitter and with correct interpretation of the format of the carried bits. Payloads can be marked as audio encoded (for example, G.711, G.723) or video encoded (for example, H.261, MPEG-4), although the meaning of these encoded payloads is opaque to RTP.

Electronic mail is carried in the Internet as Simple Mail Transport Protocol (SMTP). SMTP knows about the format of email messages but not about the meaning of the content. The World Wide Web uses the HyperText Transport Protocol (HTTP) over TCP to carry Web-page content encoded in HyperText Markup Language (HTML). Huge numbers of applications sit on top of HTTP providing useful information or functionality in the form of smartphone or other computing platform applications.

Above these application layers, however, the meaning of the content becomes important. The content of email, Web pages, video and audio streams is interpreted and used by humans. The transporting protocols and the applications that render content for human consumption are unaware of this meaning. One can imagine, however, a virtual layer above the application layer that deals with content. We might think of this as a political layer, not in the partisan sense, but in the sense that the content is viewed through a policy lens applied by some agreed-upon authority. The operators of application platforms may choose to enforce terms and conditions of use (for example, content inciting violence is not allowed, no one below a certain age is permitted to use the application or consume the content) or may be required by legal means to enforce such restrictions.

Policy is a means of expressing behavioral norms that may also be enforced through law. Policy is a way to provide positive incentive for desired behavior or negative incentive for undesired behavior. In the Internet space, so-called platform providers may self-regulate or may be regulated by external government agencies with the means to compel desired behaviors. The Internet's applications and the providers who support them form a kind of *smart medium* in which content is important. In some regimes, these platform providers are treated as common-carrier-like entities that are not responsible for content while those producing the content are held to enforceable norms. In other regimes, the smart media providers are required to become policers of content at risk of legal sanctions. Incentives for content providers may even be structured to lead to self-censorship to avoid penalties.

The choices made by governments as to the rules and means by which content is controlled have a profound effect on the nature of the society thus produced. Among the rights expressed in the Universal Declaration of Human Rights is freedom of speech. That freedom may well be abridged through incentives for platform providers either to restrict that freedom or to reveal the identity of the speaker, who then faces the consequences of prohibited speech. In the context of the World Wide Web, in some countries, search engine providers have been required to remove from their indices references to content deemed unacceptable by government authorities. On the other hand, there are endless examples of malware and misinformation that can be seen as harmful speech that threatens the safety and security of citizens and should be filtered and removed.

The permissive openness of the lower layers of the Internet is now plainly challenged by the policy choices made at the virtual political layer. Where lines are drawn in permission space will profoundly affect the utility and ubiquity of the Internet and the society in which it is embedded. The tussle in cyberspace, so aptly named by David D. Clark et al.,[a] is far from over. C

[a] https://goo.gl/rLYqU8

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Association for
Computing Machinery

ACM Chuck Thacker Breakthrough in Computing Award

The "ACM Breakthrough Award"

Nominations Solicited

Nominations are invited for the inaugural 2018 ***ACM Charles P. "Chuck" Thacker Breakthrough in Computing Award*** (the "ACM Breakthrough Award").

ACM Turing Laureate Charles P. (Chuck) Thacker (1943–2017) received the 2009 ACM A.M. Turing Award for "the pioneering design and realization of the first modern personal computer—the Alto at Xerox PARC—and seminal inventions and contributions to local area networks (including the Ethernet), multiprocessor workstations, snooping cache coherence protocols, and tablet personal computers."

The award was established in recognition of Thacker's pioneering contributions in computing.

These contributions are considered by the community to have propelled the world in the early 1970s from a visionary idea to the reality of modern personal computing, providing people with an early glimpse of how computing would deeply influence us all. The award also celebrates Thacker's long-term inspirational mentorship of generations of computer scientists.

The Breakthrough Award will recognize individuals with the same out-of-the-box thinking and "can-do" approach to solving the unsolved that Thacker exhibited. The recipient should be someone who has made a surprising or disruptive leapfrog in computing ideas or technologies that provides a new capability or understanding that influences the course of computing technologies in a deep and significant manner through its numerous downstream influences and outcomes. Due to the breakthrough nature of the award it is expected that it will be presented biennially and will not be presented if there is no candidate who meets the criteria in a particular year.

The award is accompanied by a prize of $100,000 and would be presented at the annual ACM Awards Banquet. The award recipient would be expected to give the *ACM Breakthrough Lecture* at a major ACM conference of his or her choice during the year following the announcement. The travel expenses of the recipient, and a companion, to attend the Lecture are supported by the award. Financial support of the Thacker Award is provided by Microsoft.

Nomination information and the online submission form are available on:
https://awards.acm.org/thacker/nominations

The deadline for nominations/endorsements is:
January 15, 2019, End of Day, AoE, UTC-12 hours.

For additional information on ACM's award program please visit:
www.acm.org/awards/

DOI:10.1145/3279813

Moshe Y. Vardi

Self-Reference and Section 230

One of the most amazing features of human languages is their capacity for self-reference. The consequences of this feature were explored by Eubulides, a 4th-century BCE Greek philosopher, who formulated the Liar's Paradox, "What I am saying now is a lie." Is this a lie or not? For over 2,000 years, the Liar's Paradox was a philosophical oddity. In 1902, in a letter to the mathematician Friedrich Frege, the philosopher Bertrand Russell reformulated the Liar's Paradox as a paradox in set theory, arguing that "the collection of all sets that do not include themselves as members" both cannot be a set and cannot fail to be a set. By identifying a contradiction in set theory, Russell launched a "foundational crisis" in mathematics.

The foundational crisis proved to be enormously fertile. It inspired David Hilbert to launch in the early 1920s what has become known as "Hilbert's Program," the goal of which was to demonstrate that mathematics was consistent (free of paradoxes), complete (can answer all mathematical questions), and decidable (amenable to computation). Within 15 years, Hilbert's Program was demolished first by Kurt Gödel, who proved that arithmetic is incomplete and cannot prove its own consistency, and then by Church and Turing, who showed that First-Order Logic is undecidable. The crucial technique used by Gödel, Church, and Turing is that of *diagonalization*, which is a technical term for self-reference. Turing invented Turing Machines as a model for computability in order to prove his undecidability result, so theoretical computer science rose out of the ashes of Hilbert's Program. Diagonalization went on to play a key role in computational complexity theory, where it was used to prove separation results, for example, that there are problems that can be solved in exponential time, but not in polynomial time.

Yet mathematicians, in general, took the demise of Hilbert's Program in stride. Mathematics is incomplete, undecidable, and cannot prove its own consistency; so what? Mathematics has just gone on. Perhaps computer scientists should develop a similar nonchalant attitude about negative results. The undecidability of program termination means there is no algorithm that can correctly decide termination of *all* programs. So what? As I argued in "Solving the Unsolvable,"[a] program termination may be practically decidable, even though it is theoretically undecidable. Just as mathematicians gave up on the quest to find a proof system that can prove *all* true mathematical statements, we may need to give up on the quest for algorithms that solve *all* problem instances. In other words, the quest for *universality* is self-destructive.

Self-reference was taken in a different direction by the philosopher Karl Popper, who formulated the *paradox of freedom* in 1945: "The so-called paradox of freedom is the argument that freedom in the sense of absence of any constraining control must lead to very great restraint, since it makes the bully free to enslave the meek." Closely related is the *paradox of tolerance*: "Unlimited tolerance must lead to the disappearance of tolerance." Popper's conclusion was that we must give up on the universality of freedom and tolerance, as complete freedom and tolerance are self-destructive. Even a free society must have some limits on freedom, and a tolerant society must be intolerant of intolerance.

These philosophical musings from more than 70 years ago seem these days to be quite prescient and relevant. Section 230 of the Communications Decency Act of 1996 is a fundamental item of Internet legislation in the U.S. Section 230 provides immunity from liability for providers and users of an "interactive computer service" who publish information provided by third-party users, asserting: "No provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider." Section 230 enables Internet companies such as Google and Facebook to be considered as platforms rather than as publishers, free from liability for the content they publish. The explosive growth of social-media platforms would have not been possible without Section 230.

Yet this explosive growth has led to the proliferation of "bad" speech on social-media platforms, which has become politically untenable. All social media platforms are now actively fighting "fake news"—false news stories typically spread with the intent to influence political views. Recently, social-media platforms have banned the conspiracy theorist Alex Jones for violating their "abusive behavior" policy. In spite of Section 230, social-media platforms seem to be accepting responsibility for the content they publish. In other words, they are starting to behave with some restraint, like publishers, rather than platforms. Popper would be pleased with this development!

Follow me on Facebook, Google+, and Twitter. C

Moshe Y. Vardi (vardi@cs.rice.edu) is the Karen Ostrum George Distinguished Service Professor in Computational Engineering and Director of the Ken Kennedy Institute for Information Technology at Rice University, Houston, TX, USA. He is the former Editor-in-Chief of *Communications*.

[a] https://bit.ly/2qzssiv

BLOG@CACM

The *Communications* Web site, http://cacm.acm.org, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter

Follow us on Twitter at http://twitter.com/blogCACM

DOI:10.1145/3276740 http://cacm.acm.org/blogs/blog-cacm

The Gap in CS, Mulling Irrational Exuberance

Carl Hewitt suggests computer science needs a reference resource, while Vijay Kumar decries intellectual dishonesty in technology forecasting.

Carl Hewitt
Computer Science Encyclopedia Can Fill a Gap

https://cacm.acm.org/blogs/blog-cacm/230860-computer-science-encyclopedia-can-fill-a-gap/fulltext
September 5, 2018

There is an important gap in computer science (CS) education and professional collaboration that can be filled by a nonprofit online reputable, referenceable encyclopedia supported by appropriate professionally relevant advertising. The encyclopedia should be managed by a prestigious editorial board that appoints a hierarchy of editors to moderate articles. The editorial board can guarantee editorial independence from advertisers analogous to current professional practices for journals and conferences. Anyone would be allowed to register to submit suggestions and drafts to the editors. Access to articles would be free and available to all. The encyclopedia must establish procedures to be fair and inclusive on the basis of race, sex, religion, age, disability, and national origin integrating content suitable for all from preschool to advanced researchers.

The encyclopedia could support interactive articles with videos, animations, and dynamic narrations. Within a decade, interactive content could be a requirement for most articles. Over time, the encyclopedia should be organized using ontological services supporting programmatic interfaces for a knowledge graph.

The encyclopedia should become a standard reference, a trustworthy professionally accountable educational resource for all. Currently, there is no online encyclopedia that can serve as the source of valid scientific references.

Our profession has the credibility and resources to create an encyclopedia to serve as the professional standard. Serving as a member of its editorial board could be a prestigious office for senior professionals to provide experience and judgment. Professional reputations could be enhanced by contributing to the encyclopedia because contributions would be publicly announced. The encyclopedia could knit together our profession in an important way, while fundamentally improving education and professional relationships in CS.

A nonprofit professional encyclopedia would be self-supporting through appropriate professionally relevant advertising carefully curated for high standards using existing advertising programs.

Vijay Kumar
Irrational Exuberance and the 'FATE' of Technology

https://cacm.acm.org/blogs/blog-cacm/230472-irrational-exuberance-and-the-fate-of-technology/fulltext
August 20, 2018

I am sure many of us remember the Netscape IPO in 1995 and the fivefold growth in share value in four months. Expectations for technology and its impact were in the stratosphere. The Federal Reserve Board's then-chairman, Alan Greenspan, gave a speech at the American Enterprise Institute questioning "irrational exuberance" in the market and in technology.[1] I believe today we are seeing similar exuberance with technology.

Are revolutionary technologies for cancer screening—that rely on a finger-prick drawing one-thousandth the normal amount of blood—really feasible? Theranos had everyone believe such a revolutionary advancement was possible[2] not because of new techniques in analytical chemistry, but because it had developed novel software and new automation technologies! Can we really hope to replace eight million cars in Los Angeles by boring tunnels[3] for high-speed pods that will travel at 150 MPH at $1 per ride? This is what Boring Company is selling the City of Los Angeles. Do recent advances in data science and machine learning really mean artificial general intelligence is around the corner? This is the pitch of so many startups today.

There have been advances in statistical machine learning, which have had remarkable impact in fields like computer vision and speech recognition when the underlying neural networks are trained by large-enough representative datasets. What "large enough" means, we don't yet know. Neither do we know when we have a representative dataset. Yet there are many interesting cases where deep learning "works." These success stories are oversold. In my own field—robotics—autonomy is a challenging problem, especially in tasks of manipulation and perception-action loops. Yet despite the claims being made, our best robots lack the dexterity of a three-year-old child.

Nowhere is irrational exuberance more evident than in self-driving cars. Not many people know the first demonstrations of an autonomous car were in the late 1980s at the Bundeswehr University Munich and at Carnegie Mellon University. Autonomous vehicles can have a tremendous social, economic, and environmental impact. This fact, and the technical challenges in realizing a bold vision, has attracted some of the top talent in science and engineering over the last 30 years. However, many of us don't remember history, and many choose to ignore it since problems known to have not been solved for three decades are unlikely to attract investment.

According to recent predictions,[4] fully autonomous cars will be available soon. Fully autonomous Audis and Teslas were promised several years ago by 2018. Uber even promised us flying cars powered by clean energy by 2023, even though the basic physics and chemistry underlying battery technology tells us otherwise.[5]

It is worrisome when engineers make these claims, and even more so when entrepreneurs use such claims to raise funding. However, the biggest concern should be about embedding software for autonomy in safety-critical systems. There is a difference between running tests and logging data, and verification of software guaranteed not to have unwanted, unsafe behaviors. Can we claim vehicles are safe because the underlying software has been tested with over a billion miles of data? U.S. National Safety Council statistics suggest a billion miles of human driving, on average, results in 12.5 fatalities,[6] and a billion-mile dataset cannot possibly be viewed as large enough or representative enough to train software to prevent fatalities.

The Uber-Waymo trial led to the release of documents truly shocking in this regard. They reveal a culture[7] that appears to prioritize releasing the latest software over testing and verification, and one that encourages shortcuts. This may be acceptable for a buggy operating system for a phone that can be patched later, but should be unacceptable for software that drives a car.

This irrational exuberance may have its roots in the exponential growth in computing and storage technologies predicted by Gordon Moore five decades ago. The fact that just over a decade ago smartphones, cloud computing, and ride-sharing seemed like science fiction, and technologies like 3D printing and DNA sequencing were prohibitively expensive, has led to a culture of extrapolation fueled by exponential growth. Advances in creating programs that can play board games like chess and recent results with Alpha Go and Alpha Zero have been mind-boggling. Unfortunately, from this comes the extrapolation that it is only a question of time before we conquer general intelligence.

There is at least one argument that we are not making significant progress in understanding intelligence if we take into account the exponential growth in computing due to Moore's Law. While computers have achieved superhuman performance in chess, the Elo rating of chess programs has merely increased linearly over the last three decades.[8] If we were able to exploit the benefits of Moore's Law, our chess-playing programs should be a billion times better than the programs from 30 years ago, instead of merely 30 times better. This suggests the exponential growth of technology may not even apply to algorithmic advances in artificial intelligence,[9] let alone to advances in energy storage, biotechnology, automation, and manufacturing.

Irrational exuberance in technology has led to an even bigger problem: intellectual dishonesty, which every engineer and computer scientist must guard against. As professionals, it is our responsibility to call out intellectual dishonesty.

Questions of verification, safety, and trust must be central when we embody intelligence in physical systems. Questions of fairness, accountability, transparency, and ethics (FATE) should be addressed for data and information in society. It is great to see such efforts taking shape in industry[10] and academia.[11]

As teachers, we have an even bigger responsibility, as technology is no longer taught to just computer scientists or engineers; it is now a new liberal art. It is critical to address the true limitations of what technology can really bring about in the imminent future and the real dangers of extrapolation. Every university student who designs or creates anything must be sensitized to fundamental concerns of accountability and transparency and ethical responsibilities. We must address the FATE of technology, across all activities of design, synthesis and reduction of technologies to practice.

References

1. https://www.federalreserve.gov/boarddocs/speeches/1996/19961205.htm
2. https://www.newyorker.com/magazine/2014/12/15/blood-simpler
3. https://www.cnbc.com/2018/05/18/elon-musk-promises-1-rides-for-boring-companys-la-tunnels.html
4. http://www.driverless-future.com/?page_id=384
5. https://qz.com/1243334/the-magical-battery-uber-needs-for-its-flying-cars/
6. https://www.forbes.com/sites/jimgorzelany/2017/02/16/death-race-2017-where-to-find-the-most-dangerous-roads-in-america/#68a5354a1324
7. https://www.theverge.com/2018/3/20/17144090/uber-car-accident-arizona-safety-anthony-levandowski-waymo
8. https://www.eff.org/ai/metrics
9. https://freedom-to-tinker.com/2018/01/04/singularity-skepticism-2-why-self-improvement-isnt-enough/
10. https://www.microsoft.com/en-us/research/group/fate/
11. http://warrencenter.upenn.edu/

Carl Hewitt is an emeritus professor of the Massachusetts Institute of Technology, USA. He is board chair of iRobust, a scientific society for the promotion of the field of Inconsistency Robustness, and board chair of Standard IoT, an international standards organization for the Internet of Things. **Vijay Kumar** is Nemirovsky Family Dean of Penn Engineering with appointments in the departments of Mechanical Engineering and Applied Mechanics, Computer and Information Science, and Electrical and Systems Engineering at the University of Pennsylvania, USA.

Inviting Young Scientists

HEIDELBERG LAUREATE FORUM

acm Association for Computing Machinery

Meet Great Minds in Computer Science and Mathematics

As one of the founding organizations of the Heidelberg Laureate Forum **http://www.heidelberg-laureate-forum.org/**, ACM invites young computer science and mathematics researchers to meet some of the preeminent scientists in their field. These may be the very pioneering researchers who sparked your passion for research in computer science and/or mathematics.

These laureates include recipients of the ACM A.M. Turing Award, the Abel Prize, the Fields Medal, and the Nevanlinna Prize.

The 7th Heidelberg Laureate Forum will take place **September 22–27, 2019** in Heidelberg, Germany.

This week-long event features presentations, workshops, panel discussions, and social events focusing on scientific inspiration and exchange among laureates and young scientists.

Who can participate?

New and recent Ph.Ds, doctoral candidates, other graduate students pursuing research, and undergraduate students with solid research experience and a commitment to computing research

How to apply:

Online: **https://application.heidelberg-laureate-forum.org/**
Materials to complete applications are listed on the site.

What is the schedule?

The application deadline is **February 15, 2019.**

We reserve the right to close the application website early depending on the volume

Successful applicants will be notified by **mid April 2019**.

More information available on Heidelberg social media

PHOTOS: ©HLFF / B. Kreutzer (top);
©HLFF / C. Flemming (bottom)

news

Science | DOI:10.1145/3276742

Don Monroe

AI, Explain Yourself

It is increasingly important to understand how artificial intelligence comes to a decision.

IMAGE BY LETTERS-SHMETTERS

ARTIFICIAL INTELLIGENCE (AI) SYSTEMS are taking over a vast array of tasks that previously depended on human expertise and judgment. Often, however, the "reasoning" behind their actions is unclear, and can produce surprising errors or reinforce biased processes. One way to address this issue is to make AI "explainable" to humans—for example, designers who can improve it or let users better know when to trust it. Although the best styles of explanation for different purposes are still being studied, they will profoundly shape how future AI is used.

Some explainable AI, or XAI, has long been familiar, as part of online recommender systems: book purchasers or movie viewers see suggestions for additional selections described as having certain similar attributes, or being chosen by similar users. The stakes are low, however, and occasional misfires are easily ignored, with or without these explanations.

Nonetheless, the choices made by these and other AI systems sometimes defy common sense, showing our faith in them is often an unjustified projection of our own thinking. "The implicit notion that AI somehow is another form of consciousness is very disturbing to me," said Ben Shneiderman, a Distinguished University Professor in the department of computer science and founding director of the Human-Computer Interaction Laboratory at the University of Maryland.

As AI is applied more broadly, it will be critical to understand how it reaches its conclusions, sometimes for specific cases and sometimes as a general principle. At the individual level, designers have both ethical and legal responsibilities to provide such justification for decisions that could result in death, financial loss, or denial of parole. A reform of European Union data protection tools that took effect in May highlighted these responsibilities, although they refer only indirectly to a "right to explanation." Still, any required explanations will not help much if they resemble the unread fine print of software end-user agreements. "It must be explainable to *people*," Shneiderman said, including people who are not expert in AI.

For designers, providing explanations of surprising decisions need not be just an extra headache, but it "is going to be a very virtuous thing for AI," Shneiderman stressed. "If you have an explainable algorithm, you're more likely to have an effective one," he asserted.

Explainable methods have not always performed better, though. For example, early AI comprised large sets of rules motivated by human decision criteria, and was therefore easy to understand within a restricted domain, but its capability was often disappointing. In contrast, recent dramatic performance improvements in AI are based on deep learning using huge neural networks with many hidden features that are "programmed" by exposure to huge numbers of examples. These systems apply vast computer power to these annotated training datasets to discern patterns that are often beyond what humans can recognize.

What Is an Explanation?

Considering this internal complexity of modern AI, it may seem unreasonable to hope for a human-scale explanation. For a deep learning system trained on thousands of pictures of cats and not-cats, "Maybe the best analogy is that it develops a gut instinct for what is a cat and what isn't," said Ernest Davis, a professor of computer science at New York University. Just as people devise post-hoc rationalizations for their own decisions, such as pointy ears, a tail, and so forth, "that doesn't actually explain why you recognized it as a cat," he said. "Generating that kind of account is a different task."

An important challenge is that such independently generated explanations could also be chosen for their intuitive plausibility, rather than their accuracy. Presenting favorable stories will be particularly tempting when legal liability is at stake—for example, when a self-driving car kills a pedestrian, or an AI system participates in a medical mistake.

Liability assessment requires a detailed audit trail, Shneiderman said, analogous to the flight-data recorders that allow the U.S. National Transportation Safety Board to retroactively study airplane crashes. This kind of "explanation" allows a regulatory oversight agency to analyze a failure, assign penalties, and require modifications to prevent a recurring failure. "My legal friends tell me that the law is perfectly fine," Shneiderman said. "We don't need new laws to deal with AI."

Explaining individual incidents is hard enough, but in other cases problems may only emerge in a system's aggregate performance. For example, AI programs used to assess borrowers' creditworthiness or criminals' recidivism based on socioeconomic attributes may end up discriminating against individuals whose racial cohort tends to have unfavorable characteristics. Similarly, systems analyzing medical records "might pick up something that looks like race as an important indicator for some outcome," when actually patients of different races just end up in hospitals that use different procedures, said Finale Doshi-Velez, an assistant professor of computer science in the John A. Paulson School of Engineering and Applied Sciences at Harvard University.

Many end users, however, seek less legalistic explanations that may not be provably connected to the underlying program. Like AI, "People are incredibly complicated in terms of how we think and make decisions," said Doshi-Velez, but "we are able to explain things to each other."

In medical use, for example, it can be enough to have an explanation that clarifies the diagnostic or therapeutic decision for a subset of patients with similar conditions, Doshi-Velez said. Such a "local" explanation need not address all the complexities and outliers covered by the full-blown deep learning system.

Considering the internal complexity of modern AI, it may seem unreasonable to hope for a human-scale explanation of its decision-making rationale.

"Depending on your application, you might think of different formats of explanation," agreed Regina Barzilay, Delta Electronics Professor in the department of electrical engineering and computer science at the Massachusetts Institute of Technology. At one level, for example, the system can explain by "identifying excerpts from the input which drove the decision," as her group is doing for molecular modeling. Another technique is "to find which instances in the training set are the closest" to the target.

Appropriate Trust

In view of AI's growing military importance, the U.S. Defense Advanced Research Projects Agency (DARPA) in 2017 rolled out an ambitious program to explore XAI from many different perspectives and compare them. "One of the main goals or benefits of the explanation would be appropriate trust," stressed David Gunning, the program's manager. "What you really need is for people to have a more fine-tuned model of what the system is doing so they know the cases when they can trust it and when they shouldn't trust it."

Most of the dozen projects aim to incorporate explanation-friendly features into deep learning systems; for example, preprogramming the internal network structure to favor familiar concepts.

A critical issue is whether explainable features degrade the performance of the AI. "I think there is inherent trade-off," Gunning said, although he notes that some participants disagree. Barzilay, in contrast, says that experiments so far indicate any performance hit from making an AI explainable is "really, really minimal."

As an alternative to modifying deep learning, one of the DARPA projects replaces it with an approach inherently easier to interpret. The challenge in that case is to make its performance more competitive, Gunning said.

A third strategy is to use a separate system to describe the learning system, which is treated as a black box, essentially using one learning system to analyze another. For this scheme, one question is whether the explanation accurately describes the original system.

As the results come in, beginning in fall 2018, "the program should produce a portfolio of techniques," Gunning said. An important feature of the program is a formal evaluation, in which the usefulness to human users of results with or without explanation will be compared. Some of this assessment will based on subjective impression, but users will also try to predict, for example, whether the system will correctly execute a new task.

The goal, Gunning said, is to determine whether "the explanation gives them a better idea of the system's strengths and weaknesses."

Ensuring Human Control

Ultimately, explanations must be understandable by humans who are not AI experts. The challenges of doing this and measuring the results are familiar to educators worldwide, and successful approaches must include not only computer science, but also psychology.

"This human-computer interaction is becoming more and more important," for example for medical AI systems, said Andreas Holzinger, lead of the Holzinger Group at the Institute for Medical Informatics/Statistics of the Medical University of Graz, Austria, as well as an associate professor of applied computer science at the Graz University of Technology. "The most pressing question is what is interesting and what is relevant" to make the explanation useful in diagnosis and treatment. "We want to augment human intelligence," Holzinger said. "Let the human do what the human can do well, and so for the computer."

For scientific systems, users "are thinking about mechanistic explanations," Barzilay said. "The potential is to have a symbiosis between machines and humans. If these patterns are provided to humans, can they really do better science?" she said. "I think this will be the next frontier."

Instead of teamwork between AI and humans, however, Shneiderman regards the more appropriate goal as leveraging human decision making, rather than outsourcing it. "The key word is responsibility," he said. "When we're doing medical, or legal, or parole, or loans, or hiring, or firing, or so on, these are consequential.

"It's time for AI to move out of its adolescent, game-playing phase and take seriously the notions of quality and reliability,"says Shneiderman. C

"It's time for AI to move out its adolescent, game-playing phase and take seriously the notions of quality and reliability."

Further Reading

Statement on Algorithmic Transparency and Accountability,
Association for Computing Machinery US Public Policy Council, Jan. 12, 2017,
https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf.

European Commission
2018 reform of EU data protection rules
https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., Schieber, S., Walso, J., Weinberg, D., and Wood, A.
Accountability of AI Under the Law: The Role of Explanation,
https://arxiv.org/abs/1711.01134

Fairness, Accountability, and Transparency in Machine Learning Group
https://www.fatml.org/

Explainable Artificial Intelligence (XAI Project)
U.S. Defense Advanced Research Projects Agency
https://www.darpa.mil/program/explainable-artificial-intelligence

DARPA Perspective on AI
U.S. Defense Advanced Research Projects Agency
https://www.darpa.mil/about-us/darpa-perspective-on-ai

Shneiderman, B.
Algorithmic Accountability: Designing for Safety,
Radcliffe Institute for Advanced Study, Harvard University
https://www.radcliffe.harvard.edu/video/algorithmic-accountability-designing-safety-ben-shneiderman

Don Monroe is a science and technology writer based in Boston, MA, USA.

ACM Member News

MAKING ROBOTS SMARTER THROUGH IMPROVED MOTION ALGORITHMS

Tomás Lozano-Pérez, a professor of Computer Science and Engineering at the Massachusetts Institute of Technology (MIT), had no idea what a computer was when he was growing up. During his freshman year at MIT, he took a class in programming; Lozano-Pérez says he "loved it, and never looked back."

Born in Guantánamo, Cuba, in 1952, Lozano-Pérez left for Miami at the age of 10, then attended high school in Puerto Rico. He went on to earn his B.S., M.S., and Ph.D. degrees in Electrical Engineering and Computer Science, all from MIT, in 1973, 1976, and 1980, respectively. He joined the faculty of MIT in 1981; today, he is the university's School of Engineering Professor in Teaching Excellence, as well as a member of the university's Computer Science and Artificial Intelligence Laboratory.

Lozano-Pérez says his main research interest has always been robotics, particularly with respect to algorithms for planning motions that enable higher-level functions in robots, to make them smarter.

In the early 1990s, feeling like the field of robotics was stalled and going nowhere, Lozano-Pérez took leave for several years to work at start-up Arris Pharmaceuticals, where he was immersed in computational biology and machine learning. He considers this experience invaluable, as he was able to apply what he learned about machine learning at Arris to robotics when he returned to the MIT faculty.

Today, Lozano-Pérez feels it is much more realistic to talk about intelligent robots. He is now more optimistic about the possibilities of integrating artificial intelligence and robotics, as he intended 30 or 40 years ago when he entered the field.

—John Delaney

Technology | DOI:10.1145/3276746

Neil Savage

A New Movement in Seismology

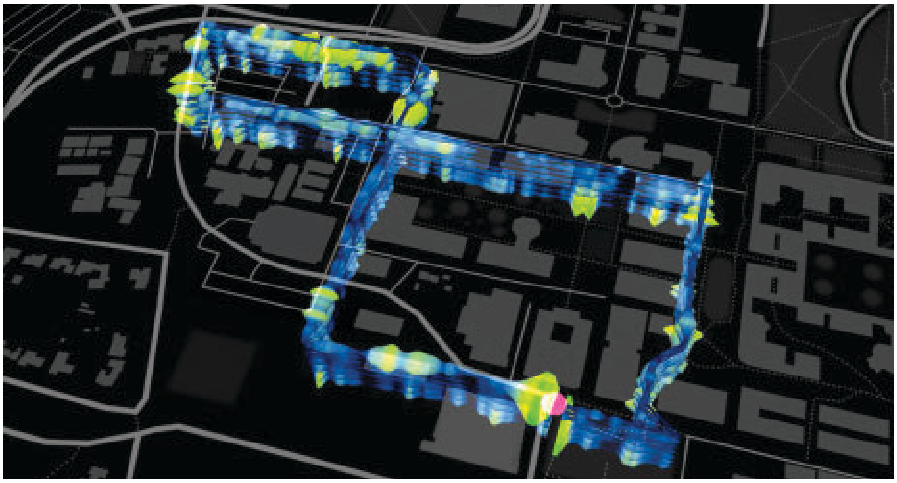
Unused telecom fiber might be used to detect earthquakes, uncover other secrets in the soil.

Whenever an earthquake strikes, news reports quickly fill in certain details, such as how strong the quake was and where it was centered. That information comes from a networks of seismometers scattered across the planet. Seismometers, though, can be expensive to install and maintain over long periods, and researchers cannot place them everywhere they might like, such as in the densely built and expensive streets of an earthquake-prone city like San Francisco.

Some scientists, however, are exploring a different approach, using a sensor that is already widely deployed beneath the streets of towns and cities around the world. That sensor is the common fiber-optic cable, used to carry telephone and Internet traffic.

The technique is called distributed acoustic sensing, and the oil and gas industry has used it for several years to monitor the ground around wells it drills. A laser beam traveling through an optical fiber will sometimes strike an impurity in the glass fiber, and part of the beam will be reflected. When an acoustic wave traveling through the Earth strikes a fiber, it stretches or compresses that fiber by a tiny amount. Using an interferometer, researchers can detect changes in the backscattered light and use that to measure the strain on the fiber which, in turn, provides information about the sound wave that struck the fiber.

Oil companies, of course, install such fiber specifically for sensing. There are, however, many thousands of kilometers of fiber lying around unused. When telecom companies install fiber in the ground, they put in far more than they actually need. Fiber is relatively cheap, but digging up and repaving city streets is expensive, so telecoms add so-called dark fiber as a hedge against future needs.



A loop of optical fibers beneath Stanford University monitors ground movement.

IMAGE BY STAMEN DESIGN AND THE VICTORIA AND ALBERT MUSEUM

That provides researchers like Biondo Biondi, a professor of geophysics at Stanford University, with an opportunity. "You can basically convert a piece of fiber into a virtual sensor," he says.

Since September 2016, Biondi has been using a 4.8-km-long loop of fiber installed under the Stanford campus to see what sort of signals he could pick up and interpret. The signals, he says, are very noisy.

Fibers used in the oil industry are cemented alongside the drill hole, and are therefore very well coupled to the ground; when the earth moves, they move in concert with it. Dark telecom fiber, though, generally lays untethered in a plastic conduit, rubbing and bumping against the conduit's wall or other fibers. That makes it a lot less sensitive than the high-quality seismometers deployed by the U.S. Geological Survey (USGS) for earthquake monitoring.

Researchers have not made exact comparisons between the quality of signal that fiber sensors pick up and those the broadband seismometers of USGS use, says Jonathan Ajo-Franklin, a staff scientist in the Energy Geoscience Division at the U.S. Department of Energy's Lawrence Berkeley National Laboratory in Berkeley, CA. He has found that the signal-to-noise ratio in the fiber sensors is about 10 to 15 decibels worse than that in geophones, another type of sensor that is itself less sensitive than a seismometer.

However, what the dark fibers lack in sensitivity, they make up for in volume. "Even in the Bay Area (San Francisco) and in the Los Angeles area, which are probably the most instrumented areas in country, you have a seismometer every 10 or 20 kilometers," Biondi says. "Here, you can have a sensor every five meters."

One interrogator box the size of a small rack-mounted server can measure reflections from hundreds or thousands of spots along a fiber, and determine their location based on the round-trip time of the laser light. The lasers are improving in quality and power, and in the not-too-distant future, Biondi says, he can imagine using perhaps 1,000 interrogator boxes to create millions of virtual sensors. "I jokingly say before I retire, I'd like to have an array under the Bay Area that has a billion sensors."

For now, Biondi's test network contains 600 virtual sensors about 5 to 7 meters from each other, taking measurements at a rate of 50 times per second. He has used it to create a database of 3,000 small, local seismic events, and to identify other sources of noise picked up by the fiber, such as traffic, construction equipment, and water pumps. His group is working on machine learning algorithms to automatically identify earthquakes, using labeled data from what they have already collected as the training set. The hope is to be able to identify even smaller earthquakes, and to use such information to develop better maps of fault areas that can be fed into earthquake simulations to project the risks to buildings and people in the earthquake zone. "Really, to have an estimate of a hazard, you need to propagate these seismic waves in the computer using the right properties," he says.

Ajo-Franklin, a sometime collaborator with Biondi, has used the same approach on the Department of Energy's Dark Fiber Testbed to detect and identify traffic noise. He wants to find ways to distinguish the seismic waves produced by a passing semi-trailer truck from those coming from a tectonic shift, particularly since fiber is often installed along roads and railroad beds. The waves from an earthquake have frequency and amplitude characteristics distinct from those of traffic, and they originate from one area, whereas the source of waves from a vehicle moves down the road along with the car. If a computer can automatically identify those differences, it can subtract them out from the waves it is trying to identify.

Ajo-Franklin is using versions of what he calls "classical seismology algorithms for ambient noise imaging scaled up to much larger computers." He has collected about eight months of measurements from a 30km. stretch of fiber north of Sacramento, generating about 300TB of data. "That's a lot of signal processing, and it's a lot of data manipulation, which is tough when you have these volumes," he says.

One challenge is finding the right compression algorithm to shrink the data to a more manageable size. Ajo-Franklin is sure there are plenty of bits that do not convey any useful information, but he is not certain of the best way to find and eliminate them. Standard audio compression techniques are designed to deal with millions of channels that have some coherence across them, which is not what this data looks like. He is exploring signal-processing algorithms, but has not yet hit on a solution.

Some sounds from sources other than earthquakes can be useful in their own right. The way seismic waves propagate is affected by factors such as how densely packed soil is and how much moisture it contains. Measuring the movement of surface waves from traffic allows scientists to map the properties of the soil, so they can tell how likely it is to give way during an earthquake. Ajo-Franklin is working on predicting soil erosion on the sides of roads in Alaska caused by melting permafrost using distributed acoustic sensing (although in that case, the researchers installed the fiber themselves).

Some seismologists want to use sound to make even deeper maps of the subsurface, to depths of 100 meters or more, which could be useful in predicting how a building will react to an earthquake.

Ocean waves produce a constant low rumble that moves around the planet. Those sound waves change as they move through the different components of the Earth's top layer—reflected, sped up, or slowed down as they encounter materials such as hard granite or soft water-saturated sediment. Scientists can measure those changes to create a picture of the subsurface, much the way an ultrasound machine can image a fetus. "The physics behind it are very similar," says Andreas Fichtner, a professor of seismology and wave physics at the Swiss Federal Institute of Technology in Zurich, Switzerland, who recently attended a workshop on medical imaging to pick up ideas on how to handle the data.

While others are studying traffic noise, Fichtner is trying to avoid any human-produced sound entirely, to gain a better understanding of what natural sounds dark fiber sensors can detect. He is setting up an experiment 500 meters deep in an old tunnel in the Alps, built to store nuclear waste but now empty. "Before we start using the cables in a quantitative way, we have to understand what we measure," he says.

While more seismologists are becoming interested in taking advantage of dark fiber for sensing, Ajo-Franklin says, they are still figuring out just what quality of data they can get. What makes the work worthwhile is that fiber is pretty much everywhere, even under oceans. "The oceans are a place where there are almost no seismometers," he says. "Except where you have islands, it's a big gap in our coverage of looking at the way seismic waves travel through the earth." He has spoken with a South American telecom company that has some undersea cables that were severed, so they cannot be used for communications, but could still serve as sensors.

If sensing with dark fiber works the way these researchers hope, it will give geologists the opportunity to study not only earthquakes, but things like distribution of ground water in almost any place they care to look. "That's the neat thing about it," Ajo-Franklin says, "because suddenly you have this sensor which is across big, big sections of important basins in the world, and you can measure all of these different attributes that were previously inaccessible." C

Further Reading

Dou, S., Lindsey, N., Wagner, A.M., Daley, T.M., Freifeld, B., Robertson, M., Peterson, J., Ulrich, C., Martin, E.R., and Ajo-Franklin, J.B.
Distributed Acoustic Sensing for Seismic Monitoring of The Near Surface: A Traffic-Noise Interferometry Case Study, *Nature Scientific Reports*, September 2017

Martin, E.R., Huot, F., Ma, Y., Cieplicki, R., Cole, S., Karrenbach, M., and Biondi, B.L.
A Seismic Shift in Scalable Acquisition Demands New Processing, *IEEE Signal Processing*, March 2018.

Kruger, L.
Dark Fiber is Lighting Up, *Communications of the ACM*, October 2013

Martin, E.R., Castillo, C.M., Cole, S., Sawasdee, P.S., Yuan, S., Clapp, R., Karrenbach, M., and Biondi, B.L.
Seismic monitoring leveraging existing telecom infrastructure at the SDASA: Active, passive, and ambient-noise analysis, *The Leading Edge*, December 2017

Ajo-Franklin, J.
Fiber-optic distributed sensing https://www.youtube.com/watch?v=-IdEZJzjkC4

Neil Savage is a science and technology writer based in Lowell, MA, USA.

Society | DOI:10.1145/3276744

Samuel Greengard

Weighing the Impact of GDPR

The EU data regulation will affect computer, Internet, and technology usage within and outside the EU; how it will play out remains to be seen.

When the European Union (EU) General Data Protection Regulation (GDPR) went into effect on May 25, 2018, it represented the most sweeping effort yet to oversee the way businesses collect and manage consumer data. The law, established to create consistent data standards and protect EU citizens from potential privacy abuses, sent ripples—if not tidal waves—across the world.

GDPR gives European citizens greater control of their data while establishing strong penalties for businesses that do not comply. What is more, any data that involves EU citizens or touches EU companies is covered by GDPR. The initiative replaces an older data privacy initiative called the Data Protection Directive 95/46/EC, which was introduced in 1995.

The implications and ramifications are enormous—and the initiative's reach is global. GDPR will change everything from the way data collection takes place to the way corporate databases are designed and used. It also will potentially change the way research and development takes place, and will impact cybersecurity practices, as well as introducing a practical array of challenges revolving around sites and repositories where groups share comments, information, and other data.

"It's a groundbreaking initiative," says Brett M. Frischmann, Charles Widger Endowed University Professor in Law, Business, and Economics at Villanova University, and Affiliate Scholar of the Center for Internet and Society at Stanford Law School. "Europe has flipped a switch and prompted reconsideration of how data can be collected, managed, and used." The EU takes the position that a person owns his or her data, and their privacy is a fundamental right that is "basic to the integrity of a human being," Frischmann adds.

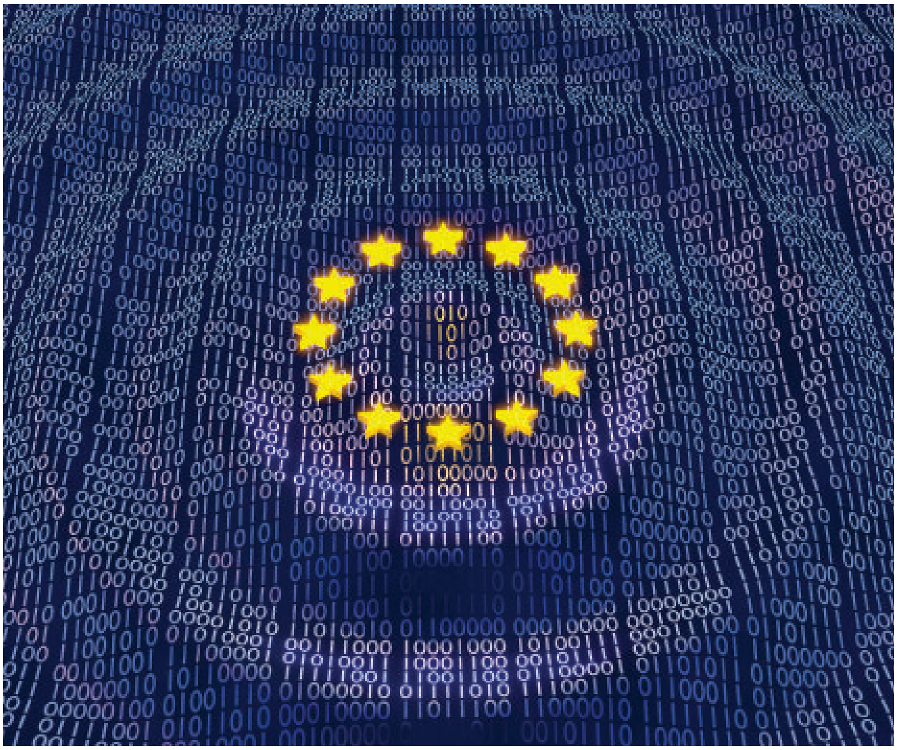


IMAGE BY MIXMAGIC

Data Wars

Digital technology has inexorably changed the face of privacy. Today, there is a perception—and plenty of evidence to support it—that personally identifiable information (PII) is under assault as never before. A Pew Research Center survey found that in the U.S., 93% of adults say being in control of who can get information about them is important; 90% say controlling what information is collected is important. The figures in Europe and other parts of the world are the same.

In a 2016 interview in *Recode*, Europe's Competition Commissioner Margrethe Vestager said, "There is no such thing as a free lunch. You pay with one currency or another—either cents, or you pay with your data, or you pay with the advertisements that you accept. And I think people are becoming more and more aware of the fact that their personal data do have a value."

Says Alison Cool, assistant professor of anthropology at the University of Colorado, Boulder, "There are a lot of questions and ambiguities that must be addressed, but it's clear that GDPR will significantly change the data landscape."

While the U.S. and a number of other countries have adopted an opt-out approach to data collection—essentially, a consumer must instruct a company if he or she doesn't want his or her data used or shared in certain ways—Europe has implemented a more restrictive opt-in approach. However, GDPR takes this concept to a new and previously untested level. Besides giving consumers near-total control of their data, they can have their data removed from a database or online source at any time and, for those who believe they have been wronged, seek an investigation and join a class-action lawsuit.

Strict rules about how organizations collect, manage, and process data anywhere in the world are only the starting point for GDPR. It allows consumers to file complaints with each nation's national data protection authority, which will investigate the claim. A company that violates GDPR could face a fine of up to 4% of its worldwide annual revenue from the previous fiscal year. The regulation also mandates consumers can remove themselves from a database at any time and take their data elsewhere—to a new bank, a new mobile provider, or a new content service.

Not surprisingly, data scientists, legal experts, and others have radically different perspectives of GDPR. Says Daniel Le Métayer, Senior Research Scientist at Inria (the French Institute for Research in Computer Science and Automation) and a leading authority on data protection and privacy, "GDPR could be a great achievement if properly implemented. It could establish a more concrete framework for data use and protection and help reduce the misuse of personal information."

Adds Cool: "It potentially changes the balance of power. GDPR takes aim at the widely used model of forced consent, which is built on the idea that in exchange for various services, there is an implicit agreement to give up your personal data."

However, there are also plenty of potential pitfalls likely to result from GDPR. Le Métayer says the complexity of GDPR, and the way regulators and courts interpret some of the intentionally vague wording, could create such rigid restrictions that the initiative becomes ineffective over time.

There is also strong opposition in the corporate arena, where the focus is on profiting from data rather than stemming the wave of abuses and breaches. Attorneys such as Tanya Forsheit, partner and chair of the Privacy & Data Practice Group at New York City-based law firm Frankfurt Kurnit Klein & Selz, demonstrate the level of frustration about changes as a result of GDPR. Forsheit describes many GDPR provisions as onerous, and suggests they could be more effectively addressed through self-regulation. "It is simply not possible to be 100% compliant. GDPR forces organizations to devote significant time and expense to comply with standards that are not consistent with the way business is done online," she argues.

GDPR allows consumers to remove themselves from a database or online source at any time; companies violating GDPR face fines of up to 4% of their global annual revenues.

Data Gets Personal

To be sure, the practical challenges of complying with GDPR are significant, especially as digital technology and artificial intelligence (AI) advance.

Personal assistants such as Siri, Alexa, and Cortana add layers of complexity to the issue of PII. Robo-advisors, chatbots, recommendation services, and other automated systems introduce additional compliance challenges. All these systems collect and store data about individuals. In the past, there was no need to determine where a person lived; under GDPR, that could amount to crucial information that would need to be added to each individual data point related to an individual. Even human resources systems, payroll systems, and similar repositories of personal data could be significantly impacted by the regulation; all may require algorithmic auditing processes that revolve around "data protection by design."

Companies already are voicing concerns that GDPR could inhibit innovation by limiting how data is handled in apps, databases, and online services—and how data is used for advertising and other purposes. The issue could impact autonomous vehicles,

Milestones

Håstad Receives Knuth Prize

The 2018 Donald E. Knuth Prize has been awarded to Johan Torkel Håstad of Sweden's KTH Royal Institute of Technology for his sustained record of milestone breakthroughs at the foundations of computer science, with major impact on areas including optimization, cryptography, parallel computing, and complexity theory.

The Knuth Prize is jointly bestowed by the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT) and the IEEE Computer Society Technical Committee on the Mathematical Foundations of Computing (TCMF). The Prize is named for Donald Knuth of Stanford University, the "father of the analysis of algorithms," and is bestowed in recognition of outstanding contributions to the foundations of computer science by individuals for their overall impact in the field over an extended period.

A professor of computer science at the KTH Royal Institute of Technology in Stockholm, Håstad received his bachelor's degree in mathematics from Stockholm University, his master's degree in mathematics from Sweden's Uppsala University, and his doctorate in mathematics from the Massachusetts Institute of Technology.

Håstad's works resolved long-standing problems central to circuit lower bounds, pseudorandom generation, and approximability. He also introduced transformative techniques that have fundamentally influenced much of the subsequent work in these areas.

Previous honors bestowed on Håstad include the ACM Doctoral Dissertation Award (1986), the Gödel Prize for outstanding papers on theoretical computer science (1994 and 2011), and the Göran Gustafsson Prize for outstanding achievement in mathematics.

robotics, and a variety of systems that rely on AI. Organizations may ultimately need to keep two separate databases—one for the EU and one for elsewhere—or find ways to differentiate records in databases.

In addition, GDPR might add a layer of complexity atop an already complex European privacy framework. For example, more than 2.4 million individuals have already submitted "right to be forgotten" requests so they can be expunged from Google searches. Cool says some people believe the law will "hinder innovation by making organizations more risk averse."

Depending on who opts in, who opts out, and what data appears or disappears from a database or other source, the situation could become even more problematic. As Frischmann puts it, "What happens when one person at a group meeting or part of a community invokes a privacy clause but it affects everyone?"

The greatest challenge may be ensuring companies in the EU and beyond adhere to the spirit of GDPR. Many companies lack expertise in how they will need to implement and manage data under GDPR; they also do not know the levels of expertise or staffing required to conduct crucial data protection impact assessments.

"If businesses view GDPR as a checklist activity rather than an issue that requires ethical reflection—and if they look to exploit loopholes and skirt the intent of the law—the long-term outcome could be negative," Cool says. "When you look at groups like bioethicists and physicians, the starting point for discussion is how to do the right thing for society; it's not about avoiding getting sued or how to sidestep legal and ethical provisions."

Cracking the Code on Privacy

How GDPR will play out is anyone's guess. The initiative could revolutionize the data landscape—or it may fizzle into a footnote in digital history. It could also change the way the Internet works and how data and information flow across sites, clouds, and more.

One wild card is how consumers react to GDPR. If large numbers of people revoke access to PII or challenge the way companies use their data, businesses may reach an inflection point where they will have to rethink the fundamental way they approach and navigate data management, or reevaluate the fundamental value of data and how it is monetized. GDPR also might mandate new tracking and data management tools, such as blockchain.

Above all else, GDPR represents the ongoing battle between unfettered capitalism and human dignity.

Le Métayer argues that businesses need to address complex issues such as conducting data protection impact assessments and implementing data portability, which requires agreeing on standard data formats. Other sources of uncertainty include the compatibility of GDPR with big data, and the rules concerning automated decision-making. Article 22 of GDPR states individuals have the right to "not be subject to a decision based solely on automated processing, including profiling." GDPR also allows consumers to contest a decision, but it is not clear what type of explanations should be provided to make this right effective. "The issue is also technical, since providing useful explanations about certain types of algorithms is a challenge in itself," he says.

GDPR could also prompt companies to directly pay for PII data, Frischmann says. "If the power balance shifts and consumers gain leverage over their personal data, companies may look to provide incentives, discounts, and direct compensation for the use of data. It could flip the current model and even lead to entirely different ways to approach data," he explains.

In fact, a recent study conducted by digital marketing agency Syzgy in Germany, which polled 1,000 respondents each from the U.S., U.K., and Germany, found citizens in all three countries would sell their data for between €130 (about US$150) and €140 (US$165) per month.

One thing is certain: amid a litany of security breaches and breakdowns, from Equifax to Cambridge Analytica, there is a growing focus on data privacy. What is more, other government entities are exploring ways to control how data is collected, managed, and used. In the U.S., the State of Vermont enacted a law in May 2018 that established standards for data. California is now eyeing an initiative—the California Consumer Privacy Act—that could extend many of the same GDPR protections to the state. Other countries, from Australia to Japan, have also revised data standards and privacy controls in recent years.

Frischmann says GDPR, above all else, represents the ongoing battle between unfettered capitalism and human dignity. The whole point of it is that it is not designed to be an efficient regulation for businesses. "To some extent, it's about a person's ability to exercise their own free will about their life."

Cool says that, in the end, it is vital to strike a balance between privacy and laws. "We need more research that looks carefully at how personal data is collected and by whom, and how those people make decisions about data protection. Policymakers should use such studies as a basis for developing empirically grounded, practical rules." C

Further Reading

Wachter, S., Mittelstadt, B.D., and Russell, C.
Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR, *Harvard Journal of Law & Technology*, 31 (2), 2018. November 2017. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3063289

Casey, B., Farhangi, A., and Vogl, R.
Rethinking Explainable Machines: The GDPR's 'Right to Explanation' Debate and the Rise of Algorithmic Audits in Enterprise, *Berkeley Technology Law Journal*. February 19, 2018. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3143325

Kaltheuner, F. and Bietti, E.
Data is power: Towards additional guidance on profiling and automated decision-making in the GDPR. *Information Rights, Policy & Practice Journal*, Vol. 2, No. 2. 2017. https://journals.winchesteruniversitypress.org/index.php/jirpp/article/view/45

Samuel Greengard is an author and journalist based in West Linn, OR, USA.

AWARD NOMINATIONS SOLICITED

As part of its mission, ACM brings broad recognition to outstanding technical and professional achievements in computing and information technology.

ACM welcomes nominations for those who deserve recognition for their accomplishments. Please refer to the ACM Awards website at **https://awards.acm.org** for guidelines on how to nominate, lists of the members of the 2018 Award Committees, and listings of past award recipients and their citations.

Nominations are due **January 15, 2019** with the exceptions of the Doctoral Dissertation Award (due **October 31, 2018**) and the ACM – IEEE CS George Michael Memorial HPC Fellowship (due **May 1, 2019**).

A.M. Turing Award: ACM's most prestigious award recognizes contributions of a technical nature which are of lasting and major technical importance to the computing community. The award is accompanied by a prize of $1,000,000 with financial support provided by Google.

ACM Prize in Computing (previously known as the ACM-Infosys Foundation Award in the Computing Sciences): recognizes an early- to mid-career fundamental, innovative contribution in computing that, through its depth, impact and broad implications, exemplifies the greatest achievements in the discipline. The award carries a prize of $250,000. Financial support is provided by Infosys Ltd.

Distinguished Service Award: recognizes outstanding service contributions to the computing community as a whole.

Doctoral Dissertation Award: presented annually to the author(s) of the best doctoral dissertation(s) in computer science and engineering, and is accompanied by a prize of $20,000. The Honorable Mention Award is accompanied by a prize totaling $10,000. Winning dissertations are published in the ACM Digital Library and the ACM Books Series.

ACM – IEEE CS George Michael Memorial HPC Fellowships: honors exceptional PhD students throughout the world whose research focus is on high-performance computing applications, networking, storage, or large-scale data analysis using the most powerful computers that are currently available. The Fellowships includes a $5,000 honorarium.

Grace Murray Hopper Award: presented to the outstanding young computer professional of the year, selected on the basis of a single recent major technical or service contribution. The candidate must have been 35 years of age or less at the time the qualifying contribution was made. A prize of $35,000 accompanies the award. Financial support is provided by Microsoft.

Paris Kanellakis Theory and Practice Award: honors specific theoretical accomplishments that have had a significant and demonstrable effect on the practice of computing. This award is accompanied by a prize of $10,000 and is endowed by contributions from the Kanellakis family, and financial support by ACM's SIGACT, SIGDA, SIGMOD, SIGPLAN, and the ACM SIG Project Fund, and individual contributions.

Karl V. Karlstrom Outstanding Educator Award: presented to an outstanding educator who is appointed to a recognized educational baccalaureate institution, recognized for advancing new teaching methodologies, effecting new curriculum development or expansion in computer science and engineering, or making a significant contribution to ACM's educational mission. The Karlstrom Award is accompanied by a prize of $10,000. Financial support is provided by Pearson Education.

Eugene L. Lawler Award for Humanitarian Contributions within Computer Science and Informatics: recognizes an individual or a group who have made a significant contribution through the use of computing technology; the award is intentionally defined broadly. This biennial, endowed award is accompanied by a prize of $5,000, and alternates with the ACM Policy Award.

ACM – AAAI Allen Newell Award: presented to individuals selected for career contributions that have breadth within computer science, or that bridge computer science and other disciplines. The $10,000 prize is provided by ACM and AAAI, and by individual contributions.

Outstanding Contribution to ACM Award: recognizes outstanding service contributions to the Association. Candidates are selected based on the value and degree of service overall.

ACM Policy Award: recognizes an individual or small group that had a significant positive impact on the formation or execution of public policy affecting computing or the computing community. The biennial award is accompanied by a $10,000 prize. The next award will be the 2019 award.

Software System Award: presented to an institution or individuals recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both. A prize of $35,000 accompanies the award with financial support provided by IBM.

ACM Athena Lecturer Award: celebrates women researchers who have made fundamental contributions to computer science. The award includes a $25,000 honorarium.

For SIG-specific Awards, please visit **https://awards.acm.org/sig-awards**.

Vinton G. Cerf, ACM Awards Committee Co-Chair

John R. White, ACM Awards Committee Co-Chair

Insup Lee, SIG Governing Board Awards Committee Liaison

Rosemary McGuinness, ACM Awards Committee Liaison

DOI:10.1145/3277562

Pamela Samuelson

Legally Speaking
The EU's Controversial Digital Single Market Directive

Should copyright enforcement have precedence over the interests of users in information privacy and fundamental freedoms?

The stated goals of the EU's proposed Digital Single Market (DSM) Directive are laudable: Who could object to modernizing the EU's digital copyright rules, facilitating cross-border uses of in-copyright materials, promoting growth of the internal market of the EU, and clarifying and harmonizing copyright rules for digital networked environments?

The devil, as always, is in the details. The most controversial DSM proposal is its Article 13, which would require online content-sharing services to use "effective and proportionate" measures to ensure user uploads to their sites are non-infringing. Their failure to achieve this objective would result in their being directly liable for any infringements. This seemingly requires those services to employ monitoring and filtering technologies, which would fundamentally transform the rules of the road under which these firms have long operated.

A more positive part of the DSM Directive is its Article 3. It would require EU member states to adopt a copyright exception to enable research and cultural heritage institutions to engage in text- and data-mining (TDM) for scientific research purposes. This is good so far as it goes, but critics argue that for-profit firms and independent researchers should enjoy similar TDM privileges, and scientific research should not be the only legitimate purpose for TDM.

This column explains the rationales for these new measures, specific terms of concern, and why critics have argued for changes to make the rules more balanced. (Column space limitations preclude attention to other controversial provisions, such as the new press publishers' rights to control online services' displays of press contents.)

Big media firms can use this new rule to extract more compensation from platforms.

Article 13's Changes to Online Service Liability Rules

For approximately the past two decades, the European Union's E-Commerce Directive, like the U.S. Digital Millennium Copyright Act, has provided Internet service providers (ISPs) with "safe harbors" from copyright liability for infringing uses of their services about which the ISPs had neither knowledge nor control.

Under these rules, ISPs must take down infringing materials after copyright owners notify them of the existence and location of those materials. But they do not have to monitor for infringements or use filtering technologies to prevent infringing materials from being uploaded and stored on their sites.

Because online infringements have greatly proliferated, copyright industry groups have strongly urged policymak-

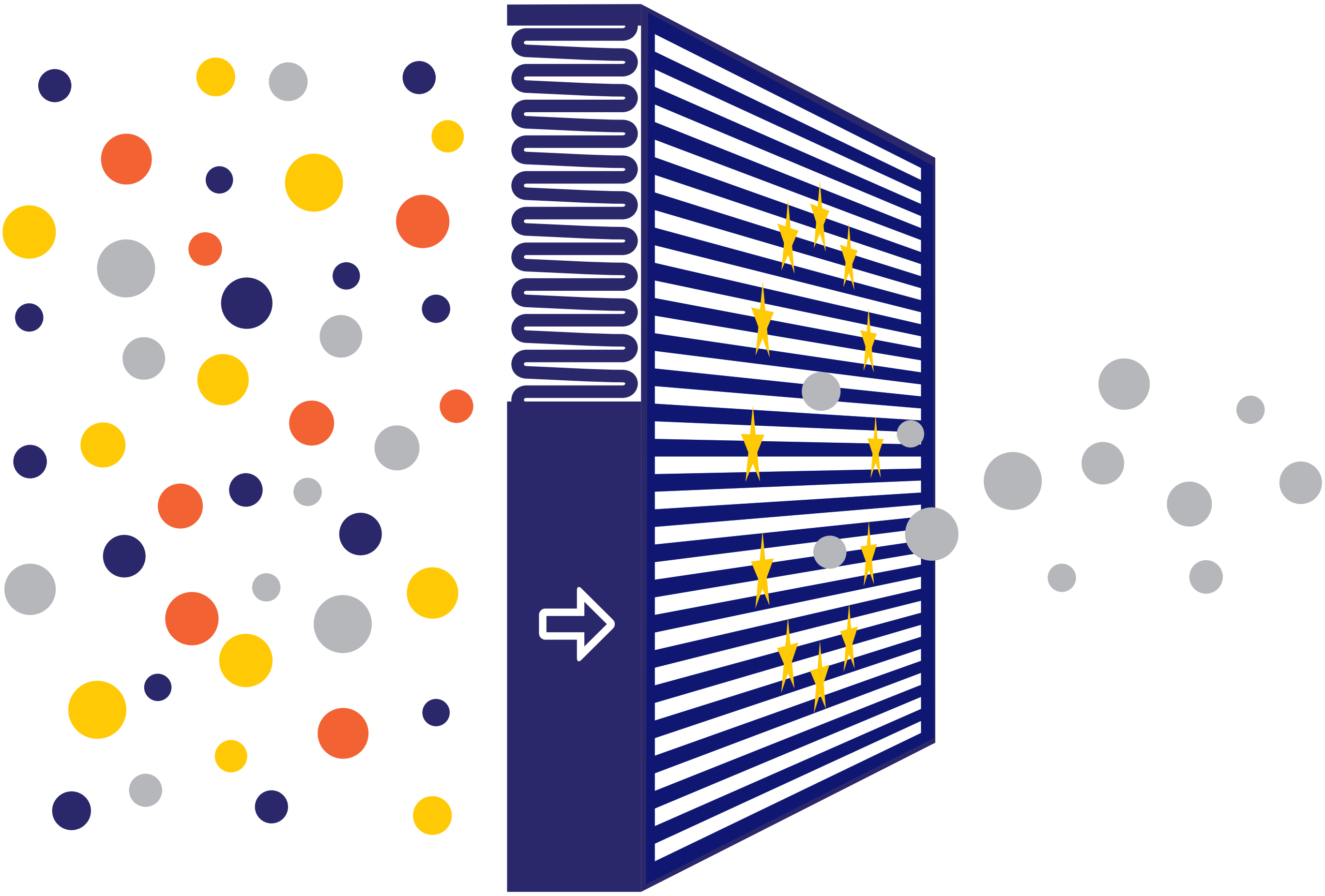


IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

ers in the EU (as well as the U.S.) to impose stronger obligations on ISPs to thwart infringements. Their goal has been the adoption of legal rules requiring ISPs to use monitoring technologies to detect in-copyright materials and filtering technologies to block infringing uploads.

In proposing the DSM Directive, the European Commission has responded to these calls by proposing certain ISPs should take on greater responsibilities on to help prevent infringements. Article 13 is aimed at those ISPs that enable online content sharing (think YouTube).

While not directly requiring the use of monitoring or filtering technologies, Article 13 can reasonably be interpreted as intending to achieve this result.

Which Online Services Are Affected?

The DSM Directive states Article 13 is intended to target only those online content-sharing services that play an "important role" in the online content market by competing with other services, such as online audio or video streaming services, for the same customers.

If the "main purpose" (or "one of the main purposes") of the service is to provide access to "large amounts" of copyrighted content uploaded by users and it organizes and promotes those uploads for profit-making purposes, that service will no longer be protected by the E-Commerce safe harbor. It will instead be subjected to the new liability rules.

Concerns about the overbreadth of Article 13 led the Commission to narrow the definition of the online content-sharing services affected by the rules. It now specifically excludes online encyclopedias (think Wikipedia), repositories of scientific or educational materials uploaded by their authors, open source software repositories, cloud services, cyberlockers, and marketplaces engaged in online retail sales of digital copies.

Article 13's New Liability Rules

The most significant regulation in Article 13 is its subsection (4):

> Member States shall provide that an online content sharing service provider shall not be liable for acts of communication to the public or making available to the public within the meaning of this Article when:
>
> (a) it demonstrates that it has made *best efforts* to prevent the availability of specific works or other subject matter by implementing *effective and proportionate measures* … to prevent the availability on its services of the specific works or other subject matter identified by rightholders and for which the rights holders have provided the service with *relevant and necessary information for the application of these measures*; and
>
> (b) upon notification by rights holders of works or other subject matter, it has acted expeditiously to remove or disable access to these works or other subject matter and it demonstrates that it has made its *best efforts to prevent their future availability* through the measures referred to in point (a).

The italicized language signals terminology that is vague and open to varying interpretations, but anticipates the use of technologies to show those "best efforts."

Copyright industry groups can be expected to assert that it is necessary to use monitoring and filtering technologies to satisfy the requirements of Article 13(4). They will also point to an alternative way that online services can avoid liability: by licensing uploaded copyrighted content from their respective rights holders.

Affected online services will have an uphill battle to fend off the efforts to interpret the ambiguous terms as imposing monitoring and filtering obligations. It is, of course, impossible to license contents for every copyrighted work that their users might upload to their site. But the big media firms can use this new rule to extract more compensation from platforms.

Concerns About Article 13's Liability Rules

Critics have raised two major concerns about this proposal. First, it will likely further entrench the market power of the leading platforms that can afford to develop filtering technologies such as YouTube's ContentID, and deter new entry into the online content sharing market. Second, it will undermine user privacy and free speech interests, leading to blockages of many parodies, remixes, fan fiction, and other creative reuses of copyrighted works that would, if examined by a neutral observer, be deemed non-infringing.

When the proposal was pending before the European Council in late May, several members, including representatives from Finland, Germany, and the Netherlands, opposed it and offered some compromise language, so it does not have consensus support. Since then, opponents have mounted a public relations campaign to urge EU residents to contact their Parliamentary representatives telling them to vote no in order to "save the Internet."

Among the many critics of Article 13 is David Kaye, the United Nation's Special Rapporteur for Freedom of Expression. He wrote a nine-page letter explaining why Article 13 is inconsistent with EU's commitments under international human rights instruments.

In addition, Tim Berners-Lee, Vint Cerf, and 89 other Internet pioneers (plus me) signed an open letter urging the EU Parliament to drop Article 13: "By requiring Internet platforms to perform automatic filtering on all of the content that their users upload, Article 13 takes an unprecedented step toward the transformation of the Internet from an open platform for sharing and innovation, into a tool for the automated surveillance and control of its users."

More than 145 civil society organizations also came out against it. These protests were successful enough to induce a majority of the European Parliament to vote for giving further consideration to the DSM directive. Several stages remain in the EU's elongated process before this directive is finalized, either in its current or some revised form.

Mandatory Text- and Data-Mining Exception

Much better news is the proposed new copyright exception to enable nonprofit research and cultural heritage institutions to engage in text- and data-mining (TDM). The European Commission and the Council recognize that digital technologies have opened up significant opportunities for using TDM techniques to make new discoveries by computational analysis of large datasets. These discoveries can advance not only natural but also human sciences in ways that will benefit the information society.

Article 3 would require EU member states to allow research and cultural heritage institutions to reproduce copyrighted works and extract information using TDM technologies, as long as the researchers had lawful access to the contents being mined. These researchers must, however, store such copies in a secure environment and retain the copies no longer than is necessary to achieve their scientific research objectives.

Importantly, rights holders cannot override the TDM exception through contract restrictions. (They can, however, use technology to ensure security and integrity of their networks and databases, which opens the possibility of technology overrides.) Article 3 also calls for rights holders, research organizations, and cultural heritage institutions to agree upon best practices for conducting TDM research.

No TDM Privilege for Profit-Making and Unaffiliated Researchers

The DSM Directive assumes that profit-making firms can and should get a license to engage in TDM research from the owners of the affected IP rights. Although the DSM contemplates the possibility of public-private partnerships, it forbids those in which private entities have control over TDM-related collaborative projects. Unaffiliated researchers (say, independent data scientists or think-tank personnel) cannot rely on the DSM's TDM exception.

Article 3 may put the EU at a disadvantage in AI research because some countries have already adopted less restrictive TDM exceptions. Japan, for instance, allows text- and data-mining without regard to the status of the miner, and does not confine the scope of the exception to nonprofit "scientific research." In the U.S., for-profit firms have been able to rely on fair use to make copies of in-copyright materials for TDM purposes, as in the *Authors Guild v. Google* case. This ruling did not limit TDM purposes to scientific research.

Commentators on the DSM Directive have expressed several concerns about the restrictions on its TDM exception. For one thing, TDM licenses may not be available on reasonable terms for startups and small businesses in the EU. Second, some EU firms may ship their TDM research offshore to take advantage of less-restrictive TDM rules elsewhere. Third, some non-EU firms may decide not to invest in TDM-related research in the EU because of these restrictions. Moreover, in the highly competitive global market for world-class AI and data science researchers, the EU may suffer from "brain drain" if its most talented researchers take job opportunities in jurisdictions where TDM is broadly legal.

Conclusion

The EU's proposed DSM Directive is highly controversial, especially the new obligations it would impose on online content-sharing services to thwart infringing uploads. In early July, the EU Parliament voted against giving approval to the May version of the DSM proposal; it voted in September to approve some amendments to the DSM Directive, which did not significantly change the Article 13 mandate. It will, however, be many months before the final text of the directive is voted on.

The prospect of bearing direct liability for the infringing activities of users will likely cause many sharing services to be overly cautious about what their users can upload.

Whether Article 13, if adopted as is, will "kill" the Internet as we know it, as some critics have charged, remains to be seen. Yet the prospect of bearing direct liability for the infringing activities of users will likely cause many sharing services to be overly cautious about what their users can upload and new entry will be chilled. In its current form, Article 13 gives copyright enforcement priority over the interests of users in information privacy and fundamental freedoms.

The DSM Directive's proposed exception for TDM research is a welcome development for those who work at research and cultural heritage institutions. However, the unfortunate withholding of the exception from for-profit firms and independent researchers may undermine prospects for the EU's achieving its aspiration to promote innovations in AI and data science industries. It will be difficult for EU-based entities to compete with American and Japanese firms whose laws provide them with much greater freedom to engage in TDM analyses.

Pamela Samuelson (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley, and a member of the ACM Council.

Calendar of Events

November 3–7
CSCW '18: Computer Supported
Cooperative Work
and Social Computing
Jersey City, NJ,
Co-Sponsored: ACM/SIG,
Contact: Mor Naaman,
Email: mornaaman@yahoo.com

November 4–7
SenSys '18: The 16th ACM
Conference on Embedded
Network Sensor Systems
Shenzhen, China,
Co-Sponsored: ACM/SIG,
Contact: Pei Zhang,
Email: peizhang@cmu.edu

November 4–9
ESEC/FSE '18: The 26th ACM
SIGSOFT International
Symposium on the Foundations
of Software Engineering
Lake Buena Vista, FL,
Contact: Gary T. Leavens,
Email: leavens@eecs.ucf.edu

November 4–9
Conference on Systems,
Programming, Languages,
and Applications: Software
for Humanity
Boston, MA,
Sponsored: ACM/SIG,
Contact: Jan Vitek,
Email: j.vitek@neu.edu

November 5–6
11th ACM SIGPLAN International
Conference on Software
Language Engineering
Boston, MA,
Sponsored: ACM/SIG,
Contact: David James Pearce,
Email: david.pearce@ecs.vuw.
ac.nz

November 5–6
17th ACM SIGPLAN International
Conference on Generative
Programming: Concepts
and Experiences
Boston, MA,
Sponsored: ACM/SIG,
Contact: Eric Van Wyk,
Email: evw@cs.umn.edu

DOI:10.1145/3277564

Steven M. Bellovin and Peter G. Neumann

Inside Risks
The Big Picture

A systems-oriented view of trustworthiness.

PREVIOUS *COMMUNICATIONS* INSIDE RISKS columns have discussed specific types of risks (to safety, security, reliability, and so on), and specific application areas (for example, critical national infrastructures, election systems, autonomous systems, the Internet of Things, artificial intelligence, machine learning, cybercurrencies and blockchains—all of which are riddled with security problems). We have also considered risks of deleterious misuses of social media, malware, malicious drones, risks to privacy, fake news, and the meaning of "truth." All of these and many more issues must be considered proactively as part of the development and operation of systems with requirements for trustworthiness.

PHOTO BY DREW GRAHAM ON UNSPLASH

We consider here certain overarching and underlying concepts that must be better understood and more systematically confronted, sooner rather than later. Some are more or less self-evident, some may be debatable, and others may be highly controversial.

▸ A preponderance of flawed hardware-software systems, which limits the development of trustworthy applications, which also impedes accountability and forensics-worthy rapid identification of culprits and causes failures.

▸ Lack of understanding of the properties of composed systems. Components that seem secure locally, when combined, may yield insecure systems.

▸ A lack of discipline and constructive uses of computer science, physical science, technology, and engineering, which hinders progress in trustworthiness, although new applications, widgets, and snake-oil-like hype continue apace without much concern for sound usability.

▸ A lack of appreciation for the wisdom that can be gained from science, engineering, and scientific methods, which impedes progress, especially where that wisdom is clearly relevant.

▸ A lack of understanding of the short-term and long-term risks by leaders in governments and business, which is becoming critical, as is their willingness to believe that today's sloppy systems are good enough for critical uses.

▸ A widespread failure to understand these risks is ominous, as history suggests they will pervasively continue to recur in the future.

▸ A general lack of awareness and education relating to all of these issues, requiring considerable rethinking of these issues.

Background

Progress toward trustworthy systems for critical security uses has been very spotty. For example, several National Academies of Science Computer Science and Technology Board studies have examined issues relating to computer and network security[4,6,11] and cryptography,[5] with extensive conclusions and recommendations that seem to have been widely ignored, or not farsighted enough, or possibly both. Other studies have examined some of the implications of using cryptography,[1,2,7] where again related problems keep arising. Cryptography is an enormously useful concept for achieving trustworthy systems and networks; unfortunately, its effectiveness can be severely limited if it is not implemented in systems with sufficient trustworthiness. Thus, it is a trustworthiness enhancer, but cannot be relied on by itself to enable trustworthy systems and networks.

Total-System Trustworthiness

Trustworthiness is a total-system problem. That is, trustworthiness must consider not just attributes of individual elements, but also how they compose and interact. It is not uncommon for systems to fail even when every individual component is correct and seems locally secure. For example, the composition problem may be as simple as having different notions of the behavior of a particular interface—where each component might assume the other does input validation—or as complex as subtle, time- and input-dependent misbehavior under unusual circumstances. Dependencies on flawed hardware must also be considered, such as the recent speculative-execution and out-of-order execution attacks (for example, Spectre/Meltdown[14] and Foreshadow/Foreshadow-NG vulnerabilities.[15]

The so-called "Martin Luther King Day meltdown" of the AT&T long-distance network in 1990 is a classic example of what can go wrong. There was a flaw in the recovery code when a phone switch rebooted and resumed normal operation. If a neighboring switch received two incoming calls within 1/100 of a second thereafter, it would crash. This, of course, triggered the same failures in its neighbors, iteratively throughout half a day.[8]

With so many known vulnerabilities, and new ones continually being discovered, it is obvious that defenses are often overwhelmed. For example, the Common Vulnerability Enumeration (mitre.cve) is approaching 110,000 vulnerabilities—approximately 16,000 since the beginning of 2018.

More recently, consider the Foreshadow/L1 Terminal attacks on SGX discussed at USENIX Security 2018, and subsequently discovered Foreshadow-NG vulnerabilities,[13,15] which broadly affect VMs, VMMs, operating systems, and SMM memory. The NG (next-generation) paper has attacks that "completely bypass the virtual memory abstraction by directly exposing cached physical memory contents to unprivileged applications and guest virtual machines." These attacks appear to be very serious.

Overall, there are no simple solutions. Precision in interface definition is one obvious approach, although obscure cases are difficult to specify—for example, call-arrival rate at a critical time.

Trustworthiness Also Must Respect Human Behavior

Achieving trustworthiness in complex systems also depends critically on the people involved throughout system development and use. Many systems have poorly defined functional and behavioral requirements—if any. System architectures seldom reflect critical requirements, and implementations seldom adhere to those requirements or design specifications. Formal methods have significant opportunities to improve trustworthiness, but are challenging to use coherently. In operation, user wisdom and sensible behavior are often assumed (instead of building people-tolerant systems), and the creativity and power of malicious misuse and malware are inadequately considered. Thus, trustworthiness must anticipate all sorts of human behavior, as well as environmental disruptions. In essence, achieving trustworthiness is very complex, and attempts to simplify it are generally fraught with vulnerabilities.

Future Directions for Systems Research and Development

A research program in systems poses many challenges. The most difficult is one of definition: What is systems research? What constitutes real innovation? Merely having multiple components is necessary, but not sufficient. Rather, what is needed is a demonstration that new techniques either contribute to the security of the full system or let us better evaluate security. Indeed, some early projects might simply be intended to

Achieving trustworthiness is very complex, and attempts to simplify it are generally fraught with vulnerabilities.

better define the problem and lay out a suitable research agenda.

One vital approach would be a unified theory of predictable subsystem composition that can be used to develop hardware-software systems for a wide range of applications out of demonstrably trustworthy components. Formal methods could be useful selectively. What is essential, though, is that the properties being composed are actually useful in real-world systems.

However, systems design is not a formal discipline today. Therefore, carefully documented open success stories that illustrate the power of an approach are also acceptable, especially if they enable constructive opportunities for the future.

On a smaller scale, developing mechanisms and tools that advance the goal of secure systems would also be useful. Thus, a scheme that provides strong protection for cryptographic keys while still leaving them useful for authorized uses is valuable.[3] This may be facilitated by specialized hardware—if that hardware is trustworthy (including available as needed). Thus, a variety of clean-slate hardware architecture specifications that can be implemented by multiple organizations and that can facilitate total systems that are much more trustworthy would also be useful. Again, formal methods could be useful selectively to prove critical properties of some of the specifications.

Conclusion

Research and its funding have often failed us. There is too much focus on narrow problems—point solutions to point problems—and too little effort devoted to systems aspects of solutions that include considerations of human behavior. Furthermore, many problems discussed long ago[8,9] still have not been adequately addressed today. In addition, underlying principles for trustworthy systems have been posited since the 1960s and recently revisited, but widely ignored in practice.[10] A recent book also has more relevant suggestions for the future.[12]

It is time to get serious about the dearth of trustworthy systems and the lack of deeper understanding of the risks that result from continuing on a business-as-usual course. C

References

1. Abelson, H. et al. The risks of key recovery, key escrow, and trusted third-party encryption. *World-Wide Web Journal 2*, 3 (Summer 1997), 241–257.
2. Abelson, H. et al. Keys under doormats: Mandating insecurity by requiring government access to all data and communications. *Journal of Cybersecurity 1*, 1 (Nov. 2015), Oxford University Press; http://www.cybersecurity.oxfordjournals.org/content/1/1/69
3. Bellovin, S.M. The key to the key. *IEEE Security and Privacy 13*, 6 (Nov.–Dec. 2015), 96–96.
4. Clark, D.D. et al. *Computers at Risk: Safe Computing in the Information Age.* National Research Council, National Academies Press, Washington, D.C., 1990.
5. Dam, K.W. and Lin, H.S., Eds. Cryptography's role in securing the information society. National Research Council, National Academies Press, Washington, D.C., 1996.
6. Goodman, S.E. and Lin, H.S., Eds. Toward a safer and more secure cyberspace. National Research Council, National Academies Press, Washington, D.C., 2007.
7. Landau, S. et al. *Codes, Keys, and Conflicts: Issues in U.S. Crypto Policy.* (ACM-sponsored study), 1994.
8. Neumann P.G. *Computer-Related Risks.* Addison-Wesley and ACM Press, 1995.
9. Neumann, P.G. Principled assuredly trustworthy composable architectures, final report. SRI International, 2004; http://www.csl.sri.com/neumann/chats4.pdf
10. Neumann, P.G. Fundamental trustworthiness principles in CHERI. In *New Solutions for Cybersecurity*, MIT Press, Cambridge, MA, 2018.
11. Schneider, F.B. and Blumenthal, M., Eds. *Trust in Cyberspace.* National Research Council, National Academies Press, 2101 Constitution Ave., Washington, D.C., 1998.
12. Shrobe, H. et al., Eds. *Solutions for Cybersecurity.* MIT Press, 2018.
13. Van Bulck et al. Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution. *USENIX Security* (Aug. 14–17, 2018); http://foreshadowattack.eu/
14. Watson, R.N.M. et al. Capability hardware enhanced RISC instructions (CHERI): Notes on the Meltdown and Spectre attacks. University of Cambridge Technical Report 916, 2017; http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-916.pdf
15. Weisse, O. et al. Foreshadow-NG: Breaking the virtual memory abstraction with transient out-of-order execution (Aug. 14, 2018); http://foreshadowattack.eu/.

Steven M. Bellovin (smb@cs.columbia.edu) is a professor of Computer Science at Columbia University, and affiliate faculty at its law school.

Peter G. Neumann (neumann@csl.sri.com) is Chief Scientist of the SRI International Computer Science Lab, and moderator of the ACM Risks Forum. Both Peter and Steven have been co-authors of several of the cited NRC study reports, and co-authors of *Keys Under Doormats*.

DOI:10.1145/3277567

R. Benjamin Shapiro, Rebecca Fiebrink, and Peter Norvig

▸ Mark Guzdial, Column Editor

Education

How Machine Learning Impacts the Undergraduate Computing Curriculum

The growing importance of machine learning creates challenging questions for computing education.

MACHINE LEARNING NOW powers a huge range of applications, from speech recognition systems to search engines, self-driving cars, and prison-sentencing systems. Many applications that were once designed and programmed by humans now combine human-written components with behaviors learned from data. This shift presents new challenges to computer science (CS) practitioners and educators. In this column, we consider how machine learning might change what we consider to be core CS knowledge and skills, and how this should impact the design of both machine learning courses and the broader CS university curriculum.

IMAGE BY METAMORWORKS

Thinking Like a Scientist, Not a Mathematician

Computing educators[1,6] have historically considered the core of CS to be a collection of human-comprehensible abstractions in the form of data structures and algorithms. Deterministic and logically verifiable algorithms have been central to the epistemology and practices of computer science.

With machine learning (ML) this changes: First, the typical model is likely to be an opaque composite of millions of parameters, not a human-readable algorithm. Second, the verification process is not a logical proof of correctness, but rather a statistical demonstration of effectiveness. As Langley[5] observed, ML is an empirical science that shares epistemological approaches with fields such as physics and chemistry.

While traditional software is built by human programmers who describe the steps needed to accomplish a goal (how to do it), a typical ML system is built by describing the objective that the system is trying to maximize (what to achieve). The learning procedure then uses a dataset of examples to determine the model that achieves this maximization. The trained model takes on the role of both data structure and algorithm. The role that each parameter plays is not clear to a human, and these computational solutions no longer reflect humans' conceptual descriptions of problem domains, but instead function as summaries of the data that are understandable only in terms of their empirically measurable performance.

To succeed with ML, many students will not concentrate on algorithm development, but rather on data collection, data cleaning, model choice, and statistical testing.

ML Education within CS Education

ML has historically been a niche area of CS, but now it is increasingly relevant to core CS disciplines, from computer architecture to operating systems.[3] It may even be fair to say that ML is now a core area of CS, providing a parallel theoretical basis to the lambda calculus for defining and reasoning about computational systems. The growing importance of ML thus raises challenging questions for CS education: How should practical and theoretical ML topics now be integrated into undergraduate curricula? And how can we make room for expanded ML content in a way that augments—rather than displaces—classical CS skills, within undergraduate degree programs whose duration must remain fairly static?

Changes to the Introductory Sequence. Most CS undergraduate programs begin with introductory courses that emphasize the development of programming skills, covering topics like control structures, the definition and use of functions, basic data types, and the design and implementation of simple algorithms.[4]

In many cases, assignments in these courses make use of existing library functions, for instance to read and write data to the filesystem. Students are not expected to fully understand how these libraries and the underlying hardware infrastructure work, so much as to use the interfaces that these libraries present. The aims of introductory courses are students' development of notional machines[2] for reasoning about how a computer executes a program, and the development of the pragmatic skills for writing and debugging programs that computers can execute.

ML has historically been a niche area of CS, but now it is increasingly relevant to core CS disciplines.

These same two aims can also describe introductory courses for an ML-as-core world. We do not envision that ML methods would replace symbolic programming in such courses, but they would provide alternative means for defining and debugging the behaviors of functions within students' programs. Students will learn early on about two kinds of notional machine—that of the classical logical computer and that of the statistical model. They will learn methods for authoring, testing, and debugging programs for each kind of notional machine, and learn to combine both models within software systems.

We imagine that future introductory courses will include ML through the use of beginner-friendly program editors, libraries, and assignments that encourage students to define some functions using ML, and then to integrate those functions within programs that are authored using more traditional methods. For instance, students might take a game they created in a prior assignment using classical programming, and then use ML techniques to create a gestural interface (for example, using accelerometers from a smartphone, pose information from a webcam, or audio from a microphone) for moving the player's character up, down, left, and right within that game. Such assignments would engage students in creating or curating training examples, measuring how well their trained models perform, and debugging models by adjusting training data or choices about learning algorithms and features.

Such activities do not require deep understanding of ML algorithms, just as reading from a filesystem using high-level APIs does not require deep understanding of computer hardware or operating systems. Yet these activities can introduce new CS students to epistemological practices core to ML, laying the foundation for encountering ML again in other contexts (whether an elective in ML theory, advanced electives in computer vision or architecture, or in professional software development). Such activities additionally enable the creation of new and engaging types of software (for example, systems that are driven by real-time sensors or social-media data) that are very difficult for novice programmers (and even experts) to create without ML.

Changes to the Advanced Core. In most CS degree programs, the introductory sequence is followed by a set of more advanced courses. How should that more advanced core change in light of ML?

Current courses in software verification and validation stress two points: proof of correctness and tests that verify Boolean properties of programs. But with ML applications, the emphasis is on experiment design and on statistical inference about the results of experiments. Future coursework should include data-driven software testing methodologies, such as the development of test suites that evaluate whether software tools perform acceptably when trained using specific datasets, and that can monitor measurable regressions over time.

Human-computer interaction (HCI) courses may be expanded to reflect how ML changes both the nature of human-facing technologies that can be created and the processes by which they are created and evaluated. For instance, ML enables the creation of applications that dynamically adapt in response to data about their use. HCI education currently emphasizes the use of empirical methods from psychology and anthropology to understand users' needs and evaluate new technologies; now, the ability to apply ML to log data capturing users' interactions with a product can drive new ways of understanding users' experiences and translating these into design recommendations. Future HCI coursework will need to include these ML-based systems design and evaluation methodologies.

Operating systems courses describe best practices for tasks such as allocating memory and scheduling processes. Typically, the values of key parameters for those tasks are chosen through experience. But with ML the parameter values, and sometimes the whole approach, can be allowed to vary depending on the tasks that are actually running, enabling systems that are more efficient and more adaptable to changing work loads, even ones not foreseen by their designer. Future OS coursework may need to include the study of ML techniques for dynamically optimizing system performance.[3]

Changes to Prerequisite and Concurrent Expectations. It is typical for CS curricula to require coursework outside of CS departments, such as courses in mathematics and physics. In many cases, and especially when CS programs are housed within schools of engineering, these requirements emphasize calculus coursework. Many programs include coursework in probability and statistics, though notably the authors of ACM and IEEE's joint Computing Curricula 2013 "believe it is not necessary for all CS programs to require a full course in probability theory for all majors."[4]

Are these recommendations still appropriate? Many programs require coursework in probability and statistics, which we enthusiastically encourage, as they are crucial for engaging with the theory behind ML algorithm design and analysis, and for working effectively with certain powerful types of ML approaches. Linear algebra is essential for both ML practitioners and researchers, as is knowledge about optimization. The set of foundational knowledge for ML is thus both broad and distinct from that conventionally required to obtain a CS degree. What, therefore, should be considered essential to the training of tomorrow's computer scientists?

Conclusion

The ACM-IEEE Computer Science Curricula 2013[4] identifies 18 different Knowledge Areas (KAs), including Algorithms and Complexity, Architecture and Organization, Discrete Structures, and Intelligent Systems. The definitions and recommended durations of attention to the KAs reflect a classic view of CS; ML is referred to exclusively within a few suggested elective offerings. We believe the rapid rise in the use of ML within CS in just the past few years indicates the need to rethink guiding documents like this, along with commensurate changes in the educational offerings of computing departments.

In addition, research on how people learn ML is desperately needed. Nearly the entirety of the published computing education literature pertains to classical approaches to computing. As we have mentioned earlier in this column, ML systems are fundamentally different than traditional data structures and algorithms, and must, therefore, be reasoned about and learned differently. Many insights from mathematics and statistics education research are likely to be relevant to machine learning education research, but researchers in these fields only rarely intersect with computing education researchers. Therefore, we call upon funding agencies and professional societies such as ACM to use their convening power to bring together computing education researchers and math education researchers in support of developing a rich knowledge base about the teaching and learning of machine learning.

References

1. Aho, A.V. Computation and computational thinking. *The Computer Journal 55*, 7 (July 2012), 832–835.
2. Boulay, B.D., O'Shea, T., and Monk, J. The black box inside the glass box: Presenting computing concepts to novices. *International Journal of Man-Machine Studies 14*, 3 (Apr. 1981), 237–249; https://doi.org/10.1016/S0020-7373(81)80056-9.
3. Dean, J., Patterson, D., and Young, C. A new golden age in computer architecture: Empowering the machine-learning revolution. *IEEE Micro 38*, 2 (Mar./Apr. 2018), 21–29.
4. Joint Task Force on Computing Curricula, Association for Computing Machinery, IEEE Computer Society (2013). Computer science curricula 2013; https://bit.ly/2E6dDGR
5. Langley, P. Machine learning as an experimental science. *Machine Learning 3*, 1 (Jan. 1998), 5–8.
6. Wing, J.M. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

R. Benjamin Shapiro (ben.shapiro@colorado.edu) is an assistant professor in the ATLAS Institute, the Department of Computer Science, and (by courtesy) the School of Education and the Department of Information Science at the University of Colorado, Boulder, USA.

Rebecca Fiebrink (r.fiebrink@gold.ac.uk) is a senior lecturer in the Department of Computing at Goldsmiths, University of London.

Peter Norvig (pnorvig@google.com) is Director of Research at Google, Inc.

DOI:10.1145/3192336

C. Liaskos, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz

Viewpoint
Using Any Surface to Realize a New Paradigm for Wireless Communications

Programmable wireless environments use unique customizable software processes rather than traditional rigid channel models.

WIRELESS COMMUNICATIONS HAVE undeniably shaped our everyday lives. We expect ubiquitous connectivity to the Internet, with increasing demands for higher data rates and low lag everywhere: at work, at home, on the road, even with massive crowds of Internet users around us. Despite impressive breakthroughs in almost every part of our wireless devices—from antennas and hardware to operating software—this demand is getting increasingly challenging to address. The large scale of research efforts and investment in the fifth generation (5G) of wireless communications reflects the enormity of the challenge.[1] A valuable and seemingly unnoticed resource could be exploited to meet this goal.

A common denominator in related research efforts is that the wireless environment—the set of physical objects that stand between two wireless communicating devices—remains a passive spectator in the data-exchange process. The ensuing effects on the data communication quality are generally degenerative: First, a transmitting device emits electromagnetic energy—carrying encoded information—which dissipates astoundingly fast within the environment. This path-loss phenomenon can be envisioned as distributing the same power over an ever-growing sphere. The power of the intended signal quickly diminishes, making its reception progressively more difficult. Second, as this ever-growing sphere reaches objects, such as walls, doors, desks, and humans, it scatters uncontrollably in multiple directions. This creates the multipath phenomenon where many, unsynchronized echoes of the original signal reach the receiver at the same time, making it difficult to discern the original. Third, the scattered signals naturally reach unintended recipients, increasing their noise levels (and allowing for eavesdropping). Finally, mobile wireless devices acquire a false perception of the frequency of electromagnetic waves, a phenomenon known as the Doppler effect. Notice that the hunt for higher data rates in 5G pushes for very high communication frequencies, at 60GHz for example, where the described effects become extremely acute.[1]

The concept of programmable wireless environments can impact wireless communications immensely.

This Viewpoint introduces an approach that could tame and control these effects, producing a wireless environment with software-defined electromagnetic behavior. We introduce the novel idea of HyperSurfaces, which are software-controlled metamaterials embedded in any surface in the environment.[6] In simpler terms, HyperSurfaces are materials that interact with electromagnetic waves in a fully software-defined fashion, even unnaturally.[2,7] Coating walls, doors, furniture, and other objects with HyperSurfaces constitutes the overall behavior of an indoor or outdoor wireless programmable environment. Thus, the electromagnetic behavior of the environment as a whole can become deterministic, controlled, and tailored to the needs of mobile devices within it. The same principle is also applicable to outdoor settings, by exemplary coating poles or building façades.

The concept of programmable wireless environments can impact wireless communications immensely, by mitigating—and even negating—path loss, multipath, and interference effects. This can translate to substantial gains in communication quality, communication distance, and battery savings of mobile devices, and even in security and privacy. Furthermore, due to the underlying physics, HyperSurfaces have no restriction in terms of operating communication frequency, which can extend up to the terahertz (THz) band.[10] This attractive trait makes HyperSurfaces potentially applicable to a variety of cutting-edge applications, such as 4G and 5G, Internet of Things (IoT), and device-to-device systems.

The Programmable Wireless Environment Concept

Consider an everyday communication scenario with multiple users within a physical environment, as shown in Figure 1. The common, non-programmable environment is oblivious to the user presence and their communication needs. The electromagnetic energy simply dissipates throughout the space uncontrollably, attenuating very quickly, causing interference among devices and allowing for eavesdropping.

In the unique programmable environments, HyperSurface-coated walls and objects become connected to the Internet of Things. As such, they can receive software commands and change their interaction with electromagnetic waves, serving the user needs in unprecedented ways. In the example shown in Figure 1, user A expresses a need for security against eavesdropping. The programmable environment, in collaboration with the user devices, sets an improbable "air-path" that avoids all other users, hindering eavesdropping. Users B, C, and D express no requirement, and are automatically treated by a global environment policy instead, which dictates the optimization of their data transfer rates. This can be attained by negating cross-interference and a minute crafting of the received power delay profile (PDP)—that is, ensuring all received wave echoes get constructively superposed at the devices. User F is observed to be inactive and—according to his preferences—has his device remotely charged by receiving a very focused energy beam. Finally, user E fails to pass the network's access policies (for example, unauthorized physical device address), and is blocked by the environment. This can be accomplished by absorbing his emissions, potentially using the harvested energy for constructive use.

From another innovative aspect, the programmable environment concept abstracts the underlying physics of wireless propagation, exposing a software programming interface to control it instead. Thus, the physics behind wireless propagation are brought into the realm of software developers, treating the electromagnetic behavior of objects with simple commands, as shown in Figure 2.

Essentially, the HyperSurface-coated objects are treated as "routers," which can forward or block electromagnetic waves in a manner very similar to the concept of routers and firewalls in wired networks. Connecting devices becomes a problem of finding a route connecting HyperSurfaces, subject to performance requirements and user-access policies.

Figure 1. Illustration of the programmable wireless environment concept. The electromagnetic behavior of walls is programmatically changed to maximize data rates (green use-cases), wireless power transfer (orange use-case), negate eavesdropping (purple use-case), and provide electromagnetic shielding (red use-case).

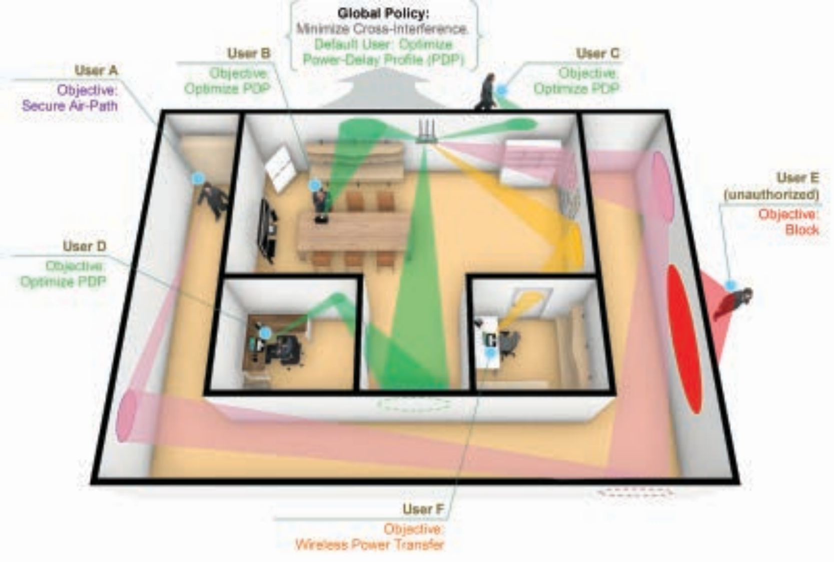


Figure 2. Software commands are combined and applied locally on walls to achieve the objectives of Figure 1.

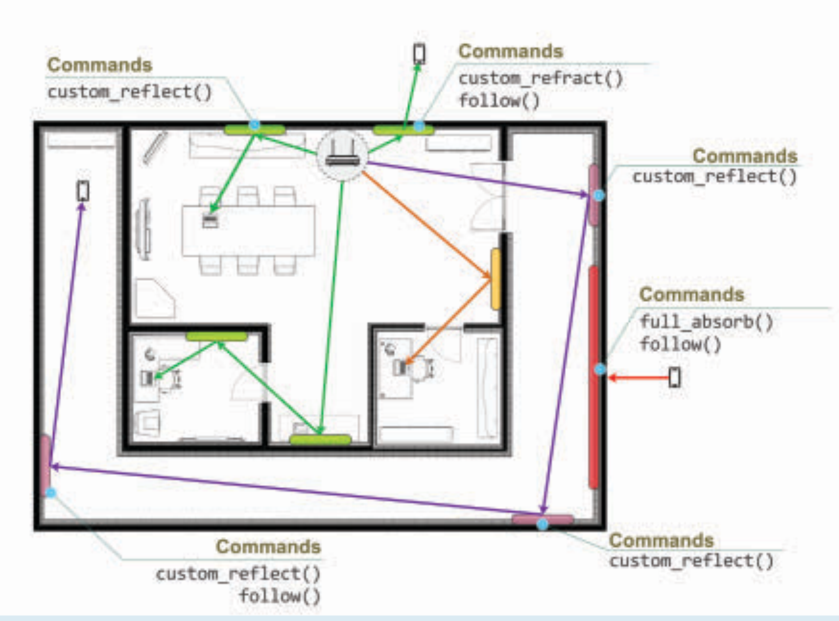
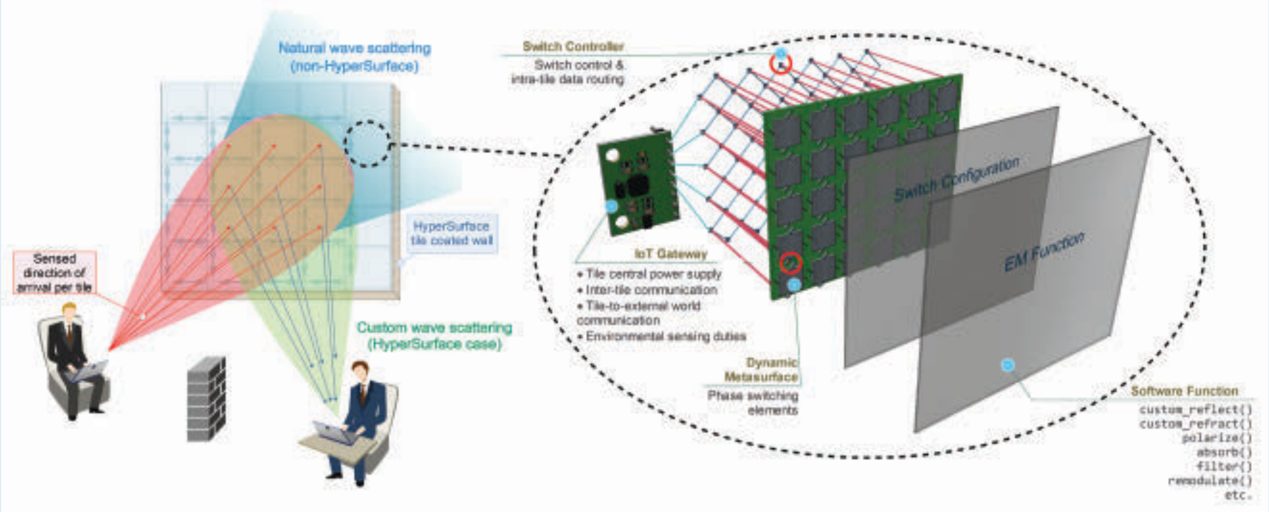


Figure 3. The proposed workflow involving HyperSurface-tile-coated environmental objects. The wireless propagation is tailored to the needs of the communication link under optimization. Unnatural propagation, such as lens-like focus and negative reflection angles, can be employed to mitigate path loss and multipath phenomena, especially in challenging non-line-of-sight cases. The main tile components are shown to the right in the figure.



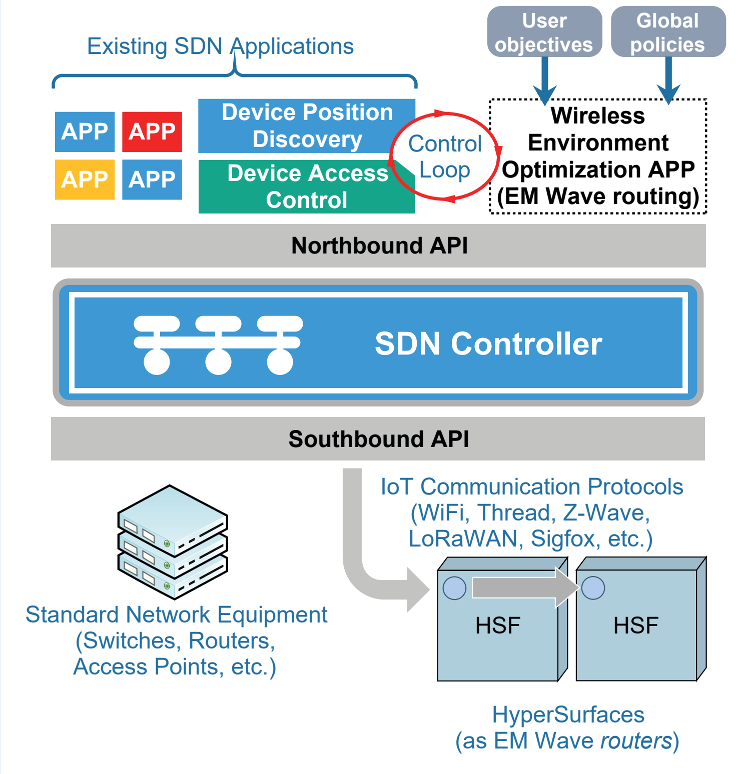
Structuring Programmable Wireless Environments

The main components that comprise a single HyperSurface tile and imbue it with control over wave propagation are shown in the right inset of Figure 3. Dynamic metasurfaces are the core technology for introducing programmable wireless environments. They constitute the outcome of a research direction in physics interested in creating materials with engineered electromagnetic properties. Most commonly, they comprise a metallic pattern, called meta-atom, periodically repeated over a dielectric substrate, and connected via switching elements.[5] The macroscopic electromagnetic interaction of a metasurface is fully defined by the form of the meta-atoms and the state of the switches. A certain state of switches may correspond to full absorption of all impinging waves from a given direction of arrival, while another may fully reflect them at an unnatural, completely custom angle.[2]

The translation of switch states to interaction types is performed by a novel software class: the electromagnetic compiler. The compiler is implemented by HyperSurface manufacturers and is transparently used by developers. In its simplest form, the compiler can be seen as a lookup table that keeps the best switch configurations corresponding to a set of electromagnetic interactions of interest. This table is populated by manufacturers, using well-known heuristic and analytical techniques in physics.[2,3]

Upon each tile there exists an IoT device that acts as its gateway. It exerts control over the HyperSurface switches, and allows for data exchange using common communication protocols (see Figure 4). Using these protocols, the tiles—and, thus, the coated objects—become connected to common networking equipment. Gateways of tiles upon continuous objects, such as walls, form a wired network to facilitate power supply and the dissemination of software commands. A selected

Figure 4. Integration schematic of programmable wireless environments in the SDN paradigm.


Existing SDN Applications
User objectives
Global policies
APP
APP
APP
APP
Device Position Discovery
Device Access Control
Control Loop
Wireless Environment Optimization APP (EM Wave routing)
Northbound API
SDN Controller
Southbound API
IoT Communication Protocols
(WiFi, Thread, Z-Wave, LoRaWAN, Sigfox, etc.)
Standard Network Equipment
(Switches, Routers, Access Points, etc.)
HSF
HSF
HyperSurfaces
(as EM Wave routers)

tile acts as the object's "representative," connecting to the external world.

Figure 4 illustrates the integration of the programmable wireless environment to common network infrastructure using the software-defined networking (SDN) paradigm.[1] SDN has gained significant momentum due to the clear separation it enforces between the network control logic and the underlying hardware. An SDN controller abstracts the hardware specifics ("southbound" direction) and presents a uniform programming interface ("northbound") that allows the modeling of network functions as applications. In this paradigm, HyperSurface tiles are treated as wave "routers," while the commands to serve a set of users, for example, as in Figure 2, are produced by a wireless environment control application. The application takes as input the user requirements and the global policies and calculates the fitting air paths. A control loop is established with existing device position discovery and access control applications, constantly adapting to environmental changes.

The scalability of the novel programmable wireless environments is a priority, both in software and hardware. In terms of software, the additional overhead comes from the optimization service, as shown in Figure 4. As described, however, the optimization pertains to finding objective-compliant paths within the graph of tiles, which is a well-studied and tractable problem in classic networking. In terms of hardware, the IoT gateway approach promotes miniaturization, low manufacturing cost, and minimal energy consumption of electronics, favoring massive tile deployments to cover an environment. Moreover, the choice of metasurfaces as the means for exerting electromagnetic control has distinct scalability and functionality benefits over alternatives. Metasurfaces comprise thin metallic elements and simple two-state switches, facilitating their manufacturing using large-area electronics methods (LAE) for ultra-low production cost.[9] LAE can be manufactured using conductive ink-based printing methods on flexible and transparent polymer films, incorporating simple digital switches such as polymer diodes.[9] On the other hand, alternatives such as antenna arrays[8] require transceivers with accurate state control and real-time signal-processing capabilities, posing scalability considerations in terms of size, power, and manufacturing cost.

Programmable environments provide a novel perspective for wireless communications.

Despite their simpler design, metasurfaces constitute the state of the art in range of wave interaction types, and with unique granularity. Advanced frequency filtering, polarization control, and arbitrary radiation-pattern-shaping functions can be potentially used for remodulating or "repairing" waves in the course of their propagation. Even in simple wave routing and absorbing functions, metasurfaces provide a degree of direction control so granular that it has been used for the formation of holograms.[4] A high degree of control granularity is required for 5G ultra-high frequency communications, as discussed earlier in this Viewpoint. Moreover, novel dynamic metasurface designs employ graphene, offering operation at the range of terahertz.[10]

Conclusion

The design and implementation of HyperSurfaces is a highly interdisciplinary task involving physics, material sciences, electrical engineering, and informatics. The combined expertise of all these disciplines results in significant value: programmable wireless environments can be enabled for the first time, allowing for programmatic customization of the laws of electromagnetic propagation, to the benefit of wireless devices. Programmable environments provide a novel perspective for wireless communications, where the usual rigid channel models are replaced by a customizable software process. Apart from unprecedented capabilities in wireless systems, this new perspective can pave the way for a completely new class of software applications, with rich interactions with existing security, device position discovery, and user mobility prediction mechanisms in the SDN world.

Recently, a related project—VISORSURF[a]—was funded under the prestigious Future Emerging Technologies call of the European Union Horizon 2020 framework. VISORSURF underwent a highly selective review phase, with a 3% acceptance rate, and attracted a total budget of 5.7 million euros. The multidisciplinary team of researchers is developing the hardware and software for the HyperSurfaces, expects to have the first prototype within a year, and begin mass production soon afterward.

[a] A hypervisor for metasurface functions; http://visorsurf.edu

References

1. Akyildiz, I.F., Nie, S., Lin, S.-C., and Chandrasekaran, M. 5G roadmap: 10 key enabling technologies. *Computer Networks 106* (2016), 17–48.
2. Chen, H.-T., Taylor, A.J., and Yu, N. A review of metasurfaces: Physics and applications. *Reports on Progress in Physics 79*, 7 (2016), Physical Society, Great Britain.
3. Haupt, R.L. and Werner, D.H. *Genetic Algorithms in Electro-Magnetics.* Wiley, NY, 2007.
4. Li, L. et al. Electromagnetic reprogrammable coding-metasurface holograms. *Nature Communications 8*, 1 (2017), 197.
5. Li, Y. et al. Transmission-type 2-bit programmable metasurface for single-sensor and single-frequency microwave imaging. *Scientific Reports 6* (2016).
6. Liaskos, C. et al. Design and development of software-defined metamaterials for nanonetworks. *IEEE Circuits and Systems Magazine 15*, 4 (2015), 12–25.
7. Lim, D., Lee, D., and Lim, S. Angle- and polarization-insensitive metamaterial absorber using via array. *Scientific Reports 6* (2016).
8. Moghaddam, S.S. and Moghaddam, M.S. A comprehensive survey on antenna array signal processing. *Trends in Applied Sciences Research 6*, 6 (June 2011), 507–536.
9. Parashkov, R. et al. Large area electronics using printing methods. In *Proceedings of the IEEE 93*, 7 (2005), 1321–1329.
10. Tassin, P., Koschny, T., and Soukoulis, C.M. Graphene for terahertz applications. *Science 341*, 6146 (2013), 620–621.

Christos Liaskos (cliaskos@ics.forth.gr) is a researcher at the Foundation of Research and Technology (Hellas), Greece.

Ageliki Tsioliaridou (atsiolia@ics.forth.gr) is a researcher at the Foundation of Research and Technology (Hellas), Greece.

Andreas Pitsillides (Andreas.Pitsillides@ucy.ac.cy) is a professor in the Department of Computer Science and the head of the Networks Research Laboratory at the University of Cyprus.

Sotiris Ioannidis (sotiris@ics.forth.gr) is a member of the staff at the Foundation of Research and Technology (Hellas), Greece

Ian Akyildiz (ian@ece.gatech.edu) is the Ken Byers Distinguished Chair Professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology in Atlanta, GA, USA, and member of the staff at the University of Cyrus.

This work was funded by the EU H2020 FETOPEN-RIA project VISORSURF: A Hardware Platform for Software-driven Functional Metasurfaces (GA 736876).

DOI:10.1145/3195179

Janne Lahtiranta and Sami Hyrynsalmi

Viewpoint
Crude and Rude?

Old ways in the new oil business.

EVER SINCE THE statement "Data is the new oil," widely credited to British mathematician Clive Humby,[a] was made in 2006, the world has experienced profound changes in the ways consumer-related digital information is used and accessed. Accordingly, it can be argued that current-generation smart devices and electronic services (including Internet service providers) are merely pieces of a pipeline in this new 'oil' business with a sole purpose of providing 'crude' for service providers to be processed and refined. In this sense, the metaphor coined by Humby (and others) is appropriate.

Taking the metaphor one step further, it must be acknowledged that the tycoons of this new 'oil' business are actually pretty smart when compared to Haroldson Hunt or Clint Murchinson, who were among the most notable businessmen during the Texas oil boom that took place in the U.S. in the early 20th century. Unlike in Texas, the 'land owners' (that is, consumers) of today pay for the gushers (smart devices), pipeline (Internet connection), and they even do the hard labor (use the services and applications).

Especially in Texas, handshake deals were a common form of an agreement—and legally binding in the eyes of the law. In today's oil business the handshake deals and notarized agreements are substituted by End User License Agreements (EULAs) and other forms of contract between the licensor and the purchaser. The problem with these agreements is that while they are in most cases legitimate and enforceable, they are ubiquitous and non-negotiated—millions of people bind themselves to the agreements every day with a simple click indicating "I agree."

The gravity of the current situation becomes even more evident if we investigate two particular technology trends; aggregation and personal health. Technology has a tendency of becoming smaller, smarter, faster, cheaper and a consequence of this is that our mobile phones have become aggregation points for different services, sensors, and data. In many cases, our phones are the only 'documentation' we carry with us when we go about our daily affairs in a bank or a grocery store.

Similarly our phones with their embedded sensors, and connected equipment (such as activity trackers), act as a singular entry point to data depicting our physical activities throughout the day (exercise, health rate, sleeping ...). Increasingly, our mobile phones are also integrated to our homes and personal transportation that are becoming smarter,[b] or at least more connected, every day. In other words, everything we regard as smart or connected in our everyday lives is becoming intertwined with our mobile phones.

In personal health, we are seeing how different mobile applications have become numerous. For example, if one has diabetes, there is an app for recording insulin levels. If one has a child with autism spectrum disorder, there is an app that can help in communication. In addition to apps, the health service providers are setting up virtual clinics, patient portals, and all kinds of "online health bazaars" that extend the reach of health services from hospitals to homes. In this, the mobile-first approach has become the prevalent one and our mobile phones act as primary access points for the services.

So when a consumer, unaware or sometimes even unconcerned, installs an app that simulates the function of a zipper (one can zip and unzip it, and that's all), or that of a stapler (one can tap a virtual stapler onscreen) and the application happens to have a malicious payload, it does not only compromise the security of the device—it may compromise the security of one's health affairs, housing, banking, business, and social life. In other words, it may compromise one's life.

Fortunately, this eventuality *is a theoretical one* as marketplaces are quick to react to malware, and services typi-

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

[a] This statement, and variations thereof, has also been credited to various authors, including Meglena Kuneva and Richard Titus; see https://bit.ly/2Mp7k9r

[b] Mikko Hyppönen from F-Secure prefers the term "vulnerable," which is also a pretty accurate definition.

cally have implemented extra layers of security (a side-channel for authentication, for example) in their function. However, one's digital life is vulnerable and currently employed measures do not live up to the task. Fair Information Practice Principles, and similar guidelines, such as those by the Markle Foundation, have had marginal impact on the situation, as has the current legislation. The impact of the General Data Protection Regulation (GDPR, EU 2016/679) that was put into effect in the EU in May 2018 remains to be seen.

Directives, regulations, and legislation are part of the solution. Another part comes from the application permissions of the operating system. In contrast to service-side solutions, these kinds of source-side solutions are more technological by nature, and linked to the function of the mobile phone. The source-side solutions are typically implemented as permissions: permission to access camera, or permission to use contacts information. Commonly, these permissions are requested from the user without pertinent information, and when a permission is granted, it is in effect indefinitely (or in some cases, until the next major update).

Looking at this kind of setting from the ethics side, one question emerges: Is this really an agreement? On one hand, the user has been given the terms of service, which can be 20,000-word document written in bulletproof legalese (as in the case of iTunes), and the user has clicked "I Agree." On the other hand, the user clicked "Allow" when the "beer drinking simulator" application requested permanent access to the camera, microphone, contacts, calendar, storage, location, and body sensors. This should be enough, right?

The land-lease agreements of the Oil Rush in general were notoriously tricky, promoting the authority of the more-aware landed. It was not rare that the lease itself, and the related royalty agreements, led to overdrilling in order to recover the investments of the oilmen. In these cases, the bias of the agreements was heavily one-sided, as is in the case of mobile applications (and electronic services) of today. However, the bias has tilted from more-aware 'landowners' to 'oilmen' who are now in the position of power, dictating the content of the agreements.

In the current end-user agreements, terms of service, and permissions the fundamental problem is that the consent is not informed. Consumers are expected to hand over the data they generate on the basis of obscure agreements and permissions that are defined on the basis of properties of an artifact—the high-tech smartphone of today that is millions of times more powerful than all of the NASA computers involved in the Apollo 11 mission to the moon in 1969.

What we need is a change of perspective and a different attitude. Similarly to the healthcare sector where the patient comes first, the consumer should come first in terms of user agreements, application-level permissions, and data use in general. In this, the healthcare principle of informed consent is of the essence. In healthcare the principle defines that the patient has bodily integration: autonomy and self-determination over one's physical body. In electronic services the principle should apply in such a fashion that the consumer has virtual integration: autonomy and self-determination over the 'digital self' (including data originating from one's activities and information stored in different devices and services).

Another central aspect in the informed consent of the health domain is that the patient must be sufficiently informed prior to making health-related decisions, such as undergoing a certain surgery. This principle is analogous to the virtual world; the consumer should be sufficiently informed prior to making decisions on the use of use of data that is part of the digital self. In this, the application permissions are part of the whole, as they are often the source of the data, or one way of communicating the data with the service provider(s).

Put in a more straightforward manner, this essentially means enforcing five things highlighted in different recommendations and in the GDPR: the service provider should make a clear case on what data is used; why it is used; how the data is collected; who has the access and what is the extent of confidentiality; and how long the data is accessible by the provider (and the named third parties). Naturally, should the premises change, the service provider's rights should be invalidated by default, and new permissions (informed consent) requested from the consumer.

Another step in the right direction would be to use colloquial language and terms familiar to the user. Instead of bulletproof legalese, the end-user agreement and terms of service should be stated in a way that reading them does not require advanced degrees in both law and computer science. This perspective on understandability also applies to the way application-level permissions are requested. Instead of requesting permission to "make/receive SIP calls" the consumer should be informed on what the SIP calls actually are, and where they are used.

This kind of consumer-centric approach to end-user agreement, terms of service, and application permissions would not only serve the purposes of informed consent—which is a goal in itself—it could also have a more profound impact on the use of technology. First, in terms of Internet and technology literacy, use of colloquial language could make the technology more visible and tangible as technological concepts would have a real name and meaning. Secondly, the users could become more privacy aware, as the data they generate and its use by the service providers, is portrayed in full.

In late-19th-century American businessmen who used shameless and even ruthless methods to get rich were often labeled as "robber barons." One of the most famous businessmen labeled as such was John D. Rockefeller, the founder of Standard Oil. Only time will tell if any of the major players in this new 'oil' business will receive a similar notorious title. Fortunately, there are already players in the field who understand the consumer comes first and regard privacy and trust as a competitive advantage instead of a cost or a nuisance. C

Janne Lahtiranta (janne.lahtiranta@turkubusinessregion.com) is a senior advisor at Turku Science Park Ltd in Turku, Finland.

Sami Hyrynsalmi (sami.hyrynsalmi@tut.fi) is an assistant professor in the Pervasive Computing department at Tampere University of Technology in Pori, Finland.

Intro | DOI:10.1145/3239530

Andrew A. Chien

Introducing *Communications'* Regional Special Sections

In my March 2018 editorial "Here Comes Everybody … to *Communications*," I announced an initiative to expand the *Communications of the ACM* community globally. I am pleased to introduce the first regional special section, which we hope will become a feature in *Communications* that you anticipate and enjoy, and of course value for the insights and perspectives it presents!

The theme for the regional special sections is, "Here Comes Everybody to *Communications*."[a] Why bring everybody to the magazine? Because the flagship publication of the world's leading computing professional society must include important voices and a variety of perspectives regarding the present and future of computing, regardless of where they may be found. Over the years, computing has grown into virtually every industry and every product, impacting every aspect of society and the economy of every nation; at the same time, the computing profession also has expanded into communities around the globe.

Computing's invention and continuing innovation is a global enterprise. While the technical foundations of computing may be universal, the design of many systems' most important aspects—how they relate to society, government, the structure of commerce, and individual enlightenment and perspective, as well as fundamental choices about security, privacy, free speech, and control—increasingly reflect distinctive regional, national, and community cultures.

Communications needs to include important voices and a variety of perspectives regarding the present and future of computing, regardless of where they may be found.

Communications should be an inclusive forum that spans the global community, with active participation from everyone, everywhere. The goal of *Communications'* global initiative is to bring untapped insights and focused coverage to our readers by providing highlights of computing from regions around the world. We will add a special section to a few issues of *Communications* each year, highlighting the leadership, unique characteristics, and distinctive development of computing in the region. The *Communications* global initiative will visit regions around the world in turn, shifting its spotlight to match the pace and impact of interesting developments in computing. We hope to revisit each global region about every two years.

Each of the Special Sections is led by a regional team that nominates, selects, and drives authorship of the section's content. We began building our team of industry and academic leaders for the China Region nearly one year ago. The team gathered at the University of Chicago Center in Beijing in March 2018 to brainstorm topics and form article writing teams.

To drive this and future regional coverage, we also have added new members to the *Communications* editorial board specifically focused on Special Sections. Led by Sriram Rajamani, they were actively involved in vetting and improving the articles in this inaugural China Region special section.

Thanks to all of the authors, the section leaders, Wenguang Chen (Tsinghua University), and Xiang-Yang Li (University of Science and Technology of China), and a special thanks to Lihan Chen, who created and drove the process that made the China Region special section a reality.

I hope you enjoy it! C

Andrew A. Chien, EDITOR-IN-CHIEF

Andrew A. Chien is the William Eckhardt Distinguished Service Professor in the Department of Computer Science at the University of Chicago, Director of the CERES Center for Unstoppable Computing, and a Senior Scientist at Argonne National Laboratory.

P.S. The next regional special section is already under way, focused on Europe. Look for it in early 2019!

[a] This title borrows from Clay Shirky's 2008 book *Here Comes Everybody: The Power of Organizing Without Organizations*, which described the growing power of groups of individuals to organize large-scale activities without relying on traditional corporate organizations.

ILLUSTRATION BY SPOOKY POOKA AT DEBUT ART. FOR CREDITS ON IMAGES IN COLLAGE, SEE P.2.

Alibaba.com
Alibaba.com

DOI: 10.1145/3239532

Welcome to the China Region Special Section

CHINA'S UNIQUE LANGUAGE, culture, governance practices, and research funding systems have had great impact on its Internet industry and technology development. For example, people in China seem less sensitive about privacy, which may be an important factor in the fast acceptance of mobile payment systems; combining that with the huge population of China could motivate many exciting technology innovations.

Some have speculated the Chinese government's strict supervision of Internet content and exclusion of some multinational competitors are important factors in the development of China's Internet industry, but it is difficult to assess the validity of such conjecture. In some non-content-based areas, such as e-commerce, the sharing economy, and Internet travel agencies, Chinese Internet companies have matched, and even surpassed, their international competitors.

For this special section, we invited contributors from a wide range of academic and industry communities spanning the Chinese mainland, Macau, and Hong Kong. We brainstormed article topics in a workshop in Beijing in March 2018. The response was terrific, and the resulting collection of articles, while far from comprehensive, offers an excellent snapshot of the most exciting computing trends and activities in the China region.

We are pleased to present the China Region special section, which includes:

▸ A series of short articles ("Hot Topics") that provide context and flavor of the region's distinctive growth, ranging from tech idols and computing culture to gaming, and

▸ Longer articles that document some of the "Big Trends" shaping the computing landscape of the China region, ranging from financial technology and last-mile autonomous delivery to SuperAI and cloud bursting.

— Wenguang Chen and Xiang-Yang Li
China Regional Special Section Co-Organizers

Wenguang Chen is a professor in the Department of Computer Science and Technology at Tsinghua University in Beijing, China, and co-chair of ACM China Council.

Xiang-Yang Li is a professor and Executive Dean of the School of Computer Science and Technology at the University of Science and Technology of China in Hefei, Anhui, China, and co-chair of ACM China Council.

Members of the China Region Workshop. Top left: Tong Zhang, Haibo Chen, Xiaoyang Wang, Hai Jin, Yuan Qi, Xiang-Yang Li, Xundong He, Wenguang Chen, Jing Xiao, Huaxia Xia, Liang Yu, Chaoyang Lu. Seated from left: Lihan Chen, Hong Gao, Andrew A. Chien, Yue Zhuge, and Yutong Lu.

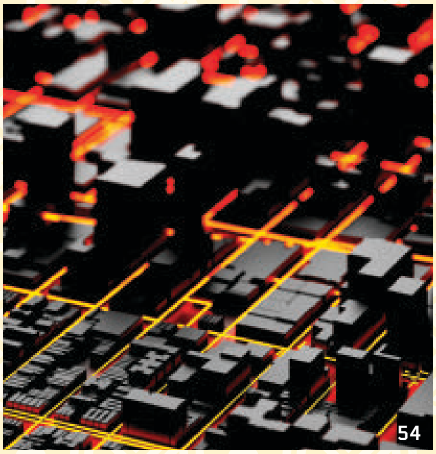
Watch the co-organizers discuss this section in the exclusive *Communications* video.
https://cacm.acm.org/videos/china-region

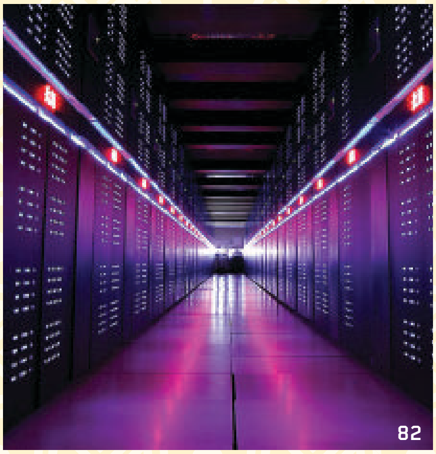
CHINA REGION

SPECIAL SUPPLEMENT

Hot Topics

Big Trends





(L) IMAGE BY YURCHANKA SIARHEI; (C) IMAGE BY TOSTPHOTO

China's Ambitions | DOI:10.1145/3239534

China's Computing Ambitions

BY ELLIOTT ZAAGMAN/Technode

CHINA PLANS TO become the world's high-tech leader, and quickly. In 2015, the Chinese government's State Council approved "Made in China 2025," an initiative designed to position China as a world leader in fields such as robotics, aviation, advanced information technology, and new-energy vehicles in less than a decade. In support of this governmental initiative, China's Ministry of Industry and Information Technology (MIIT) released a three-year action plan[a] to drive growth in areas including smart drones, facial recognition, AI-supported medical diagnosis, speech recognition, and language translation. If successful, the initiative would grow China's AI industry[b] to a size of $150 billion by 2020, approximately 100 times its size in 2016. As China pushes AI forward, here are a few names, trends, and technologies to watch.

Facial recognition technology used in Shenzhen, China, identifies jaywalkers and automatically issues fines by text.

IMAGE BY STREET VJ/SHUTTERSTOCK.COM

Facial Recognition and Surveillance

Earlier this year, an Alibaba-led funding of $600M made SenseTime the world's most valuable startup[c] at a valuation of $4.5B. SenseTime specializes in facial recognition technology with applications including payment verification[d] and automated checkout.[e]

Demand from China's public security agencies drives demand. "SenseTime ... can grow so fast compared to elsewhere in the world because video surveillance is a big deal in China ... there is a huge budget for it so they can manage society," explained Justin Niu of IDG, an

[a] https://bit.ly/2CFrMtZ
[b] https://bit.ly/2mrvxN4
[c] https://bit.ly/2GS1OJI
[d] https://tcrn.ch/2gDsG3X
[e] https://bit.ly/2mtlZ4b

early SenseTime investor. China's facial recognition firms are finding demand outside China as well. Shortly after news of SenseTime's massive cash influx, Cloudwalk, a company based in South China's Guangdong province, signed a cooperation agreement with Zimbabwe's government for a mass facial recognition project; the first foray of a Chinese AI firm into Africa, a growing focus for China's diplomatic, military, and financial resources.

Facial recognition[f] is just one dimension. This past March, the Guizhou provincial government, Tsinghua University, and Beijing-based d-Ear Technologies announced a pilot project intended to create a national database of "voiceprints" and link them to national ID information.

Pursuit of "Self-Reliance" in Core Technologies

In early 2018, the U.S. Department of Commerce placed a seven-year ban[g] on ZTE for violating Iran sanctions. Although the Trump administration and ZTE have since reached a deal to lift the ban,[h] the ban was a "wake-up call" for China's computing industry. With an estimated 25%–30% of ZTE's components sourced from U.S. firms, ZTE's case highlighted the dependence of Chinese tech firms' on others. Subsequently, Xi Jinping and other Chinese leaders have reiterated calls to achieve self-reliance[i] in "core technologies."

For example, heavy investment in the next generation of computing technologies may exceed that of the West (see the article by Y. Lu et al. on p. 82 of this section). First and foremost is quantum computing—it has been suggested the first general-purpose Chinese quantum computer could have a million times the computing power of all other computers currently on earth.

China aspires to global leadership. The $10-billion National Laboratory for Quantum Information Sciences[j] in Hefei, Anhui province, is expected to open in 2020. The laboratory, nearly four-million square feet, has two major research goals: quantum metrology and building a quantum computer. The laboratory also includes quantum communication and supports Chinese military efforts and commercial development (see article by C. Lu et al. on p. 42 for more information).

"Saudi Arabia of Data"

While China may lag in some core technologies, it has an unquestionable advantage in data (see the article by X. Li et al. on p. 50 of this section). With 772 million Internet users[k] and growing fast, China has approximately twice the number of users than the U.S.,[l] with data protections less stringent than the European Union. As Europe's GDPR regulations came into effect this past spring, so did China's Personal Information Security Specification (known as "the Standard"), called by Chinese observers a "business-friendly GDPR." Key differences include first intent, as Chinese regulators want to support development of AI that relies on access to massive datasets. And second, how user consent is defined, with exemptions that allow for data processing outside consent. Deeper analysis of "the Standard" can be found in a report from earlier this year.[m]

While China's top-down control risks government overreach, more open digital systems risk instability and have other weaknesses.

What Makes China Perfect for AI?

Around the world, concerns over job loss and privacy fears threaten to impede AI's progress and use. However, China may be the ideal environment to overcome these challenges. The first reason is demographic need. China's demography dictates a shortage of qualified people to support its aging population. Perhaps AI can fill those gaps. The second is that China's top-down political system can enable rapid, widespread application of innovative technologies, overriding commercial or popular opposition. Consider dockless bike-sharing,[n] for example, which grew from a novelty concept to a signature aspect of Chinese urban life in less than one year. In decentralized western systems, dockless bike-sharing typically requires independent approval from various government agencies; support from the Chinese Communist Party allowed China's dockless players to grow quickly. With driverless vehicles, there may well be a similar dynamic (see the article by H. Xia et al. on p. 70 of this section).

The fact that China does not offer its citizens the same legal protections or provide government or corporate transparency as Western democracies is a real concern. However, technology is something all societies must inevitably deal with, and how each manages it will reflect those societies' strengths and weaknesses. While China's top-down control risks government overreach, more open digital systems risk instability and have other weaknesses—exposed by recent "fake news" and conspiratorial election interference examples.

As China becomes a proving ground for AI technology, the world is paying attention. It may choose not to apply it in the same ways, but the world should certainly learn from it. C

Elliott Zaagman is a writer, speaker, and communications consultant focusing on technology, culture, and society in China. His work can be read on Tech in Asia, Technode, and in Chinese at Huxiu.com.

[f] https://bit.ly/2A08jay
[g] https://bit.ly/2uO3ATs
[h] https://reut.rs/2uxEXeu
[i] https://bit.ly/2uMdMM1
[j] https://bit.ly/2gtaV4s
[k] https://bit.ly/2DV0wrQ
[l] https://bit.ly/2qP6dFw
[m] https://bit.ly/2DV0wrQ

Quantum | DOI:10.1145/3239536

Quantum Communication at 7,600km and Beyond

CHAO-YANG LU, CHENG-ZHI PENG, AND JIAN-WEI PAN
University of Science and Technology of China

THE EXPONENTIAL GROWTH of the Internet and e-commerce shows the importance of establishing a secure network with global protection of data. Cryptography, the use of codes and ciphers to protect secrets, began thousands of years ago. While conventional cryptography methods predominantly rely upon mathematical complexity, their encryption can usually be defeated by advanced hacking.

IMAGE BY YURCHANKA SIARHEI

The idea of quantum cryptography, proposed by Bennett and Brassard in 1984[1] and by Ekert in 1991,[2] offered a radical, secure solution to the key exchange problem based on information theory, ensured by the laws of quantum physics. Quantum key distribution (QKD) allows two distant parties to produce a random string of secret bits, called a secret key.

Since the first table-top QKD experiment in 1989, a strong research effort has been devoted to achieving secure quantum cryptography over long distances, aiming at a global scale for practical use. To this end, there are two major challenges. First, quantum cryptography is ideally secure only when perfect single-photon sources and detectors are employed. Unfortunately, ideal devices never exist in practice and device imperfections have become the targets of various attacks. In 2007, Pan's group at the University of Science and Technology of China demonstrated the decoy-state QKD protocol to close the loophole due to imperfect single-photon sources.[5] In 2013, the same group demonstrated the first measurement-device-independent protocol that made the QKD immune to all hacking strategies on detection.[4] This work established secure QKD as a viable technology under realistic conditions.

The world's first quantum science satellite has now been combined with metropolitan quantum networks to form a space-ground integrated quantum network.

Since 2007, many intra-city and inter-city quantum communication networks have been built aiming for real-world applications. For example, Pan's team constructed metropolitan quantum communication networks in Beijing, Jinan, Hefei, and Shanghai in China, and connected these into the longest backbone line to date with a fiber distance exceeding 2,000km, on which real-world applications from banks, government, securities, and insurance industries are now on trial.

The second major challenge is long distance. For example, at 1,000km with a perfect GHz-rate single-photon source, ideal photon detectors, and telecommunications optical fibers (with a loss of 0.2dB/km), one would detect only 0.3 photons per century! One solution is quantum repeater protocols that divide the whole transmission line into *N* smaller segments, and combine the functionalities of entanglement swapping, entanglement purification and quantum storage. In spite of remarkable progress in demonstrations of the three building blocks and even prototype quantum repeater nodes, these laboratory technologies are still far from being practically applicable in realistic long-distance quantum communications.

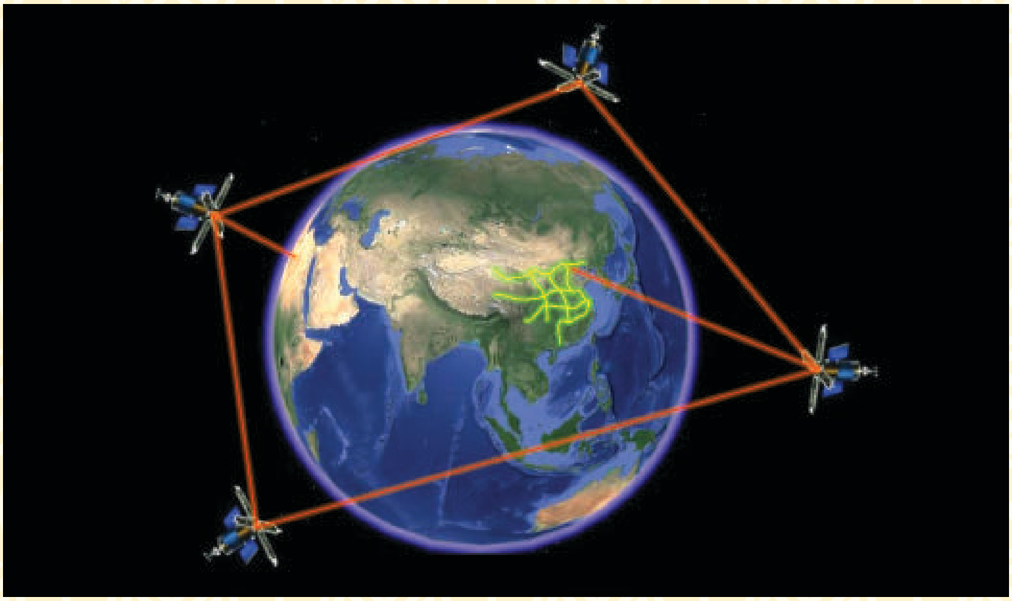
Satellite-based free-space quantum communication offers a unique and more efficient approach for global quantum networks. The key advantage of this approach is that the photon loss and turbulence predominantly occurs in the lower ~10km of the atmosphere, and most of the photons' transmission path is virtually a vacuum with almost zero absorption. A cross-disciplinary multi-institutional team of scientists led by Pan spent more than 10 years developing a sophisticated satellite dedicated to quantum science experiments.

Nicknamed Micius, the satellite was launched Aug. 16, 2016. Five ground stations in China connect with the satellite. Within a year of its launch, three key milestones were achieved: satellite-to-ground decoy-state quantum key distribution with ~kHz final key rate over a distance of 1,200km;[3] satellite-based entanglement distribution to two locations on the Earth separated by 1,200km with a two-photon count rate of 1Hz, and test of quantum nonlocality;[7] and, ground-to-satellite quantum teleportation over 1,400km.[6] The effective link efficiencies in the satellite-based channel were achieved to be ~20 orders of magnitude larger than direct transmission through optical fibers at the same length of 1,200km.

The world's first quantum science satellite has now been combined with metropolitan quantum networks to form a space-ground integrated quantum network, and has been further exploited as a trustful relay to conveniently connect any two points on Earth for high-security key exchange. On Sept. 29, 2017, intercontinental quantum communication between Beijing and Vienna at a distance of 7,600km was demonstrated, where secret keys based on the principle of quantum mechanics were used for the transmission of images and a videoconference.

Encouraged by the success of the Beijing-to-Shanghai backbone, similar quantum cryptography projects are being planned both in Europe and the U.S.

China will build new lines further to the South (Shanghai-to-Shenzhen), to the West (Beijing-Wuhan-Guangzhou) and to the North (Harbin-Changchun-Shenyang-Beijing). There is also a plan to launch, with both public and private funding, more low-Earth-orbit satellites in the near future to form a satellite cluster. In addition, a higher-orbit satellite is to be developed that aims to significantly increase QKD time, area coverage, and bandwidth. Encouraged by the success of the quantum science satellite and the Beijing-to-Shanghai backbone, similar quantum cryptography projects are being planned both in Europe and the U.S. The former recently kick-started a €1 billion Quantum Flagship project, and the latter committed $1.3 billion to a National Quantum Initiative in June 2018. C



A space-ground quantum network formed by China's quantum science satellite and metropolitan quantum networks.

References

1. Bennett, C. and Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, 175 (1984).
2. Ekert, A.K. Quantum cryptography based on Bell's theorem. *Physical Review Letters 67*, 661 (1991).
3. Liao, S.-K. et al. Satellite-to-ground quantum key distribution. *Nature 549*, 43 (2017).
4. Liu, Y. et al. Experimental measurement-device-independent quantum key distribution. *Physical Review Letters 111*, 130502 (2013).
5. Peng, C.-Z. et al. Experimental long-distance decoy-state quantum key distribution based on polarization encoding. *Physical Review Letters 98*, 010505 (2007).
6. Ren, J.-G. et al. Ground-to-satellite quantum teleportation. *Nature 549*, 70 (2017).
7. Yin, J. et al. Satellite-based entanglement distribution over 1200 kilometers. *Science 356*, 1140 (2017).

Chao-Yang Lu is a professor of physics at the University of Science and Technology of China, Hefei.

Cheng-Zhi Peng is a professor of physics at the University of Science and Technology of China, Hefei.

Jian-Wei Pan is a professor of physics at the University of Science and Technology of China, Hefei.

The Future of AI | DOI:10.1145/3239540

The Future of Artificial Intelligence in China

JUN ZHU/Tsinghua University, TIEJUN HUANG/Peking University,
WENGUANG CHEN/Tsinghua University, WEN GAO/Peking University

CHINA'S RESEARCH EFFORTS in artificial intelligence (AI) began later than the U.S. and Europe. Early contributions in the 1970s included automated theorem proving, logic reasoning, search, and knowledge engineering. For example, Wen-tsün Wu is a pioneer in automated theorem proving. He received the State Preeminent Science and Technology Award in 2000, an honor bestowed on only 25 Chinese scientists across all fields to date. Bo Zhang and Ruqian Lu received the Life Achievement Award from the China Computer Federation (CCF) for their fundamental contributions respectively on problem solving and knowledge engineering.

With the establishment of basic research funding to include AI research and development (R&D) in 1986, two agencies—the National Natural Science Foundation of China (NSFC), which supports basic research, and the 863 Program (State High-Tech Development Plan) for applied research—began funding diverse AI-related research topics, such as hardware and software for intelligence, human-computer interaction (HCI), intelligent application systems, neural networks, genetic algorithms, machine learning, natural language processing, computer vision, and robotics.

The first Smart China Expo was held last August in Chongqing and featured cutting-edge technologies from the fields of AI, robotics, 5G, and more.

In the late 1980s and 1990s, an emphasis on research in Chinese natural language processing took hold. Xuan Wang, a pioneer in applying AI to Chinese character printing and layout processing, became another recipient of the State Preeminent Science and Technology Award in 2001. He created the Founder Group, one of the largest computer companies on Mainland China in the late 1990s. Other AI-related companies started during that period include iFlyTek (Chinese voice synthesis and recognition), Hanvon (handwriting recognition), and TRS (Chinese full-text retrieval system).

After 2000, China's Ministry of Science and Technology (MOST), NSFC, other central government agencies, and local governments including Beijing, Shenzhen, and Hangzhou, increased funding tremendously to facilitate the new AI boom. The financial boost enabled Chinese researchers to attend international conferences and become deeply involved and integrated into international research communities. It is now common to see China's researchers attending top conferences, and their success includes having their research published extensively in leading AI conferences and journals, such as AAAI, IJCAI, ICML, NIPS, CVPR, ACL, PAMI, *Artificial Intelligence*, and more. For example, 23% of the accepted papers for the AAAI 2017 conference were from China, rising from only 10% in 2012.[a] For the IJCAI 2017 conference, nearly one-third of both submitted and accepted papers came from China.[b]

Important technical contributions from the region have been made in machine learning, computer vision, natural language processing, robotics, and more. For example, in machine learning, extensive work has been done on ensemble learning,[8] transfer learning,[4] artificial neural networks, evolutionary computing,[6] and probabilistic machine learning.[9] In computer vision, much progress has been made on Markov's random field modeling for image analysis,[3] handwritten character recognition,[1] facial recognition, and so on.[2] Finally, progress in AI hardware has enhanced accelerators for deep neural networks.[7]

With this increasing impact and recognition, more researchers from China have been invited to serve the community, in roles such as program chairs or area chairs for leading conferences, and as associate editors for top journals. For example, Qiang Yang from the Hong Kong University of Science and Technology (HKUST) is the president of the IJCAI Board of Trustees (2017–2019), and Zhi-Hua Zhou from Nanjing University is serving as a program co-chair for AAAI 2019. As the local community is growing fast, many top

[a] http://www.nber.org/papers/w24254.pdf

[b] https://bit.ly/2uOphTV

IMAGE BY HELLOABC/SHUTTERSTOCK.COM

conferences such as IJCAI, ICML, and ICCV are now held (or will be) in China.

Technical Giants Commit to AI Research

In addition to government-supported academic research, industry has also been very active in AI exploration.

China's technical giants, such as Baidu, Alibaba, Tencent, and Huawei, are actively investing in AI research and related development. These corporations have established their own worldwide AI labs, typically directed by world-renowned AI scientists such as Andrew Ng, who led Baidu Lab from 2014–2017. Moreover, these companies have branches throughout China, the U.S., and Europe.

AI research in industry labs is generally more business oriented. They focus on inventing and developing AI algorithms and systems to optimize not only their current businesses, such as online advertisement, payments, social networking, and gaming, but also new businesses such as smart city, healthcare, and auto-drive technologies. For example, Alibaba's ET Brain project uses AI to reduce traffic jams. It has been reported that traffic delays have been reduced by 15.3% by controlling 128 traffic signals in a select area of Hangzhou, where Alibaba's corporate headquarters is based. Moreover, ambulance response times in the same area were cut in half.

In addition to specific applications, corporate giants are also trying to build their own ecosystems. For example, Baidu launched DuerOS, a system that allows users to embed many AI functionalities, such as voice, natural language processing, and image recognition into devices. It also released open source platforms, such as Apollo for autonomous driving, and PaddlePaddle for deep learning.

International companies such as Microsoft, IBM, and Intel, also have built research labs in China with active AI research. They not only have very high-quality and impactful research, such as the Dual Learning theory proposed by Tieyan Liu et.al. at Microsoft Research Asia, but also feed China's AI industry with many high-quality researchers and technical managers.

Leading Startups Founded by Professors

The AI boom has given rise to smaller AI-focused companies, including Cambricon (AI chips), iFlytech (voice), SenseTime and MegeView (computer vision), and UBTECH (robotics). Researchers from academic institutions and universities founded many of these firms. Cambricon was founded by Tianshi Chen and Yunji Chen, both researchers at the Institute of Computing Technology, Chinese Academy of Science. They are pioneers in AI processor architecture and won the best paper awards at ACM's premier computer architecture conferences, ASPLOS and MICRO. Xiaoou Tang, a professor at the Chinese University of Hong Kong who has won best paper awards at top computer vision conferences like CVPR and ICCV, founded SenseTime—now the most valuable AI startup in the world, with a valuation of over $4.5 billion.

The Future of AI Research in China

AI research is young, but growing up fast in China. While still short on groundbreaking works and highly influential researchers, we are optimistic China's fast-growing economy and the aging population will drive strong demand for novel AI techniques, and ensure the successful future of China's AI research.

AI research is young, but growing up fast in China.

Recognizing the strong demand for AI, the Chinese government is planning support for AI education, research, and applications. In 2017, NSFC's Information Science Department reorganized its five information science areas (electronic engineering, computer science, automation, semiconductors, and optoelectronics) to incorporate the new sixth area, artificial intelligence.

The AI 2.0 proposal from the China Academy of Engineering[5] triggered the launch of a 15-year New Generation Artificial Intelligence Development Plan in July 2017. The plan is focused on a forward-looking blueprint for basic theories and common key technologies, including big data intelligence, swarm intelligence, cross-media intelligence, hybrid enhanced intelligence, and autonomous systems, and their applications in manufacturing, urbanization, healthcare, and agriculture, as well as AI hardware and software platforms, policies and regulations, and ethical concerns. Another R&D project related to AI is the so-called "Brain Science and Brain-Inspired Research," comparable to Europe's Human Brain Project, the BRAIN Initiative in the U.S., and other state-level projects. It is expected to be approved this year and should run for 15 years.

Last but not the least, a positive feedback loop between academia and industry has been established, which we believe will trigger more fundamental breakthroughs in AI in the future.

References

1. Ding, X. and Wang, Y. *Character Recognition: Principles, Methods and Practice.* Tsinghua University Press, 2017.
2. Gao, W. and Chen, X. *Computer Vision—Algorithms and System.* Tsinghua University Press (in Chinese), 1999
3. Li, S.Z. *Markow Random Field Modeling in Image Analysis.* Springer Science & Business Media, 2001.
4. Pan, S.J. and Yang, Q. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering 20*, 10 (2009).
5. Pan, Y. Heading toward artificial intelligence 2.0. *Engineering*, 2016,409–413.
6. Yan, P. and Zhang, C. *Artificial Neural Network and Evolutionary Computing.* Tsinghua University Press, 2005.
7. Zhang, S. et al. Cambricon-X: An accelerator for spare neural networks. In *Proceedings of the of the 49th Annual IEEE/ACM International Symposium on Microarchitecture*, 2016.
8. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms.* Chapman & Hall/CRC, Boca Raton, FL, 2012.
9. Zhu, J., Chen, J., Hu, W. and Zhang, B. Big learning with Bayesian methods. *National Science Review 4*, 4 (2017), 627–651.

Jun Zhu is a professor of computer science at Tsinghua University and director of TSAIL. Beijing.

Tiejun Huang is a professor in the School of EECS and chair of the Computer Science Department at Peking University, Beijing.

Wenguang Chen is a professor of computer science and technology at Tsinghua University, Beijing.

Wen Gao is a professor of computer science at Peking University, Beijing, and a Fellow of the Chinese Aademy of Engineering.

Consumers | DOI:10.1145/3239538

Consumers, Corporations, and Government: Computing in China

PETER GUY/South China Morning Post

UNIQUE HISTORICAL, SOCIOECONOMIC, and political conditions have created a distinctive path for China's rapid integration of computing and technology into its economy and society.

Over the last 20 years, major Chinese technology companies such as Baidu, Alibaba, Tencent, and Didi Chuxing have gained prominence through their ability to shape and deliver democratized computing power and services into the daily lives of China's consumers.

The rise of China's computing industry has transformed the way its citizens live and consume. Computing is a broad term that today encompasses a wide variety of industries and companies that utilize computing power through the Internet and personal computing devices. The critical mass of distributed computing power has reached new levels through smartphones, distributed computing, the Internet, and instant communication platforms like social media, which have spawned both large-scale innovations and challenges unique to China.

In the context of technology innovation and disintermediation, the number of Internet users in China grew 12% from 2015 levels to 717 million in 2016, while average hourly Internet usage grew 30% over the same period, providing both foundation and momentum to drive acceptance of a variety of online services, ranging from e-commerce to ride-hailing apps.

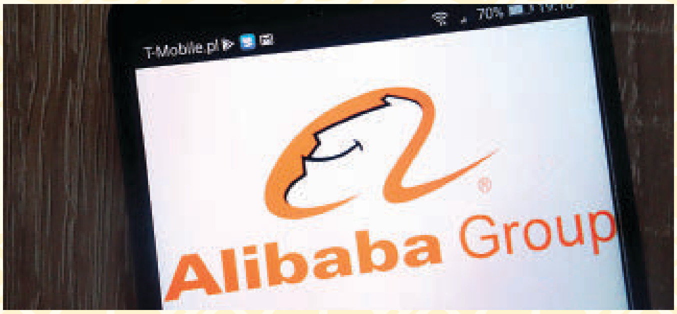
One development unique to China is the country's unprecedented rapid conversion to a nearly cashless consumer economy. While the technologies underlying this have been available for nearly a decade, the factors driving Chinese consumers to largely stop using cash include the country's relatively high smartphone usage, its weak traditional financial infrastructure (see the article by Y. Qi and J. Xiao on p. 65 of this section), and its large population.

In 2017, while China ranked 25th in the world in terms of smartphone penetration of its population (51.7%), due to its enormous population (roughly 1.38 billion in 2017, according to the CIA World Factbook), China had more smartphone users than any other country in the world in 2017 (717.31 million, followed by India with 300.1 million, and the U.S. with 226.2 million).

Today, payment apps such as AliPay and WeChat provide cashless payment services covering the full spectrum of daily life, including buying goods from street markets. As a result, even food trucks and street vendors often do not accept cash, and beggars are equipped to accept digital donations.

Another area distinctive to China is ride-hailing, a sector in which Uber, Didi Chuxing, and several other companies have engaged in a multiyear competition. Beijing-based Didi's rapid innovation of new services tied to local patterns and societal structure, as well as its hometown advantage, ultimately led to its August 2016 acquisition of Uber's China operation, a deal that propelled the global expansion and influence of the resulting $35-billion Didi.

Online innovation has propelled a cycle of growth in online users, and robust online usage is creating monetization for growth, which has encouraged substantial and growing investment at all stages of technological development. In 2017, China experienced record venture capital deal activity, with half of the top 10 largest deals globally involving Chinese telecoms and Internet companies according to Preqin, a source of data and intelligence for the alternative assets industry. Didi Chuxing's $5.5-billion financing in April 2017 emerged as the



biggest venture capital-backed deal of the past 10 years.

The Chinese are often portrayed as naïve, and even gullible, for allowing their government and technology businesses to collect so much data on them without their consent or knowledge. This has become an accepted observation of the unquestioned status quo in China.

However, since it was learned that Cambridge Analytica had mined the personal data of 87 million U.S. Facebook users (up from the initial estimate of 50 million) and used that data to affect the results of the 2016 U.S. presidential election, it appears private companies exploiting personal data for advertising and more proved to be less than ideal for protecting individual privacy.

Companies in China

Whether authoritarian state control or improved government regulation in the West will prove more effective in combating data-mining malpractice and protecting privacy remains to be seen. It is especially important as China's advanced computing ambitions continue to incorporate tight government control of content and a closed Internet firewall, a distinctly different model from that of the West's open Internet.

In 2016, Chinese technology company and search engine provider Baidu was investigated after a student researching a medical condition was directed to ineffectual, experimental medical treatment ads, which caused him to miss seeing genuine medical solutions.

Online entertainment in China is becoming the key driver of mobile time spent online, while e-commerce and games remain the best models for monetizing time spent online, according to data from venture capital firm Kleiner Perkins that showed the Internet accounted for 55% of total Chinese media usage in 2016 (up from 50% in 2015). Desktop Internet usage made up 37%, and mobile Internet usage 28%, of average daily media consumption in China in 2016. The growing amount of computing power in the hands of consumers supported the growth of online entertainment end-user revenue from $8 billion in 2011 to $31 billion to 2016.

The cashless digital economy arrived in China empowered by government regulations and cheap mobile devices, but citizens may not be aware of the profound implications on their daily lives and economic freedom. The dwindling importance of cash implies the wholesale elimination of bank runs. Banks can arbitrarily impose and charge fees and control funding transfers. The central bank can directly influence consumer behavior such as spending and saving, and the government can directly control the infrastructure required for basic actions (food, transport, work) essential for daily life.

China's advanced computing ambitions continue to incorporate tight government control of content and a closed Internet firewall, a distinctly different model from that of the West's open Internet.

President Xi's anti-corruption campaign has been made more efficient through the increasingly digitized economy and financial system, but it has also driven the shadow banking economy even deeper underground. According to Bloomberg, China's $15-trillion shadow banking industry (non-bank financial intermediaries that provide services similar to those of commercial banks, but outside normal banking regulations) threatens the stability of China's financial system, as $3.8 trillion of interest-bearing trust products, many sold online, threaten to default this year.

Do the Chinese understand the freedoms they are giving up in exchange for convenience and entertainment? Government power increasingly extends beyond privacy to direct tracking and control of the actions of daily life. This has not been fully debated in the Chinese media, which tends to celebrate the dominance of home-grown tech firms; nor has it been widely discussed in Western media.

Indeed, the development of computing in China is distinctive in many ways, and these differences present an opportunity to learn about the benefits and shortcomings of contrasting approaches.

Peter Guy is a business columnist for the *South China Morning Post* and co-founder and editor of *Regulation Asia*. He has an international background in venture capital and investment banking in Hong Kong and Guangzhou.

ALL IMAGES BY SHARAF MAKSUMOV/SHUTTERSTOCK.COM AND PIOTR SWAT/SHUTTERSTOCK.COM

Regional Culture | DOI:10.1145/3277554

Regional Computing Culture and Personalities

DESPITE THE GREAT Firewall, China's 772 million Internet users are adept at using smartphones for social media, live streaming, ordering home delivery, booking taxis, and sharing bicycles. Despite this massive market and bountiful opportunities for computing careers, Chinese corporations face great obstacles in attracting high-tech talent.

In April 2018, Chinese tech giants Baidu, Alibaba, and Tencent were called out for sexism in their recruitment campaigns.[a] These corporate computing leaders were criticized by the members of Human Rights Watch for advertising male-only jobs and for portraying women as "goddesses" to entice young male programmers to apply. The companies apologized and promised remedial action, but the flawed practice reflects a serious challenge in China: The world's most populous country does not have enough technologists to drive its lofty tech ambitions.

The average computer science doctoral graduate[b] in China earns 121,000 yuan ($19,000) a year, and those with AI skills can command 300,000 to 500,000 yuan. "There continues to be a massive need for talent in China," said Jerry Yang,[c] Yahoo co-founder. While plenty of jobs await computing graduates in China, some see their first or second employer as simply a steppingstone to starting their own company.

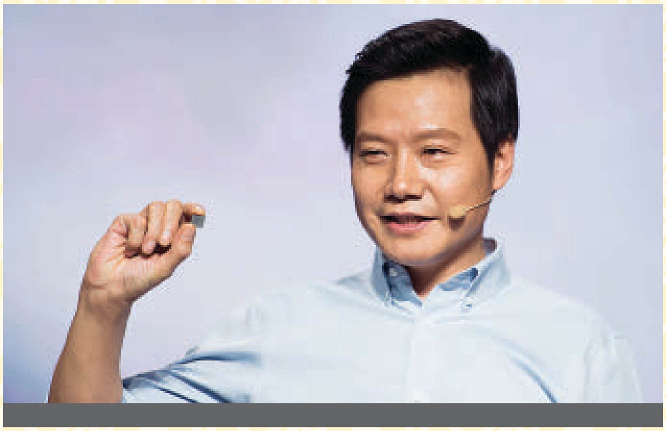
Lou Tiancheng has been called "one of the world's best hackers."

To address the technologist shortage, China aggressively recruits talent from abroad through government programs like "Thousand Talents," which through 2017 had attracted more than 7,000 top-level overseas Chinese scientists and engineers home with the promise of a 2 million yuan ($317,150) research grant, a 500,000 yuan "personal reward," as well as medical and housing benefits. Even major cities have entered the recruiting act, with Shenzhen spending 500 million yuan last year under its so-called Peacock Plan to attract overseas high-tech industry experts and academics.

Rock Stars

In China, tech geeks are rock stars, occupying the same limelight as teen idols. The late Stephen Hawking has a huge following in China, and Elon Musk is widely admired for his visionary work in electric cars and space travel. While some Chinese leading lights have achieved international fame, here we highlight four homegrown geek stars:

Lei Jun, CEO and co-founder of the smartphone and Internet services company Xiaomi, is a role model for young Chinese



Lei Jun, CEO and co-founder of Xiaomi, is a role model for young technologists.

LEI JUN PHOTO BY ZHANGJIN_NET/SHUTTERSTOCK.COM; PHOTO OF LOU TIANCHENG GIVING SPEECH IN EMTECH CHINA. SOURCE: MIT TECHNOLOGY REVIEW CHINA.

[a] https://www.hrw.org/report/2018/04/23/only-men-need-apply/gender-discrimination-job-advertisements-china

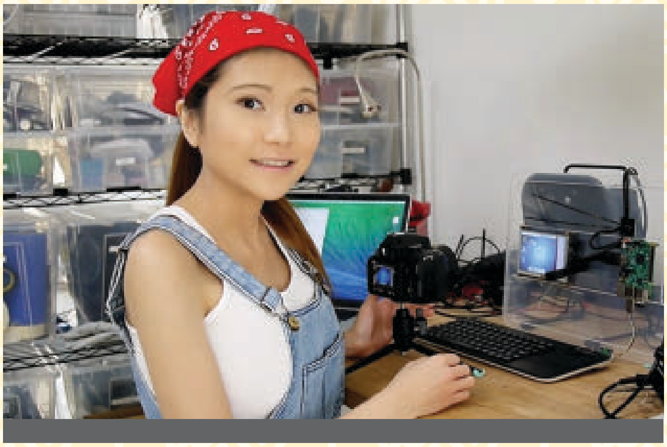
[b] 7http://www.chinadaily.com.cn/a/201806/02/WS5b11977e-a31001b82571dc75.html

[c] https://996.ggvc.com/category/podcast/page/2/

technologists hoping to strike it rich using their tech skills. Once referred to as the Steve Jobs of China, and derided for producing Apple copycat products, Lei has forged his own identity. Xiaomi expects to raise $6 billion in its initial public offering (IPO). To believe Lei, money is not the motivation to build Xiaomi—he has pledged to hold profit margins on hardware down to 5%, and return the surplus to users. Speaking for Xiaomi's founders, Lei has declared[d] "The spirit of engineering runs through our veins," and "All of us are hardcore fans of technology."

Lou Tiancheng, 32, has been called "one of the world's best hackers." He emerging as runner up at Topcoder Open in 2010 and racked up back-to-back wins at the Google Code Jam in 2008 and 2009. Lou left Baidu in December 2016 to co-found China's autonomous vehicle start-up Pony.ai, where he serves as chief technology officer. He is a role model for young Chinese techies interested in turning their coding skills into fame and fortune. While at Baidu, Lou advised[e] graduates not to think of programming as a shortcut to get into a better college or find a better job. "Learn to code for the fun of it," he said. "In the process, you will improve your reasoning and problem solving abilities."

China has 164 unicorns, which are private startups valued at more than $1B. The largest, with a valuation of $56B, is ride-hailing giant Didi Chuxing. Its president, **Jean Liu**, is a role model for Chinese women in tech. Born in 1978, the daughter of Liu Chuanzhi, the founder of China's computing giant Lenovo, she started the Didi Women's Network to help break what she calls the "mid-career bottleneck." Liu joined Didi in 2014 after a career at Goldman Sachs. She was inspired to study computer science at Peking University after reading Bill Gates' *The Road Ahead*.



Naomi Wu is on a mission to inspire women to work in the computing field.

Didi president Jean Liu was named one of *Time* magazine's "100 Most Influential People."

JEAN LIU PHOTO BY TOMOHIRO OHSUMI/GETTY IMAGES; NAOMI WU PHOTO COURTESY OF NAOMI WU/WIKIMEDIA CC-BY-SA-4.0

In 2017 Liu was named one of *Time* magazine's "100 Most Influential People," where Apple CEO Tim Cook called her a "disruptor." (Indeed, a year earlier, she convinced Apple to invest $1 billion in Didi). After acquiring Uber's China business, she was confronted with a new crisis recently when a female passenger was raped and murdered by a Didi driver, prompting the company to tighten up security and privacy measures to protect its female riders.

While Didi, Baidu and Xiaomi are all based in Beijing, China's Silicon Valley is in Shenzhen, a metropolis of 12 million people just north of Hong Kong. Shenzhen is home to some of China's biggest tech companies, including Huawei Technologies, ZTE, drone maker DJI, and Internet giant Tencent. Shenzhen is a talent magnet, and hosts an entire hardware electronics ecosystem, making it an ideal base for makers—DIY techies who dabble in electronics, robotics, and 3D printing, and more.

One extraordinary maker in Shenzhen is **Naomi Wu**,[f] an accomplished 20-something geek known as SexyCyborg to her YouTube and Twitter fans—most of whom are outside of China since both platforms are banned in China. Wu, who learned computer coding after finishing high school, is on a mission to inspire women to work in the computing field. "I'm not a rock star coder or anything," she told *Newsweek* in an interview in November 2017. "But I am good at cleaning up after rock star coders." Wu's DIY projects are mainly focused on wearable tech items for women.

With role models like Jun, Lou, Liu, and Wu, China's computing culture is vibrant and distinctive. C

[d] http://blog.mi.com/en/2018/05/03/open-letter-from-our-chairman/

[e] http://global.baidu.com/news-single/qa-with-top-coder-tiancheng-lou-baidu-autonomous-driving-team/

[f] http://www.atimes.com/article/meet-chinas-sexycyborg-goddess-geeks/

Data Trading | DOI:10.1145/3239542

Can China Lead the Development of Data Trading and Sharing Markets?

XIANG-YANG LI/University of Science and Technology of China,
JIANWEI QIAN/Illinois Institute of Technology, **XIAOYANG WANG**/Fudan University

At the same time the European Union is implementing new strict data protection regulations, China's data trading and sharing markets are booming. Here, we survey the status of these developing markets driven by growing demand from artificial intelligence (AI)-related industries, covering government encouragement as well as critical concerns and research opportunities including privacy and security.

China, with the world's largest e-commerce and mobile payment markets,[a] has an estimated big-data market of $70B circa 2015, which has been projected to grow to $155B by 2020.[2] As in much of the world, over 80% of data in China is privately held by the governments and private companies, restricting its exploitation for productivity and profit. The President of China, Xi Jinping, particularly emphasized the importance of data's open sharing and fusion as part of the national strategy for big data on December 8, 2017, encouraging data sharing across government sections and local governments, and data sharing/trading between governments and private companies.

Data exchange markets in China.

Data Exchange Platform	Date Est.	URL
Global Big Data Exchange (Guiyang)	Dec 2014	http://www.gbdex.com
East Lake Trading Center for Big Data	Jul 2015	http://www.chinadatatrading.com
Jiangsu Big Data Exchange	Nov 2015	http://www.bigdatahd.com
Chongqing Big Data Trading Market	Oct 2015	http://www.crazyapi.org
Shanghai Data Exchange Corp.	Apr 2015	https://www.chinadep.com
Qingdao Big Data Exchange	2017	http://www.qddata.com.cn
Zhejiang Big Data Exchange	Mar 2016	http://www.zjdex.com
Harbin Data Exchange	June 2016	http://www.hrbdataex.com
Central China Data Exchange	July 2016	http://www.ccbde.cn
Qiantang Data Exchange	Dec 2015	http://www.qtjiaoyi.com
Data Tang	June 2011	http://www.datatang.com
You-e Data	Dec 2015	http://www.youedata.cn
Lei Ju	May 2011	http://www.leiju.cc
Fa Yuan Di	Sept 2015	http://www.finndy.com
JD Wan Xiang	Unknown	https://wx.jcloud.com
BDG Store	Nov 2016	http://www.bdgstore.cn
Ali Data Market	Sept 2009	https://market.aliyun.com/chn/data
Baidu Data	Unknown	http://apistore.baidu.com
Big Ocean	Unknown	http://www.dahaiyang.com
Markway Mall	2001	http://www.markwaymall.com

The founder of Fa Yuan Di Ltd. said the market size of data trading in China was approximately $3.2 billion in 2016. The market size is estimated to grow to $8.7B by 2020.[1] Examples of data trading and sharing include:

- Didi Chuxing became the winner in the fiercely competitive ridesharing industry with the help of big data from Wechat, the dominant app for messaging, social media, and mobile payment;
- Vanke Real Estate utilizes big data provided by China Mobile to find the best locations for investment; and,
- Ping An Insurance takes people's online behavior data from Baidu, Tencent, and Alibaba to accurately pinpoint potential customers and create new insurance products.

Twenty data markets have been established by various local government authorities and private enterprises in China (see table at left), trading whole datasets, Web crawlers, APIs, and analytical results. They can be traded either off-the-rack or via customization, and come from many directions including banking, energy, health care, transportation, industry, agriculture, tourism, education, telecommunication, and much more. One of the largest is the Global Big Data Exchange in Guiyang with over 2,000 corporate members and more than 150PB of reported stored data circa March 2018.[b]

Policies and Environment

Following the U.S. and other countries, the Chinese policy encourages governments to share data to enhance transparency and efficiency. The State Council Guidelines for Promoting Big Data Development, released in 2015, proposed the goal of establishing a united platform for open government data by the end of 2018. The top priority is to share data from several important realms, including credit, transportation, and health care. The Chinese government also encourages private data trading to expand the digital ecosystem. There are no regulations specialized for data sharing and trading yet, except those aiming at protecting national security, trade secrets, and copyrights, and banning the

[a] https://mck.co/2GXOoqS

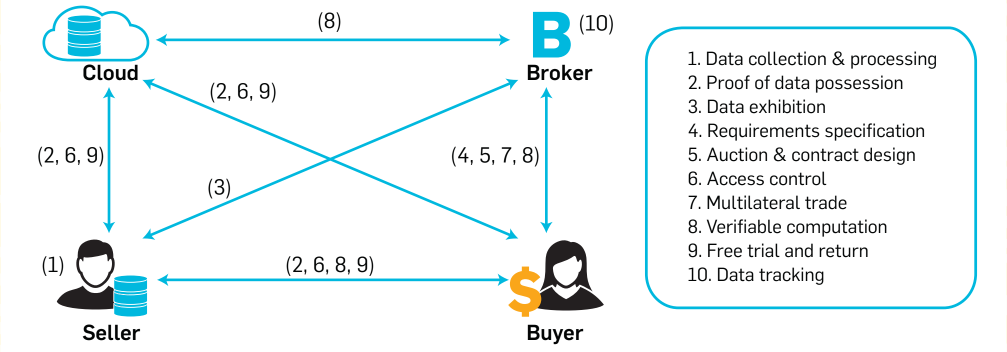
[b] https://bit.ly/2LqeREB

propagation of illegal content (such as terrorism, fake news).[2] Lack of regulation allows experimentation, but some data owners hesitate to share their data due to potential legal or business consequences. There have been public discussions about sharing and trading of personal data, which is openly traded. While the EU's General Data Protection Regulation (GDPR) governs companies collecting, processing, and selling their consumers' data, China has no national regulation on data protection; only fragmented regulations exist, like the Cyber Security Law and the Personal Data Infringement Interpretation that came into effect in June 2017.[2]

A GfK survey[c] indicated 38% of people in China are very willing to share personal data for better service, whereas the ratio is 25% in the U.S. and lower than 20% in most European countries. However, awareness and concerns are growing in China. In March 2018, a survey conducted jointly by China Central Television and Tencent Research indicated 76.3% of 8,000 Chinese people interviewed were worried about the threat AI posed to their privacy. A few days later, Baidu's CEO said Chinese people are willing to sacrifice privacy in exchange for convenience, triggering an enormous public backlash.[d] Compared to the EU and the U.S., however, China's market is a lenient, more permissive environment for data trading and sharing.

Concerns and Opportunities

Data markets are growing in China, but are still immature with most datasets small in scale, poor in quality, and low in value. Critical concerns inhibit data exchange, as depicted in the figure here.



A general workflow of data exchange.

Preprocessing. To increase usability, sellers must preprocess data cleaning, labeling, reconciliation, fusion, and desensitization, which require automation for big data. Manual approaches are still common. Behind China's booming AI industry are almost one million data labelers: mostly rural, part-time workers.[e]

Pricing. Data products have unlimited supply because of the little marginal cost, causing the Arrow-Debreu equilibrium price to be close to zero.[5] While early attempts have explored data pricing,[4,6] it is still an open problem. Today, standard whole datasets are sold at fixed rates, and customized data is priced by negotiation. The pricing strategies for APIs include pay-as-you-go and wholesale.

Security. Data is vital to the information asymmetry between different companies and government sections. Inappropriately sharing may reduce ability to compete, expose wrongdoings, or harm their public images. Most data is sold through API, but many sellers are worried the buyers can infer data content. Utilizing techniques such as query auditing can ensure data security and alleviate sellers' concerns. Data transactions should also be tracked to achieve accountability.[3] Many platforms, such as Global Big Data Exchange (Guiyang) and JD Wan Xiang, have adopted blockchain to strengthen trading security because of its favorable decentralized and tamper-resistant qualities.

Privacy. Before being traded, data must be desensitized to protect personal information and privacy. Sellers should also take into account potential privacy leaks from temporal, spatial, and different owners' data linkage. Although China lacks strict regulations like the GDPR, most trading markets in China claim to desensitize personal data. For instance, Shanghai Data Exchange Corp. protects personally identifiable information through encryption and encoding when they perform data linkage.

Verifiability. There are cases where the traded data was forged, producing distrust. When sellers list their datasets on the marketplace, they must prove to the trading broker their ownership of the data, the data's authenticity and accessibility, and that the data content and quality are as claimed. In addition, the proof should not disclose the data content. When buyers purchase the API of a dataset, sellers must prove the API was correctly computed over this exact dataset—extremely difficult if buyers also want query privacy.

Conclusion

While these challenges are daunting, the strong government encouragement and rich data collection enable rapid growth of large-scale data exchanges in China. With technology advances (blockchain, AI, big data analytics, cloud computing) and maturing policies (privacy, digital ethics), we are optimistic that better data sharing and trading ecosystems will help China's economy transition to a new level of global competitiveness.

References

1. Global Big Data Exchange (Guiyang). White Paper on China's Big Data Exchange. 2016.
2. Institute of Information and Communication of China. Big Data White Paper. 2018.
3. Jung, T. et al. Accounttrade: Accountable protocols for big data trading against dishonest consumers. In *INFOCOM* (2017), 1–9.
4. Li, C., Li, D. Y., Miklau, G., and Suciu, D. A theory of pricing private data. *Commun. ACM 60*, 12 (Dec. 2017), 79–86.
5. Quah, D. *Digital Goods and the New Economy.* 2003.
6. Zheng, Z. et al. An online pricing mechanism for mobile crowdsensing data markets. In *MobiHoc* (2017), 26.

Xiang-Yang Li is a professor and Executive Dean of the School of Computer Science and Technology of China, Hefei.

Jianwei Qian is a Ph.D. candidate of computer science at the Illinois Institute of Technology, Chicago.

Xiaoyang Wang is a professor and dean of Computer Science at Fudan University, Shanghai.

c https://bit.ly/2K2E0zt
d https://ab.co/2uUL4JJ
e https://bit.ly/2NRtxZV

Psychology | DOI:10.1145/3239544

Exploiting Psychology and Social Behavior for Game Stickiness

LUYI XU/NetEase Fuxi Lab

CHINA'S GAME MARKET is the largest worldwide, recording $32.5 billion in sales in 2017. However, the lack of copyright protection (common in the European Union and U.S.) has forced China's game developers in the offline PC game industry to pursue Web and mobile-based game development. Today, few developers in China focus on developing PC or console games.

Of the 507 million gamers in China as of June 2017 (an estimate by the China Audio-Video and Digital Publishing Association and CNG Games Research Center in the China Gaming Industry Report January–June 2017), the number of console game owners (an estimated 9.09 million, according to global gaming research firm Newzoo in 2016) and offline PC gamers (18.5 million, based on data provided by Steam in 2017) together account for less than 10%.

The Chinese video game market boom has been driven by escalating quality, attracting gamers with amazing in-game graphics, music, stories, level design, and gameplay, but increasingly players are attracted by the desire for socialization. Games give players what they don't have in real life—the happiness of making friends, a feeling of superiority in their skills, the feeling they are "part of something." As a result, interactive video games like the relatively new *PlayerUnknown's Battlegrounds* (PUBG), which support various forms of collaboration and confrontation among gamers, create an obsessive stickiness for Chinese users.

Games give players what they don't have in real life—the happiness of making friends, a feeling of superiority in their skills, and the feeling they are "part of something."

Promotion and marketing of video games increasingly relies on socialization. Tencent, a conglomerate with greater revenues in 2017 ($21.9 billion) than the combined revenues of Activision Blizzard ($7.07 billion), Electronic Arts ($4.845 billion), Ubisoft ($1.704 billion), and Take Two Interactive ($1.779 billion), touches tens of millions of users via its Web portal QQ and mobile chat service WeChat, and attracts them to its online games. For example, players are drawn to Tencent's multiplayer online battle game *Arena of Valor* through the company's operation of the *League of Legends* multiplayer online battle arena game on the PC platform.

Socialization's prominent role in Chinese video games has attracted special attention from computer scientists. Several news publications labeled 2017 the "Year of Artificial Intelligence," marking a transformation in human-computer interaction. A good example of this can be seen in NetEase.

NetEase, Inc., founded in 1997, is a leading Internet company providing online services centered on content, community, communications, and e-commerce. It entered online gaming in 2001,

IMAGE BY HAFIEZ RAZALI/SHUTTERSTOCK.COM

and develops and operates large online games that accounted for 60% of its 2017 revenue of $5.3 billion, making NetEase the sixth-largest game company in the world.

NetEase architects in-game social systems as a core competitive advantage. The company cultivates player loyalty by constructing in-game social networking systems in which players collaborate and compete just like they do in the real world, through their avatars.

Yet gamer presence and immersion in virtual worlds is crucially tied to the intelligence of in-game non-player characters (NPCs). In conventional games, most NPCs stick around certain areas and mechanically repeat conversations and behaviors when prompted, providing limited interaction. If NPCs can have sophisticated logic, responding spontaneously to player input, a game's explorative and immersive experience could reach a whole new level.

With artificial intelligence advancing rapidly, NetEase is investing aggressively to improve in-game social interaction between human and machine. NPC-based artificial intelligence (AI) was first incorporated in offline role-playing games (RPGs), such as the Nemesis System that enabled the game *Middle Earth: Shadow of Mordor* to stand out, as it allowed enemies in the game to actually evolve by slaying the player's character, resulting in their growing into dangerous adversaries.

From the earliest massively multiplayer online role-playing games (MMORPGs), social interaction has formed an intrinsic part of online game play. MMORPGs have been designed as virtual worlds that are socially realistic, achieving immersion based on diversity of interaction. When *World of Warcraft* hit the market in 2004, that game's designers at Blizzard Entertainment had already started to leverage "reputation" to enrich interactions between players and NPCs (Figure 1).

New game releases by NetEase add more granularity to NPCs' interactions with players. In popular multiplayer online role-playing games, players interact with several characters as groups, and reputation determines how their interactions with each faction play out. In *Justice Online* (Ni Shui Han) and *The Legend of Chu Liuxiang* (Figure 2), among other online RPG titles, players' avatars interact with NPCs as individuals. Each character behaves within a framework of dimensions including personalities, emotions, and health. The states of players' avatars and NPCs are influenced by their history and actions, which affect how they interact with each other. For example, an NPC with a growling stomach might decline a martial arts contest with other players; those with evil morals may find it difficult to improve righteous NPCs' approval ratings; and some of the NPCs will give gifts to and team up with players they favor. Such interactions are not new to users in offline RPGs, but are still innovative to developers of online games, and have proven popular with players.

Figure 1. The NPC interactive interface on *Nishuihan Online* (still in development from NetEase).

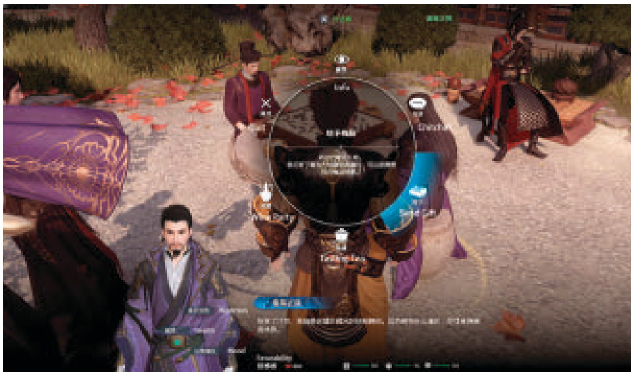
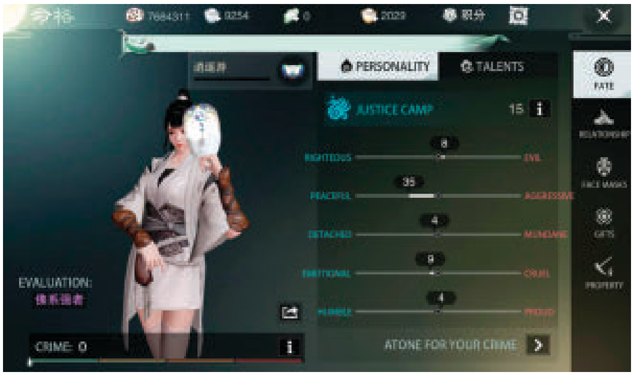


Figure 2. The personality traits of a player in *The Legend of Chu Liuxiang*, recently released by NetEase.



In future games, NetEase plans to incorporate personified chatbots behind NPCs and speech synthesis, enabling diverse and even spoken interaction. Another conceptual project at the company is the creation of voice-based games in which the player's commands and directions are spoken, and several intelligent characters can behave in complicated ways and respond to situations independently.

To accomplish those goals, the company is exploring the use of reinforcement learning to train AI in ways including behaviors associated with any kind of language. NetEase developers also are working to improve the facial expression and body language of NPCs using facial expression transfer, motion transfer, and emotion recognition from text for 3D facial expression rendering. As a result, NPCs may become sufficiently human-like for gamers to have sentimental interactions with them, and even relationships.

NetEase is a trailblazer of in-game human-machine social interaction. Although human-machine social interaction is new to the market, we believe artificial intelligence will grow into a major driver of games that meet players' social-psychological needs. C

Luyi Xu is a senior systems specialist at NetEase Fuxi Lab, Hangzhou.

big trends

DOI:10.1145/3239546

BY WANLI MIN/ALIBABA CLOUD COMPUTING,
LIANG YU/ALIBABA CLOUD COMPUTING,
LEI YU/CTRIP, **SHUBO HE**/CTRIP

People Logistics in Smart Cities

CITIES IN CHINA are growing rapidly in terms of both size and complexity. Governments have been searching for new technologies to make cities more efficient, and smart mobility has been the top priority in all solutions.

The past few years have seen a paradigm shift for smart mobility in China, that is, data-centric companies, mostly Internet companies, are taking a leading role in such initiatives instead of governments and academic researchers. For example, Alibaba, Baidu, Tencent, Ctrip, and Didi, among others, are spearheading the smart mobility initiatives. The driving force is twofold: these companies have accumulated a huge volume of data and invested a great deal of resources in the AI arena. Their main focus involves AI systems able to predict city traffic conditions with full spatiotemporal coverage and optimize transportation systems accordingly.

The blossoming of smart mobility initiatives is due to the fact the potential of big city traffic data has not been fully mined. Researchers are still pursuing a better way to break the data silos while also preserving privacy. Fortunately, we have seen significant efforts from both industries and governments in China to promote data sharing for innovations.

Here, we will focus on the smart mobility scenarios that are representative for cities in China. We will elaborate on two aspects: in-city and intercity transport. For the in-city scenario, we take Alibaba's City Brain program as an example to introduce how the big city data can be used to optimize the traffic signal scheduling and accelerate access for life-saving emergency vehicles (EV). For the intercity scenario, we focus on the weekend/holiday crowdedness unique to China, and introduce how big tourism data can be leveraged to divert tourists to more suitable attractions to avoid traffic congestion.

Real-Time and Holistic Situational Awareness on City Traffic

High-quality, real-time, and holistic traffic condition sensing is the prerequisite of further transport optimization. To overcome the limit of spatiotemporal coverage of traditional sensor data, the data from active navigation apps, including private cars and taxis, are employed to rebuild the trajectories using map-matching techniques, which can cover almost the entire road network of a city. Trajectory data can be used for various purposes, such as traffic parameter extraction (speed/volume) or origin-destination (OD) analysis. Various traffic indices can be generated based on these parameters, such as delay index that measures the travel time delay comparing to no-congested situation, and imbalance index that measures the speed difference between upstream and downstream road links. By monitoring the indices,

IMAGE BY TOSTPHOTO

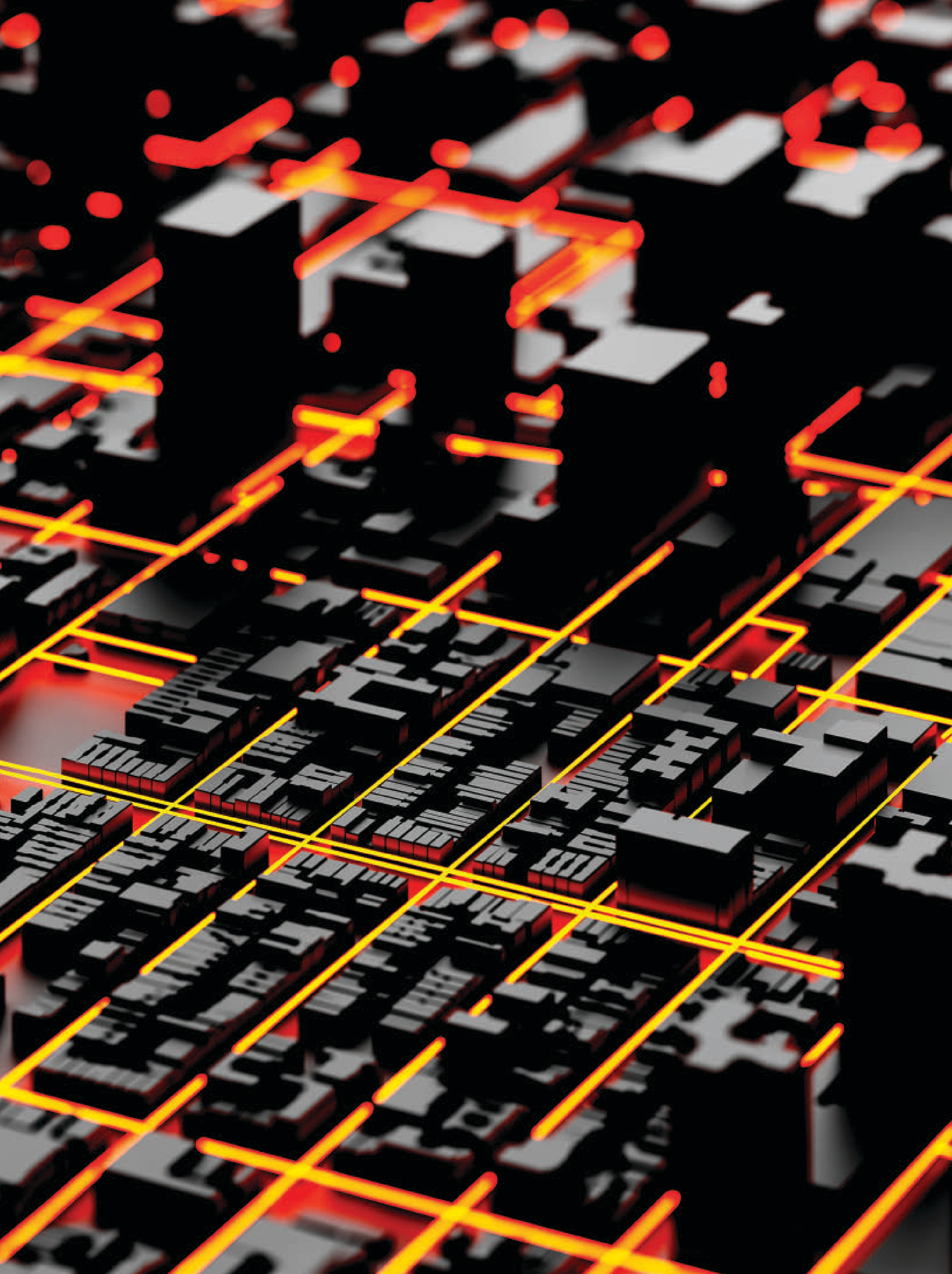


Figure 1. Incident detection and tracking in Hangzhou city brain.

The system is able to recognize more than 10 types of incidents and track the related vehicles crossing multiple cameras.

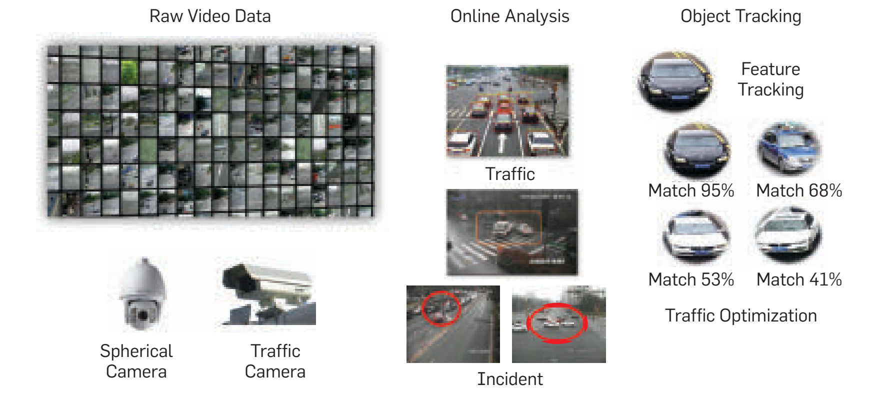
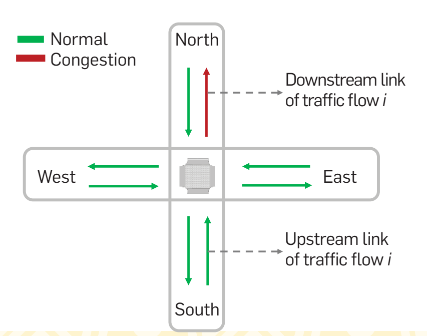


Figure 2. Intersection with four entrances.


Normal
Congestion
North
Downstream link of traffic flow *i*
West
East
Upstream link of traffic flow *i*
South

abnormal traffic conditions can be detected and altered for attention in real time.

Navigation data is in the hands of the leading map/transport service providers in China, for example, AutoNavi (Alibaba), Baidu, Tencent, and Didi. All of these companies have participated in smart mobility initiatives. For example, in the City Brain project, AutoNavi[a] provides near real-time traffic data services (traffic parameters are updated with two-minute intervals) for traffic sensing and optimization purposes.

Another important type of data is CCTV. Though not quite new as a data source, it is currently undergoing an AI transformation. There are tens of thousands CCTVs deployed in each big city in China for traffic surveillance. Traditionally, they only perform snapshot capturing rather than analytical jobs. It still requires human intervention to review and interpret current traffic conditions. It is a very tedious and error-prone job and becomes even more challenging as the number of CCTVs continues to increase.

Thanks to advancements in computer vision technology and elastic cloud computing, it is possible to empower computers to do the more analytical work usually handled by humans.[2] Alibaba's City Brain project adopted a cloud-based architecture to stream the large volume of video data into the cloud, where it is processed in parallel to generate structured results such as speed/volume, queue length, and incidents. The results can be used in three ways: data fusion and cross-validation with other data sources, for example, speed/volume; input for other applications, for example, queue length can be used by traffic signal optimization; and, incident detection/tracking that greatly improves the efficiency of traffic surveillance and emancipates human labor (see Figure 1).

AI-based CCTVs can also be used in many other scenarios such as security and policing. For example, if a car runs from the accident scene, the cameras can collaboratively track it in real time; face recognition technology can be used for CCTV data—even though a human face is typically a very small and vague camera image difficult for humans to recognize, the computers can still achieve a stronger view. Many unicorn start-up companies have emerged in this area, such as SenseTime,[b] Hikvision,[c] Dahua[d]

About 80% of the serious traffic congestions are caused by accidents. Quick response to accidents and anomalies is a very effective way to improve traffic conditions. Moreover, once multiple data sources are integrated to generate a holistic view of the city traffic, it not only improves the performance of the traffic management, but also enables further applications that can systematically use this data to optimize the city transport.

Self-Adaptive Traffic Signal Optimization

The traffic signal is one of the most important means of directing city traffic. The dominant traffic signal systems run at a cycle/split-based scheme, that is, each split corresponds to a phase within which only certain directions are allowed. Then the problem is how much (green) time to allocate to each phase. Existing signal systems either run at fixed timing schedules, or rely on the fixed loop detectors to do self-adaptive scheduling. However, loop detectors or speed cameras are fixed at certain locations and the data is considered nearsighted, while the root cause of a traffic jam might originate from a long distance. As holistic traffic sensing is now possible by fusing multiple data sources, there are a few new ideas for optimizing traffic signals.

The first idea is to balance the traffic condition of upstream and downstream road. As depicted in Figure 2, for each driving direction, for example, east to north, there is an upstream link and a downstream link. The imbalance value of each driving direction is defined as the difference of the normalized speed (actual speed over free speed) from upstream and downstream road links.

$$d_t^t = \frac{v_{u_i}^t}{\hat{s}_{u_i}} - \frac{v_{d_i}^t}{\hat{s}_{d_i}}, \Delta_t^t = \beta d_t^t$$

$$\arg\min_{\delta} \sum_{l=1}^{n} \Big(w_t^t(\Delta_t^t = \sum_{J=s_1}^{n} \delta_j^t)\Big)^2, A \leq \sum_{J=1}^{m} \delta_j^t \leq B \quad (1)$$

Given there are m signal phases, n driving directions, and the many-to-many relations between them; Equation 1 specifies the objective function for the optimization. The goal is to find the best incremental green time allocation for the m signal phases $\delta_j, j \in \{1,2,\ldots, m\}$. $v_{u_i}^t$

[a] www.amap.com. AutoNavi is one of the largest web mapping, navigation and location based services providers, founded in 2001 and acquired by Alibaba Group in 2014.

[b] https://www.sensetime.com/

[c] http://www.hikvision.com/cn/

[d] https://www.dahuasecurity.com/

and $v_{d_i}^t$ are respectively the actual speed of the upstream and downstream roads of driving direction i at time t, $\hat{S}_{u_i}$ and $\widehat{S_{u_i}}$ are the free speed; d_i^t is the imbalance value for driving direction i at time t, and Δ_i^t represents the expected incremental green time for direction i where β is a hyperparameter(s). The objective is to minimize the total (volume weighed sum) difference of the real allocation and the expected allocation of all driving directions, where w_i^t is the volume of direction i at time t, and s_i is the set of all related phases of direction i. Finally, A and B are used to ensure the total cycle time of the new schedule is within an acceptable range, which are also hyperparameter(s) in signal systems.

Another idea is based on the partition-and-conquer paradigm that applies to large-scale optimization, for example, a city or district. One of its applications is the so-called greenwave, which means multiple traffic signals are coordinated to reduce the number of stops. The coordination is achieved by setting appropriate phase difference for two traffic signals with the same cycle time, so that a vehicle traveling at normal speed can drive through the next traffic signal without stop. Greenwave is normally applied to arterial roads where there is a large volume of traffic crossing consecutive traffic signals. The key of conducting greenwave is to identify the route that can maximize the performance gain (normally the route with the maximal volume), which can be identified from trajectory data.

The underlying philosophy of the arterial road greenwave is its portability to any randomly shaped area. Figure 3 shows the result of the traffic signal partition in Huangpu district of Shanghai city. Navigation trajectories can be used to derive the volume data between each pair of traffic signals, which are used as the input of the network partition algorithms.[1,5] The result is a good suggestion for further optimizations. For example, for the in-group coordination, a very simple idea is to rank the adjacent traffic signal pairs by their connectivities and set appropriate phase differences one by one; for the intergroup coordination, we focus on the traffic signals on the boundary, and take into consideration the overall traffic conditions in each group.

More and more cities in China have benefitted from such efforts including Hangzhou, Suzhou, Guangzhou, Shanghai, and Wuhan. Take Hangzhou as an example: its City Brain system is processing a large volume of data, including one million+ trajectories, 2,000+ cameras' video streams and many other traditional sensor data. It reports around 2,500 events daily with 95% accuracy. The average travel time of all trips in the city is reduced by 15.3%.

Reduction of response time from the field test in Hangzhou City's Xiaoshan District.

Time	Normal(s)	Optimized(s)	Gain (%)
9:00–10:00	150	101	32.67
10:00–16:30	150	96	36.33
16:30–18:00	2017	154	25.85

On-Demand Greenwave for Emergency Vehicles

The response time of emergency vehicles (EV) is critical to saving lives. Governments across the globe set ambitious response time targets. The National Health Service (NHS) of the U.K. set a target of eight minutes for most serious medical calls.[e] New York City mandates a 10-minute response time on emergency calls.[f] In Singapore, in 87.1% of cases, an EV arrives within 11 minutes.[g] As the last few years have seen rapid urbanization in China, the demand for faster EV response times continues to rise.

Functionally, there are two basic approaches to reducing response time: optimize the route for EVs to avoid traffic, obstacles, and any other risks; and, preempt traffic signal systems to allow EVs to pass swiftly through intersections. Both approaches still remain challenging: the estimated time of arrival (ETA) used by routing algorithms can be delayed by ever-changing traffic conditions, and the signal preemption must be dynamic and precise according to the traffic conditions to avoid negative impact on the overall traffic flow.

The time-dependent vehicle-routing problem (TDVRP) has long been researched.[4,6] Traditional time-varying path searching algorithms are too optimistic: vehicles are expected to drive exactly at the predicted speed. In reality, the actual travel time at each individual road link can slightly vary from expected

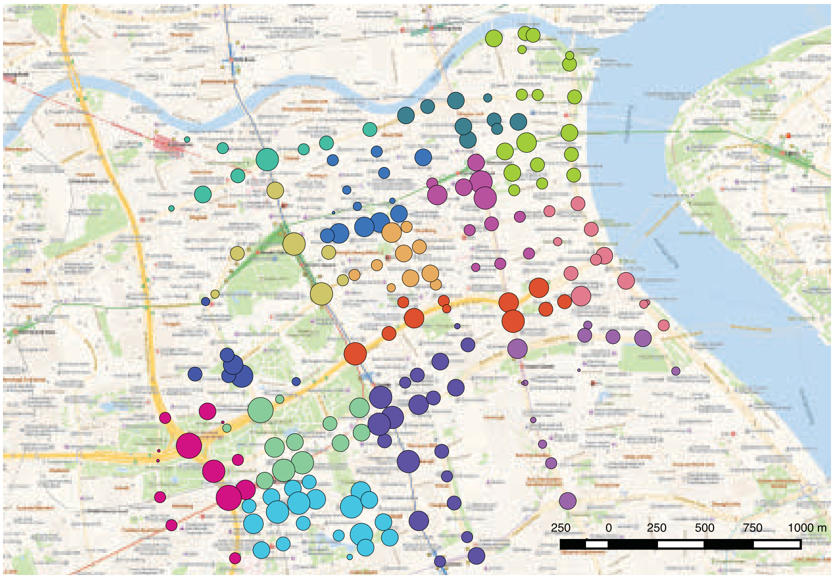
[e] https://www.nao.org.uk/wp-content/uploads/2017/01/NHS-Ambulance-Services.pdf

[f] http://www.nytimes.com/1990/03/25/nyregion/new-ems-response-time.html

[g] https://www.scdf.gov.sg/sites/www.scdf.gov.sg/files/EMS%20Stats%202016.pdf

Figure 3. Enhancing traffic signals.

In Huangpu district of Shanghai city, 188 traffic signals are partitioned to 15 groups based on their connectivity—the traffic volume that connects one traffic signal with another. Each traffic signal junction is sized by its traffic volume and rendered by its group color. This process helps traffic engineers to optimize traffic signals more efficiently.



Navigation trajectories can be used to derive the volume data between each pair of traffic signals, which are used as the input of the network partition algorithm.

values. As illustrated in Figure 4, the speed prediction has a significant variance, which indicates a variable speed for actual driving. Cumulatively, this can lead to a large difference between the ETA and the actual arrival time. The higher the variance between arrival and ETA, the higher the risk for real people in critical conditions. Therefore, the question is: How best to plan a route that is fast and robust on ETA?

An improved route-searching algorithm can answer that question. Instead of trying to minimize merely the overall travel time, the variance of ETA is also taken into consideration.

$$\arg\min_{j \epsilon N} \left(\mu(t^j_{n_j}) + \alpha\sigma\left(t^j_{n_j}\right)\right) \qquad (2)$$

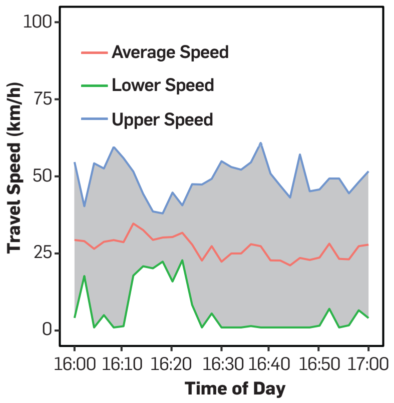
Equation 2 is the revised objective function for selecting a path j from totally N candidates to minimize the weighted sum of mean (α) and standard deviation (σ) of travel time. A path p_j is represented by a sequence of nodes $\{v^j_1, v^j_2, \ldots v^j_{n_j}\}$ where n_j is the number of nodes for path p_j, and α is the weight, which is a hyperparameter. t^j_i is the arriving time at the i^th node of path j, and thus $t^j_{n_j}$ is the ending time of path p_j.

The key to solving the equation is to calculate the distribution of ending time. Let us assume a simple case: Traveling from node A to B: given the arriving time distribution at A, a time-varying speed function on edge AB, and an random perturbation imposed on the speed (a speed offset follows a normal distribution with mean 0), to compute the arriving time distribution at node B. Once this is solved, the whole searching algorithm can use it in an iterative way, that is, from t^j_i to $t^j_{n_j}$ where t^j_i is a fixed value. This problem can be modeled as a continuous Markov process.

As the EV travels along the planned route, it constantly communicates with the control center and shares its location and speed (by GPS devices). The control center fuses the real-time feedbacks with the historic data to predict the ETA at the next traffic signal junctions, and inform the signal control system to prioritize the EV's driving direction. The key challenge to this task is twofold: How to determine the most appropriate timing to start the green signal to clear the residual vehicle queue before the EV arrives; and, how to minimize the impact on opposite driving directions.

Figure 4. Exemplar speed observation generated by AutoNavi.

The three curves represent respectively the mean speed (Red), mean speed +3 standard deviation (SD) (Blue), mean speed -3 SD (Green).



The residual queue length is defined as the length the vehicles fail to pass the junction in one cycle. Video analytics is one way to detect the queue length, and trajectory data can also be used to estimate the queue length where cameras are missing. Once the queue length is determined, the control system can gradually allocate extra green time to clear the queue before the EV arrives.

To minimize the negative impact introduced by the signal preemption, the algorithm dynamically searches for a optimized solution that balances the overall green time allocated to each phase, rather than simply dwelling on the target phase and causing problems to other directions. This problem is modeled as a mixed integer programming problem, which aims at smoothing the change of signal scheduling by starting the preemption as early as the ETA's variance is limited to a certain range.

Our test in Xiaoshan District of Hangzhou city has shown a significant improvement in EV travel times, as illustrated in the accompanying table. This test is conducted on a route from (30.138384 120.280503)(lat/lon) to (30.186592 120.266079) where there are 19 traffic signals.

Tourism Recommendation to Solve the Holiday Crowdedness

During public holidays in China, popular tourist cities are flooded with large numbers of visitors that can swell to several times the number of residents. Increased needs for accommodation, food, and entertainment exert extreme pressure on the local environment and public services, especially for transportation. Take the China National Day (an annual weeklong holiday beginning October 1) for example: In 2015, visitors to Huangshan mountain spent nine hours on average waiting in line. In 2016, more than 25 million tourists visited Chongqing city—a western metropolitan city whose residential population is 30 million. In 2017, the traffic congestion on the Hukun expressway was accumulated to maximally 49.73km. Such overcrowded populations, as we know, lead to problems like pollution, congestion, and loss of open spaces, and causes inconvenience and negative experiences for both tourists and local residents.

As the largest online travel agency in China, Ctrip discovered an insight from its big data—that is, there is an imbalanced situation between the distribution of tourists and the collective capacity of attractions. The agency envisioned that a good recommendation system could help divert tourists to less-crowded attractions to resolve the problem. The basic idea is to build a tourist prediction component, and once an attraction is predicted as overcrowded, a recommendation component will be triggered to try to divert tourists to other places.

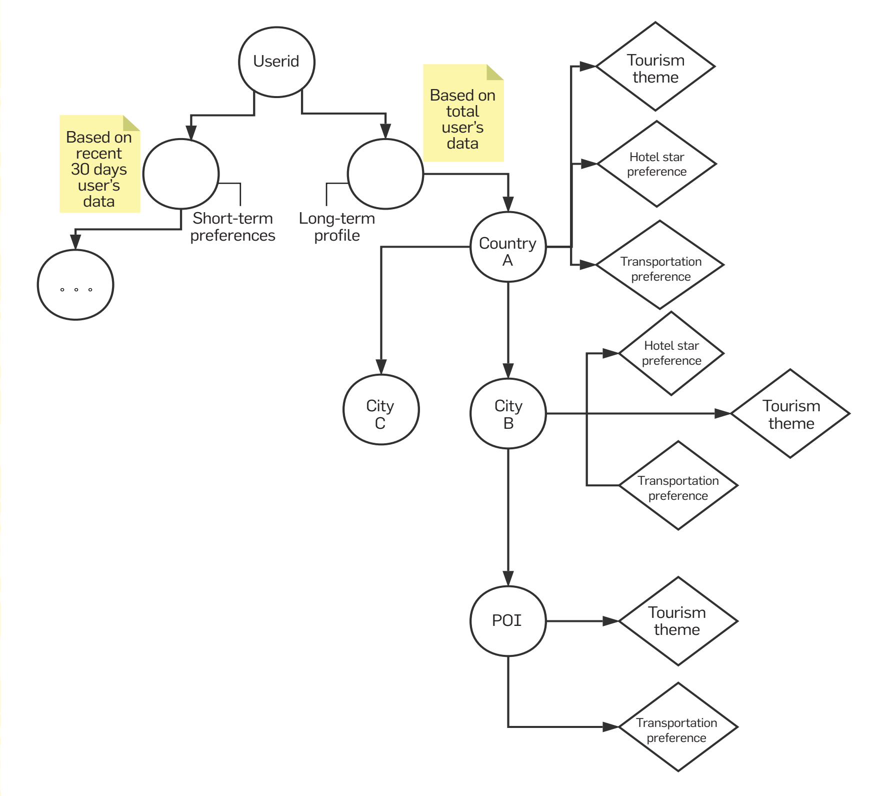
However, online tourism products are very different from regular commodities due to several factors, including: holiday travel is a low-frequency event, most people travel only 1–2 times per year; and, numerous travel packages generate different combinations of transport means, restaurants, and hotels. Thus, most travel products have very few or even zero customers, and it is very difficult to simply apply traditional recommendation algorithms to this scenario.

Ctrip's solution for recommendation is twofold: user-profiling based on its big tourism data accumulated over the last 18 years, and developing a hybrid collaborative filtering model that specifically targets the sparse data and cold-start problem.

Figure 5 is the user preference tree built from historical travel data. The short-term profile has the same structure of the long-term one, but is limited to the latest 30 days' data. The system can quickly iterate the tree and generate a preference vector for a user, as the input for the recommendation system.

The key to the enhanced recommendation algorithm is the so-called Additional Stacked Denoising Autoencoder (aSDAE),[3] which employs the deep learning model to learn the latent variables of users and products, and combine it with the classic matrix factorization. The latent variables learned from the two models are used to fit the product-scoring table that is initialized by users' feedbacks. Moreover, the overall loss function is a linear combination of two models' loss functions. Lastly, a text-generation AI component will creatively generate poetry to characterize the recommended attraction and push to users. The test has shown the algorithm performs better than traditional ones for the sparse data and cold start scenarios.

Figure 5. User preference tree built from Ctrip's big tourism data.



The system has been deployed to governments such as Henan province, Guiyang City, and many others. In the Henan province, for example, the recommendation system was deployed last March. According to Ctrip's online travel booking data, during the Labor Day holiday (a period of three days starting around May 1), the total number of tourists in the Henan province reached 2.04 million, which is a 41.5% increase from Labor Day in 2017. To evaluate the effect of its recommendation system, the tourist distribution over 18 areas throughout the province is calculated. The standard deviation (SD) of the distribution is 234,355 in 2017 and 202,208 in 2018. The SD decrease suggests a more balanced experience visiting the province's many attractions, which benefits both tourists and local residents. C

References

1. Blondel, V.D., Guillaume, J-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *J. Statistical Mechanics: Theory and Experiment 10*, (2008), P10008.
2. Chu, W., Liu, Y., Shen, C., Cai, D., and Hua, X-S. Multi-task vehicle detection with region-of-interest voting. IEEE Trans. *Image Processing 27*, 1 (2018), 432–441.
3. Dong, X., Yu, L., Wu, Z., Sun, Y., Yuan, L., and Zhang, F. A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of AAAI* (2017), 1309–1315.
4. Gao, S. and Chabini, I. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological 40*, 2 (2006), 93–122.
5. Lambiotte, R., Delvenne, J-C., and Barahona, M. Laplacian dynamics and multiscale modular structure in networks. arXiv preprint arXiv:0812.1770 (2008).
6. Malandraki, C. and Daskin, M.S. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science 26*, 3 (1992), 185–200.

Wanli Min is Chief Data Scientist and Senior Director at Alibaba Cloud Computing in Hangzhou.

Liang Yu is Senior Data Scientist at Alibaba Cloud Computing in Hangzhou.

Lei Yu is head of the AI Department at Ctrip in Shanghai.

Shubo He is manager of the AI Department at Ctrip in Shanghai.

china region

DOI:10.1145/3239548

BY HAI JIN/HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY, **HAIBO CHEN**/SHANGHAI JIAO TONG UNIVERSITY, **HONG GAO**/HARBIN INSTITUTE OF TECHNOLOGY, **XIANG-YANG LI**/UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA, **SONG WU**/HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Cloud Bursting for the World's Largest Consumer Market

CLOUD INFRASTRUCTURE IS information technology consisting of various hardware resources and software technologies. It enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be delivered with minimal management effort, often through the Internet. Cloud infrastructure today is a critical platform for many applications, providing basic support for the development of emerging areas, including big data, the Internet of Things (IoT), and artificial intelligence (AI). In 2016, International Data Corporation's *Cloud Computing Survey* reported cloud technology is becoming a staple of organization infrastructure, as 70% of organizations have at least one application in the cloud, and 56% of organizations are still identifying IT operations as candidates for cloud hosting.[1] In 2017, IDC predicted that by 2021, spending on cloud infrastructure and cloud-supported hardware, software, and services would double to more than $530 billion.[2]

For China, the overall market for cloud computing in 2016 was 51.49 billion RMB (China's currency), with an overall annual growth rate of 35.9% in 2016, which was significantly greater than the global average. It is expected that the China cloud computing market will continue to grow significantly over the next two years, reaching 136.6 billion RMB by 2020.[3]

The development and popularization of cloud computing, especially in emerging domains, brings great convenience. It also poses new challenges involving design and construction of modern cloud infrastructure. Cloud computing in China is also quite different from other countries, as it includes special requirements for the related infrastructure.

Here, we first explore the background of cloud infrastructure in China, then turn to its characteristics, and conclude with an outlook on development.

Background

Greatest number of netizens and IT employees. As of December 2017, the number of netizens in China was 772 million, or 55.8% of the total Chinese population, the number of online shoppers was 533 million, with annual growth of 14.3% over 2016, and the number of online payers was 531 million, including 527 million who pay through their smartphones.[4] As of January 2018, the total number of employees in computer, communications, and other electronic-equipment-man-

IMAGE BY VALERY BROZHINSKY

The 13th Five-Year Plan identified cloud computing as an important emerging national strategic industry.

ufacturing industries in China was 8.264 million.[5] Development of the cloud infrastructure in China involves both challenges and opportunities to meet the needs of such a large number of users and developers.

Best cellular infrastructure. The cellular infrastructure in China facilitates development of cloud infrastructure and services. As of September 2017, the total number of base stations in China was 6.04 million, including 4.47 million 3G and 4G base stations.[24] The base stations covered over 95% of the country, including even small villages with only dozens of residents. Moreover, as of December 2017, the number of smartphone netizens in China was 753 million and rising, with an annual growth rate of 8.2% over 2016, reflecting deep penetration of the mobile Internet.[25] As of November 2017, there were 3.91 million active smartphone apps available to Chinese consumers, along with more than 2.24 million local third-party apps, surpassing the number of apps (1.78 million) provided by Apple's app store in China.[4] Some unicorn apps, including Didi Taxi for a Chinese taxi-hailing service, create demand for unprecedented computing power.

Greatest demand for applications. With regard to construction of the cloud infrastructure, the most significant difference between China and other countries is that China has the largest and fastest-growing application demand due to having the greatest number of users. For example, the number of WeChat "HongBao"[a] sending and receiving activities on Chinese New Year's Eve in 2014 was 16 million, followed by a dramatic increase to 14.2 billion in 2017.[6] There were 46 billion "HongBao" activities over six days during the 2017 Spring Festival, with a 43.3% increase over 2016.[7] Another example of dramatic growth is that for the official website of the China Railway Corporation, "12306," the average number of page views per day was 55.67 billion and 81.34 billion during the peak period, and more than 10.2 million train tickets were sold through the site on January 11, 2018.[8] Moreover, on the 2017 "11.11"[b] sales day, peak database throughput for Alibaba was 472 million per second and the peak transactions traffic was 256,000 per second, a 2.1× increase over 2016.[9]

Innovation in the mobile Internet industry. China has seen rapid innovation in information industries and services in recent years. In terms of smart transportation, Didi Taxi, for example, had 450 million users and 20 billion daily planning requests and processed 4,500TB data per day in 2017. By accessing information from traffic cameras on smart traffic lights and applying cloud and big data analysis, Didi Taxi was able to improve its activity by 10%–20% in 2017.[10] Meanwhile, the number of active users of shared-bike applications was 221 million, or 28.6% of all Chinese netizens, as of February 2017.[4]

In digital gaming, as of December 2017, market penetration of mobile games in China was 76.1%, with each game user installing 3.35 mobile game apps on average.[11] In May 2017, more than 200 million people were playing "King Glory," a popular mobile game in China, with 54.1 million playing it daily.[12] The Chinese Electronic Commerce Research Center reported that during the "11.11" Shopping Festival of 2017, the total online trading volume in China was 253.97 billion RMB, a 45% increase over the same period in 2016. Moreover, major e-commerce enterprises reported 850 million express logistics orders on November 11, 2017, with courier volume during this "11.11" Shopping Festival of more than 1.5 billion, historic peak levels.[13] The data shows the cloud infrastructure in China is able to serve billions of users concurrently.

Characteristics

Cloud infrastructures worldwide share certain characteristics, including resource-on-demand, elasticity, and geo-distribution. Due to the deep penetration of the mobile Internet and proliferation of mobile apps in China, China's cloud infrastructure

[a] HongBao is a red envelope with money inside traditionally given by older people to young people as a gift during important festivals; it went digital via WeChat and Alipay.

[b] 11.11 is an online shopping carnival (such as Taobao and TMall) run annually by Alibaba on November 11.

largely centers on the app ecosystem, as characterized in the following ways:

(Super) app-oriented infrastructure. Consider the top two largest cloud service providers in China: Alibaba Cloud and Tencent Cloud. Alibaba Cloud originally sought to address the need for massive computing resources for the "11.11" Shopping Festival, with 90% of sales from mobile apps like Taobao and TMall. Tencent Cloud sought to address the computing infrastructure for Tencent's flagship mobile apps: QQ and WeChat. Its cloud infrastructure thus centered on such super apps, serving hundreds of millions of people daily. The cloud infrastructure cannot be developed simply by reusing open source cloud stacks like OpenStack due to the need of unparalleled concurrency from such apps. To this end, major cloud service providers in China tend to build their own infrastructures, with heavy optimizations tailored for their super apps.

WeChat's infrastructure evolution. Here, we consider the evolution of WeChat as an example of how Tencent builds and customizes its cloud infrastructure.[18] WeChat's initial goal was to develop a message-oriented chat app, with messages synchronized between a sender and its receivers. China is thus the key market worldwide. To satisfy this extremely large-scale requirement, Tencent customized its own Remote Procedure Call library (called Svrkit), an infrastructure pillar for connected distributed planetary services for WeChat. A key design decision in such synchronization was how to propagate messages. Many such designs have adopted the "pull" mode, whereby receivers pull messages from a sender. In contrast, WeChat adopted a "push" propagation mode, because there was a strict upper limit in chat groups, originally 20, later increased to 100 and today 500. The cost for push propagation is bounded, and the receiver, or each WeChat app, can deliver much lower latency. Due to having to serve a rapidly growing user population, WeChat initially adopted a micro-serviced architecture, including aggressive division of functionalities of business representation layer into multiple logic servers (logicSrv); even the same functionality with different priority is divided. For example, message-delivery functionality is divided into three service modules—message synchronization, voice and text message sending, and figures and videos sending—allowing out-scalability with increased numbers of users.

With its customized infrastructure, WeChat grew from a chat app to a massive digital payment system; "HongBao" can be viewed as a special kind of payment, an enterprise business platform, and a development-and-delivery platform (WeChat Mini apps). With its developing ecosystem, the WeChat platform could become not only a cloud platform itself but also attract a large number of third-party apps to access its services on Tencent Cloud, providing seamless integration and winning strong user loyalty.

Scalable and hybrid infrastructure for bursty loads. Many super apps exhibit strong bursty loads, especially under the extra demand on special days or during certain seasons. The cloud infrastructure needs to not only be able to scale up easily but also scale in afterward. Unlike Amazon and Google, which mainly deploy services in their own large-scale datacenters, major cloud service providers in China tend to rent existing datacenters to scale out their services and cloud-based systems, in addition to their own datacenters. These providers rent datacenters because a larger number of small- to medium-size datacenters were built during the IT revolution of the 2000s but had relatively low utilization. For example, as of 2017, there were more than one million datacenters in China, but most were small, each occupying less than 500 square meters.[21] Cloud service providers usually rent datacenters to quickly scale up their services under bursty loads, then scale in to avoid possibly wasting the infrastructure cost. To allow quick deployment of services, they usually built their customized infrastructure to allow quick deployment of services and increase resource utilization.

Technologies behind Alibaba's "11.11" Shopping Festival. The unprecedented peak transactions per second requires technologies that are not only from the off-the-shelf open source stack. Alibaba, for example, uses its own complete customized software stack, from infrastructure software to cloud software.[19] Its current world-record TPC-C result is approximately 500,000 transactions per second,[22] while it handles up to 42 million operations/second in its database involving 325,000 and 256,000 NewOrder and Payment Transactions per second.[c] To this end, Alibaba has created its own open-source database (called OceanBase) and distributed file system (called TFS). In order to meet the resource requirements of peak traffic, Ali Cloud is able to expand capacity by 100,000 servers in one hour. To quickly deploy such services, it created Pouch, a customized container framework and aggressively deployed and scheduled online services with offline services through its Sigma scale-out scheduler. While hybrid cloud has been advocated for years, Alibaba has pioneered seamless integration of its public cloud with the datacenters of its partners to deliver one-stop handling of individual transactions among multiple service providers.

Deeply integrated/optimized infrastructure. Like other technology giants Amazon, Microsoft, and Google, the growth of cloud scale makes efficiency a top optimization target, as even a single-digit increase in utilization or performance density could save tens of millions of U.S. dollars. This motivates cloud service providers in China to provide deeply integrated and optimized infrastructure.

China's cloud infrastructure is moving toward hardware/software co-design to improve efficiency and flexibility. Huawei Cloud, another very large cloud service provider in China, publically released its *Service Driven Infrastructure* plan in 2014, including software-defined storage and software-defined networking.[20] This allowed offloading key processing functionalities in hardware while retaining software flexibility. Ali Cloud recently released its X-Dragon Cloud Server to aggressively offload VM management services, as well as

[c] Note real-world transactions are much more complex than TPC-C, as each user-facing transaction generates a large number of transactions.

customized data-processing services, to a tightly coupled physical installation. It also recently announced it would produce its own neural processing units for AI-related tasks. And UCloud, a top-five cloud service provider, announced its release of near-data-processing infrastructure for big-data applications, yielding improved efficiency.

Outlook

China's cloud infrastructure has made great strides, supporting large-scale applications and millions of users. The rapid development of cloud infrastructure has been promoted both through national research projects and through the corporations involved. The Chinese central and local governments now plan to push development of cloud computing while mainstream enterprises pursue a new round of cloud computing designs.

The government's plan for developing cloud computing. The Chinese central government is emphasizing development of cloud computing and its underlying infrastructure. For example, the 13th Five-Year Plan identified cloud computing as an important emerging national strategic industry.[14] And the Ministry of Industry and Information Technology of China adopted a Three-Year Development Plan for cloud computing, 2017 to 2019, aiming to increase the cloud computing industry in China to 430 billion RMB by 2019.[15] The Chinese central government is also funding a series of projects for cloud computing. In 2017, the "Cloud Computing and Big Data" Special Program of the National Key Research and Development Plan launched 15 projects with total funding of 409 million RMB.[16] In 2018, it plans to start 20 projects with a total budget up to 625 million RMB.[17]

Enterprises' plan for developing cloud computing. Chinese enterprises are developing an increasingly powerful cloud infrastructure to provide competitive cloud computing products and services. For example, Inspur and Sugon launched a series of scientific projects to research key technologies in cloud datacenters and servers. And Alibaba expects to use its cloud unit to carry it through the next decade. According to a Gartner research report in September 2017, Ali Cloud has surpassed Google in IaaS Public Cloud Service and is today the third largest cloud provider in the world.[23] In 2015, Tencent adopted its "Cloud Plus" plan to develop Tencent Cloud, which will invest 10 billion RMB to build a cloud platform and ecosystem over the next five years. Meanwhile, Huawei has established a new business group dedicated to developing Huawei Cloud.

Emerging computing paradigms and cloud computing. Information technology is evolving quickly. Emerging computing paradigms like AI, the IoT, and Cloud-Edge computing have begun to influence the cloud infrastructure and offer opportunities for addressing cloud-related challenges. Machine- and deep-learning algorithms and models for AI are relevant for cloud computing researchers and practitioners. On the one hand, the cloud can benefit from machine and deep learning to support more smart resource management. On the other, machine- and deep-learning requires large-scale computing power, and the cloud is an essential platform for hosting AI services due to its potential for high scalability and ready access to computing resources.

With the rapid development of the mobile Internet and IoT applications in China, the existing centralized cloud computing architecture faces significant challenges. Edge computing is being investigated as a way to better exploit capabilities at the edge of the network to support the IoT. In edge computing, the massive amount of data generated by different kinds of IoT devices can be processed at the network edge instead of having to first transmit it to the centralized cloud infrastructure due to bandwidth- and energy-consumption concerns. Edge computing can thus provide services with quicker response and greater quality compared to traditional cloud infrastructure and is more suitable for being integrated with IoT to provide more efficient and secure services for a vast number of end users.

Further Reading

1. 2016 IDG Cloud Computing Survey; https://www.idg.com/tools-for-marketers/2016-idg-enterprise-cloud-computing-survey/
2. IDC FutureScape: Worldwide IT Industry 2018 Predictions; https://www.idc.com/getdoc.jsp?containerId=US43171317
3. White Paper on Cloud Computing Development in China (2017); http://www.fx361.com/page/2018/0112/2686558.shtml
4. The 41st China Statistic Report on Internet Development; http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201803/P020180305409870339136.pdf
5. China Entrepreneur Investment Club; https://www.ceicdata.com/zh-hans/china/no-of-employee-by-industry-monthly/no-of-employee-computer-communication--other-electronic-equipment
6. Tencent Tech: HongBao War; http://new.qq.com/omn/20180215/20180215C0EIMO.html
7. 2017 WeChat Spring Festival Data Report; http://tech.qq.com/a/20170203/010341.htm
8. China Railway Site Sees 5.93 Billion Clicks Per Hour as Busiest Travel Season Starts; https://technode.com/2018/01/16/chunyun-data/
9. Alibaba Tech: Fight Peak Data Traffic on 11.11: The Secrets of Alibaba Stream Computing; https://medium.com/@alitech_2017/how-to-cope-with-peak-data-traffic-on-11-11-the-secrets-of-alibaba-stream-computing-17d5e807980c
10. 2017 Annual Urban Transportation Report. DidiChuxing; http://index.caixin.com/upload/didi2017.pdf
11. JiGuang. 2017 Mobile Gaming Market Research Report; https://community.jiguang.cn/t/topic/24810
12. JiGuang. 2017 King Glory Research Report; https://www.jiguang.cn/reports/72
13. 2017 '11.11' E-Commerce Platform Shopping Festival Evaluation Report; http://www.100ec.cn/zt/upload_data/17sh11bg.pdf
14. The development plan of the 13th Five-Year Plan national strategic emerging industry; http://www.gov.cn/zhengce/content/2016-12/19/content_5150090.htm
15. The Three-year Development Plan of Cloud Computing (2017–2019); http://www.miit.gov.cn/n1146290/n4388791/c5570594/content.html
16. 2017 Project List of 'Cloud Computing and Big Data' Special Projects in The National Key Research and Development Plan; http://app.myzaker.com/news/article.php?pk=59a4e2d41bc8e03727000029
17. 2018 guide for projects of 'Cloud Computing and Big Data' Special Projects in The National Key Research and Development Plan; http://www.stdaily.com/kjzc/top/2017-10/10/content_582554.shtml
18. The evolution of WeChat Infrastructure; http://www.infoq.com/cn/articles/the-road-of-the-growth-weixin-background
19. Techniques behind TMall's 11.11 Shopping Festival; https://jaq.alibaba.com/community/art/show?articleid=1201
20. Huawei SDI Innovation Architecture; http://www.cnetnews.com.cn/2014/0918/3034037.shtml
21. Analysis of 2017 China Datacenter Sector Development and Evolution; http://www.chyxx.com/industry/201709/564441.html
22. http://www.tpc.org/tpcc/results/tpcc_results.asp
23. Iaas Public Cloud Service Market Share; https://www.channele2e.com/news/gartner-public-cloud-iaas-market-share-amazon-aws-microsoft-azure-google-growth/
24. Number of base stations in China; http://tech.sina.com.cn/roll/2017-10-22/doc-ifymzksi0587142.shtml
25. Data analysis on the number of smartphone users nationwide; http://www.chinabgao.com/k/zhinenshouji/28395.html

Hai Jin is the Cheung Kung Scholar Chair Professor at Huazhong University of Science and Technology, Wuhan.

Haibo Chen is a professor and Director of the Institute of Parallel and Distributed Systems at Shanghai Jiao Tong University, Shanghai.

Hong Gao is a professor at Harbin Institute of Technology, Harbin.

Xiang-Yang Li is a professor and Executive Dean of the School of Computer Science and Technology at the University of Science and Technology of China, Hefei.

Song Wu is a professor and Director of the Institute of Parallel and Distributed Computing at Huazhong University of Science and Technology, Wuhan.

DOI:10.1145/3239550

BY YUAN QI/ANT FINANCIAL, JING XIAO/PING AN TECHNOLOGY (SHENZHEN) CO., LTD.

Fintech: AI Powers Financial Services to Improve People's Lives

FINANCIAL TECHNOLOGY, ALSO known as fintech, is a fast-evolving field that has reshaped the financial industry. Ant Financial has redefined digital financial services, specifically mobile payment and microloan services, and Ping An Technology has developed innovative fintech to reshape the insurance, investment, and banking businesses.

Computing technologies play an important role in the transformation of modern financial services. Ant Financial, an affiliate of Alibaba and a leading Chinese fintech company, has participated in this transformation by using technology to bring financial services to hundreds of millions of individuals and small businesses in China and throughout the world. Two examples demonstrate its impact.

Ci Ren Ge Dan (Figure 1) runs a tent store at the foot of Mount Everest, 5,200 meters above sea level. He used to take half a day to go to the nearest bank. Carrying and keeping a lot of cash was inconvenient. Now he is one of many small merchants served by Ant Financial. Last year, Ci Ren Ge Dan added QR codes to items in his store, which allows tourists to pay him using their cellphones. He can also use his phone to pay electricity bills, deposit money, and acquire funds without leaving his store.

Zhang Yousheng (Figure 2) is a herdsman who has raised cattle for decades. In the past, he worried about having the funds he needed to buy calves and fodder, and about selling his cattle. After Ant Financial partnered with a cattle industry company to provide low cost microloans, Zhang said he no longer worries about funds and sales. His life as a herdsman is easier. Ant Financial's services have helped tens of millions of small and micro merchants in China, from prosperous cities to remote rural areas. The technologies behind these stories are a series of innovations that make financial services more accessible and affordable to everyone.

When an accident happens, the customer only needs to take a few pictures of the damaged car to file a claim from the accident site.

A Level of Trust

The innovations behind Ci Ren Ge Dan's story are payment technologies. Ant Financial started an escrow payment service 14 years ago, and held shoppers' payments until merchants delivered purchased items, providing a needed level of trust to e-commerce users. A second innovation came in 2010 when Ant Financial designed an express payment system that gave both users and banks a trusted payment platform. Although the initial technological challenge was expected to be connecting all banks, it turned out that the real challenge was controlling risk given a rapid increase in transaction volume. As a result, Ant Financial developed real-time risk management technologies that used rules and algorithms to analyze hundreds of thousands of transactions per second, improving transaction security dramatically. The express payment method has become a standard for Web and mobile payment.

In 2013, Ant Financial created a new service called Yu'e Bao. The service pays interest on users' current account balances including funds "left over" from digital transactions; it also serves as a payment platform. Yu'e Bao was quickly adopted by hundreds of millions of users, and is now the largest money market fund in the world. To handle such large traffic volume, Ant Financial redesigned its payment systems using distributed architecture and cloud computing, and migrated its partners' fund management systems to the architecture. Furthermore, the rise of the mobile Internet led users to explore different options—including NFC, Bluetooth, sound wave, barcode, and QR code—for so-called offline mobile payment. None of these options was perfect in terms of user experience, cellphone support coverage, and cost. Balancing these factors, Ant Financial chose a QR code- and barcode-based offline mobile payment method: a merchant can scan a consumer's QR code or barcode to record a transaction, or the consumer can scan the merchant's QR code. The innovation of QR payment builds a point-of-sale transaction upon a cheap sticker that's affordable to even the smallest merchant and serves as the foundation of mobile payments.

In 2017, Ant Financial launched the Smile-to-Pay service based on computer vision technology. Instead of using a cellphone, a user smiles to a vending machine to complete a payment. As the first commercial facial recognition payment system, Smile-to-Pay took security and the user experience to a new level. The AI-driven product is based on imaging and vision analysis technology developed internally by Ant Financial.

Extending Microloans

Microloans are another area of Ant Financial innovation. When the service was launched in 2010, the first loan was for only 1,300 RMB (~$180 USD). Ant Financial built a credit model based on data of merchants' previous sales and transactions. The combination of computing power and past behavior extended microloan services to more merchants. However, the operational costs were quite high, the user experience needed improvement, and it took three days for a merchant to get a loan.

A leap forward occurred when the system adopted advanced machine learning methods—including boosting, deep learning, and graph-based machine learning—for accurate credit modeling. The new system is characterized by three digits: 3, 1, and 0; a merchant takes less than 3 minutes to complete a loan application, obtains the decision in 1 second, with zero human intervention. The integration of systems engineering and algorithmic advances into the microloan operation makes the cost of each loan less than 2 RMB ($0.31 USD). Combining a convenient user experience with low operational costs, Ant Financial now serves tens of millions of merchants in China with accessible and affordable loans.

Financial service providers face three challenges when digitizing service for the future economy—connection: how to link users, merchants, and service partners in a low-cost, fast, and intelligent way; risk: how to control for aspects of financial risk; and trust: how to grant equal opportunity for all to be trusted, and trustworthy, in the digital space. To address these challenges, Ant Financial focuses on five technologies: blockchain, AI, security, IoT, and computing (BASIC). Blockchain helps to build a trusted global interconnected system capable of storing, exchanging, and processing values; AI enables companies to build intelligent systems that better serve customers and business partners, and drive new product design; security is a pillar that makes digital systems safe and stable; IoT (Internet of Things) is a bridge that connects the physical and digital realms to transformative effect; and computing engines provide the digital space with computational power. The following paragraphs give more thoughts from Ant Financial on two of them: blockchain and AI.

Two of the BASIC technologies merit a closer look. Blockchain provides a new trust mechanism to transactions. Over the past two years, Ant Financial has used it to improve the transparency of charities, strengthen the trust of insurance contracts, ensure the authenticity of house rental contracts, and improve the traceability of e-commerce supply chains. Ant Financial's applications are based on a consortium blockchain. However,

Figure 1. Ci Ren Ge Dan uses Ant Financial's services to receive and make payments for the tent store he operates at the foot of Mount Everest, 5,200 meters above sea level.

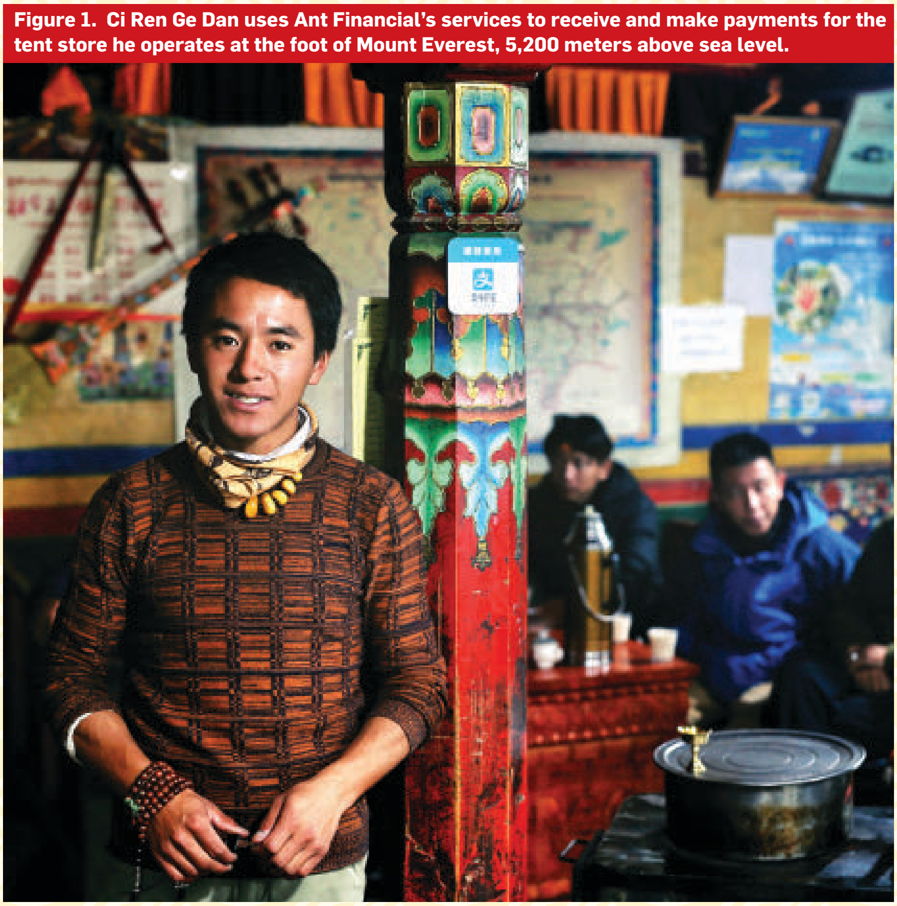


Figure 2. Zhang Yousheng, a herdsman, uses Ant Financial microloans to purchase calves and fodder.

current blockchain technologies face several key challenges in large-scale financial applications. Take a global e-commerce supply chain as an example. To support a global supply chain, blockchain nodes should be deployed in different continents, which affect the fairness of the consensus algorithm used by the blockchain system. If all supply, distribution, and sales records are stored in the same chain, the chain must be able to support hundreds of thousands of transactions per second. Not all records should be transparent to all participants, so a comprehensive mechanism is needed to protect the privacy and ownership of the data on the chain. All these are serious hurdles to a blockchain system.

Thus Ant Financial has developed an industrial-grade blockchain system to address these challenges. The company plans to share the system's value and open its blockchain technologies to the public in 2018.

Deep learning and natural language processing technologies helped intelligent customer service robots achieve higher customer satisfaction rates than live service staffs.

Robots Service Ratings

Ant Financial uses AI to create a financial brain for the digital world. Recent years have witnessed the huge success of machine learning and deep learning in machine perception areas such as speech recognition and image analysis, but financial services need more, including prediction and decision-making. These capabilities, combined with a comprehensive financial knowledge graph, are the foundation of the financial brain at the core of Ant Financial's risk, credit, and customer service engines. The brain enabled Ant Financial to reduce its payment loss rate to less than one in a million, automatically answer millions of customer inquiries a day, automatically assess car damages based on computer vision and a vehicle knowledge base, and improve other services. In particular, deep learning and natural language processing (NLP) technologies helped intelligent customer service robots achieve higher customer satisfaction rates than live service staffs. During the popular Singles' Day 2016 shopping occasion, 97% of customer service inquiries on Ant Financial's Alipay service were handled by the intelligent customer service robots.

Fintech Revolution

The surge of fintech in the past few decades has revolutionized the way financial industry personnel work, think, and live. Ping An has developed numerous technologies to advance the industry. Its areas of fintech concentration can be summarized as ABCDS: artificial intelligence, blockchain, cloud, big data, and security. AI is the core engine that drives industry automation and intelligence. Blockchain provides a revolutionary trust mechanism. Cloud computing lays the foundation for processing massive amounts of online transactions. Big data aids knowledge mining and decision making. Security is the essential element for safe and stable systems.

The core of Ping An's AI platform is the Ping An Brain engine. Covering a broad range of data analytics and AI techniques such as biometrics, NLP, image recognition, and more, Ping An Brain can provide full-stack AI solutions to enhance financial services scenarios such as marketing, customer service, and decision support. It has been successfully deployed across Ping An's insurance, investment, and banking businesses, greatly improving their effectiveness, efficiency, and costs.

For blockchain, Ping An was an early adopter, and has deployed a blockchain-based production system since 2016. By the end of 2017, it had over 12 blockchain-based platforms, covering fixed income trading, asset-backed securities, post trade reconciliation, and other transactions. By March 2018, its blockchain network had over 20,000 nodes across China and handled transactions valued at over one-trillion RMBs, including over 90% of those for Ping An One-Connect, the Ping An Group's fintech subsidiary.

A series of applications showcases its use of AI, big data, and blockchain.

As the capacity and scope of the insurance industry expands, the number of claims increases and leads to issues such as processing latency, high risk, potential misjudgment, and possibly fraud. To resolve such issues, Ping An has leveraged AI techniques across all insurance industry scenarios, including fraud detection, customer acquisition, and claims processing.

AI Assessment of Claims and Risks

Take auto insurance as an example. When an accident happens, it often takes a long time to process a claim. Customers wait onsite for investigators to arrive and assess the damage, they wait as the claim is filed and processed, and they wait for a final decision. It is inconvenient for customers and costly to insurance companies. The process is also vulnerable to fraudulent claims. To address such problems, Ping An developed a system where the customer only needs to take a few pictures of the damaged car to file a claim from the accident site. The claim is processed within seconds and the customer given a precise payment calculation. The system involves a series of key modules: picture quality assessment, verification of insurance, car segmentation, identification of damage and related parts, payment calculation, and fraud detection. A number of AI techniques, such as image processing, image segmentation, and object recognition, were developed to support the functions. The system has been running in production at Ping An for over a year, successfully processing over 30,000 claims each day. It not only improves claim processing efficiency and thus customer experience, but also stops potential frauds on the order of multi-billion RMB. This system is now available to the insurance industry through the Ping An OneConnect platform.

Investment banks often need to assess the potential value and risk of targeted customers—individuals for retail banking or enterprises for corporate. In today's big data era, information comes from a broad range of resources with complex relationships between them. To make a precise assessment, it is crucial to organize and analyze such complex information in an efficient and effective manner. This is exactly what knowledge graph is designed for. Ping An has developed various knowledge graph techniques for retail and corporate businesses.

Take corporate risk assessment, for example. There are over 70-million registered enterprises, including households, in China. Their information comes from three major sources: commercial registration and daily operation; public news announcements and social posts; and business relationships including the supply chain, investments, and legal actions. To organize and analyze such rich and dynamic data, Ping An developed Euler Graph, an enterprise knowledge graph. The graph covers nearly all of China's 70-million enterprises, using data from all three sources. Millions of legal proceedings are automatically interpreted and over 40-million lawsuit relationships have been extracted and incorporated into the graph. Signals on enterprises are collected from over 300 news and social sites, totaling hundreds of thousands of articles daily, and updated every 10 minutes.

Information from these and other sources grows quickly. Deep graph analysis algorithms support business decisions on risk assessment and other matters. One advantage of Euler Graph is that business logic is directly integrated. For example, risks are assigned using different business logic for investments, bonds, or loans, and signals are extracted from an analysis of social and news data. Upstream and downstream relationships may also be encoded as risk indicators. When a risk event occurs upstream, the incident passes through the graph network and may influence an assessment. Through effective analysis by Euler Graph, risks such as defaults were successfully detected three to nine months ahead of occurrence. Euler Graph is also used for other applications, such as precision marketing and exploring investment opportunities.

Large-Scale Blockchain Architecture

Ping An OneConnect has identified various shortcoming impeding the wide scale adoption of blockchain. Performance and scalability bottlenecks have hindered its potential in building high volume financial transaction systems, and issues of data privacy and confidentiality have limited its usage in public service areas where few entities are willing to share data.

Ping An's blockchain research and cryptography team responded with the FiMAX platform. The architecture is designed to address all key problems hindering large scale blockchain adoption, with performance matching traditional databases systems and privacy protection enabled by advanced cryptology including various Ping An designed zero knowledge proof algorithms.

FiMAX has not only earned praise from Ping An's business partners, it has also gained recognition with its selection for some of the largest international blockchain networks being built for banks and regulators. For example, one cross border blockchain network to be launched later this year will comprise over 10 international banks and over 100 nodes.

Conclusion

Ant Financial made a series of innovations that led to key technologies behind mobile payment and microloan services in China. Ping An used innovative techniques to improve financial services for insurance, investment, and banking industries. Much progress has been made, but every problem solved opens the door to further questions and considerations. How should we model the transaction systems in a large-scale dynamic network, and implement intelligent inference and reasoning for better financial services? How can data be utilized and user privacy protected at the same time yet better than through current methods such as differential privacy? How can causal inference be applied in a complex system and when only observational data are available? Answering these questions will lead to tomorrow's breakthroughs. C

Yuan (Alan) Qi is Vice President and Chief Data Scientist at Ant Financial, Zhejiang.

Jing Xiao is Chief Scientist and Executive Member at Ping An Insurance (Group) Company of China LTD, Shenzhen.

DOI:10.1145/3239552

BY HUAXIA XIA/MEITUAN,
HAIMING YANG/JD CTO GROUP

Is Last-Mile Delivery a 'Killer App' for Self-Driving Vehicles?

CHINA'S E-COMMERCE BOOM has generated a huge logistics demand, both in terms of express package delivery and on-demand food delivery.[a]

Chinese express firms delivered an estimated 40 billion parcels in 2017; up 28% from the previous year.[2] Indeed, China's on-demand delivery market exceeded over 30 million food orders daily by yearend 2017.

This growth in delivery services has created a healthy job market, with the number of employees in this sector up 130% from 2014 to 2017, according to the China Federation of Logistics and Purchasing. *Last-mile delivery*—the movement of goods from a transportation hub to final destination—has witnessed huge growth. Today there is an estimated three million couriers working in last-mile express delivery in China, and another one million couriers working for on-demand delivery services.

The country expects the express delivery industry to increase another 60% by 2020,[3] requiring an even greater delivery force. However, the working population in China is aging; its numbers in steady decline since its peak in 2011 (a situation mainly caused by China's 40-year-old "one-child policy," which ended in 2016). This decline is expected to continue for at least another decade.

These two conditions have motivated the industry to seek a more efficient delivery solution—the autonomous delivery vehicle. But there are two dimensions of challenge—a profitable business model, and of course technology.

The Business Model of Autonomous Driving in Last-Mile Delivery

We have seen two major upgrades to the last-mile delivery business model: the warehouse is getting closer to the end users, and deliveries are merged into "multi-deliveries."

In traditional logistics, the warehouse is a motionless unit used only for storage. Goods are transported from many other locations and then distributed to the multiple end users. The new express logistics, however, redefines the warehouse concept to be not only storage but also a mobile facility that relocates to serve many more users. Last-mile delivery is one such scenario applied to this "motion warehouse" concept, and the autonomous driving vehicle is one of its core technologies. The concept of motion warehouse is a revolution of logistics, which has combined the three major elements of retail business—people, goods, and warehouses—into the new concept. The warehouse is aware of the needs of people, and provides the goods selection through the AI based big data analysis. In the new "motion warehouse" model, consumers can get the goods they want more accurately and quickly.

[a] http://news.iresearch.cn/content/2017/11/271315.shtml/

IMAGE BY CHESKY

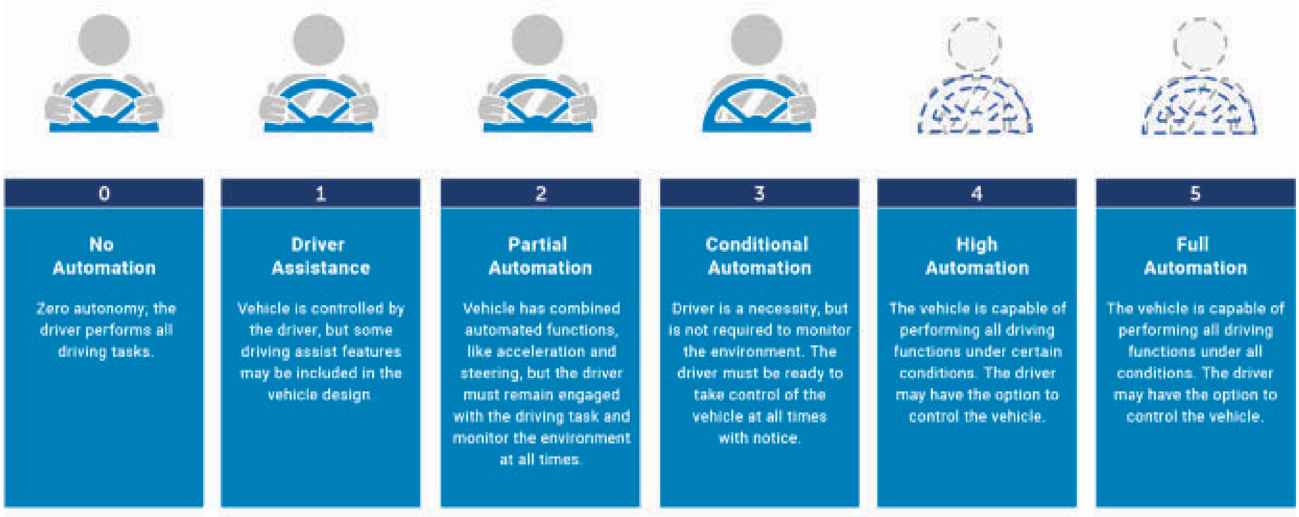
This approach extends the supply chain concept from the warehouse all the way to the end users. In a traditional logistic system, the connection is from the preposition warehouse directly to the end users. (A preposition warehouse is one in close proximity to the consumer). It could be an office building, or a small warehouse set up to serve a community that enables user delivery in 1–2 hours. However, with increasing population and demand, the "campus" model is emerging as an efficient way for even better user experience. Campuses' aggregate delivery demands are always large so preallocation to customers and traditional (human) last-mile delivery is no longer the most efficient. By transporting a large number of packages together before the are allocated to customers, and delivering passively (users pick up packages at designated location) and proactively (carried directly to the end users), both user experience and efficiency are increased. Autonomous vehicles are key to achieving these benefits.

The Technology of Autonomous Driving in Last-Mile Delivery

Autonomous driving technology is not ready to replace human drivers in passenger vehicles. Many luxury vehicles may be equipped with advanced driver-assistance systems (ADAS)—including emergency braking, backup cameras, adaptive cruise control, and self-parking systems—but they are not fully autonomous. According to SAE International's levels of vehicle autonomy as depicted in Figure 1, current ADAS functions mostly hover around levels 1 or 2. Technology is still far from realizing level 4—a truly driverless car.

Two key challenges for fully autonomous passenger vehicles are trustworthiness and price. A typical autonomous passenger vehicle (as shown in Figure 2) is equipped with an array of state-of-the-art sensors and other technologies that can cost hundreds of thousand dollars.[1] But even with such expensive equipment, autonomous vehicles are still far from reliable in terms of passenger safety. For example, last March a Tesla driver was killed when his car, set in "autopilot mode," collided with a median barrier on the highway, causing the vehicle to catch fire.

Figure 1. SAE automation levels.



0

No Automation

Zero autonomy; the driver performs all driving tasks.

1

Driver Assistance

Vehicle is controlled by the driver, but some driving assist features may be included in the vehicle design

2

Partial Automation

Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.

3

Conditional Automation

Driver is a necessity, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice.

4

High Automation

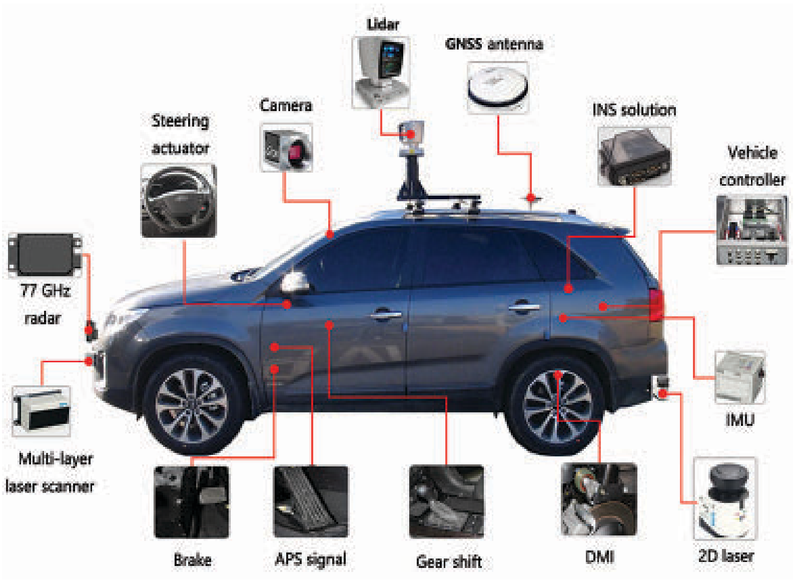
The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.

5

Full Automation

The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.

Figure 2. Sensors on a typical autonomous passenger vehicle.



Summary of California autonomous vehicle disengagement reports from 2017.

Company	Autonomous miles	Number of disengagements	Miles per disengagement
Waymo	352545	63	5596
GM Cruise	131676	105	1254
Zoox	2255	14	161
Baidu	1979	43	45
Bosch	1454	598	2.4
Mercedes Benz	1087	652	1.7

China just issued its first license for self-driving tests last March, and has not yet published any test result data, so we refer to autonomous vehicle disengagement reports issued by California's Department of Motor Vehicles in 2017 (and summarized in the accompanying table). Waymo's autonomous car required human assistance every 5,596 miles on average, which is the best among all tested vehicles. By comparison, a human driver on average has one accident approximately every 165,000 miles. Regulators will require autonomous cars prove much safer than current human behavior before drivers are no longer needed behind the wheel. Much more research and engineering efforts, possibly over a decade's worth, is required to improve autonomous driving technology, including higher-resolution sensors, better algorithms, and faster computing chips.

Last-mile delivery, however, is a solid scenario illustrating how autonomous-driving technology has been successfully and safely employed. There are a few key differences between a delivery vehicle and a passenger vehicle. The last-mile delivery vehicle usually runs slowly, typically 20mph. A slow vehicle requires shorter perception distance, shorter braking distance, and less computing frame rate. Secondly, the last-mile delivery vehicle is typically smaller and lighter than passenger vehicles; this further decreases the risk of possible damage or harm when an accident happens. Finally, a delivery vehicle is free of passengers, therefore, it has fewer requirements for safety, planning, and control algorithms.

There are two major challenges in last-mile delivery, which autonomous driving can help: The distribution location of the package, and the delivery path. From past experience, when a delivery person arrived on location, the majority of time was spent waiting, especially when the planned delivery consisted of a large number of small packages destined for office buildings, campuses, and apartments. The waiting time for customers, and the handling time to delivery to customers, killed any efficiency of last-mile delivery. Moreover, the delivery person is paid by the number of packages delivered, meaning the company often pays a great deal of money for a trip to a single location, which kills any cost efficiency of last-mile delivery.

To be effective, last-mile delivery must determine the best route to dispense the most parcels. In a city, the best route is often *not* the shortest route, and road conditions constantly changed over time. An autonomous driving cart is similar to a larger "self-closing cabinet," which can save both the average waiting time and the distribution time. On the other hand, equipment costs increase with the autonomous approach. Ideally, there would be a cost of only ¥1.5 per autonomous delivery compared with current ¥7–10 per delivery. Achieving this requires reducing costs of trial carts from ¥600,000 to ¥50,000 if the autonomous delivery (and vehicles) are in mass production.

Autonomous driving vehicles have major technology challenges, too. One obstacle is the behavior of the motion detection system when out in the real world. The algorithm may work perfectly in lab test conditions, but may not perform well when it is on the open road. Another difficulty with these vehicles is the range of vision when driving in dark or shadowing areas. Like the human eye, the range of the vision may vary in different levels of brightness; even with infrared light detection, the vehicle may not "see" in foggy and dusted environments.

How JD Uses the Vehicles in Delivery Scenarios

JD.com is the largest e-commerce platform by revenue, and offers a world-class set of online retail services to its legion of users, who now number close to 300 million in total. As a technology-driven company, JD.com has focused considerable effort in developing a robust and scalable retail platform that not only supports the company's rapid growth but also allows it to provide cutting-edge technology and services to its partners and customers.

JD selected last-mile delivery practices as the first line of defense in its campaign to upgrade its logistics infrastructure. JD Logistics' autonomous delivery vehicles will primarily be used for last-mile delivery in urban areas, carrying packages from dedicated stations to office buildings, pick-up stations, residential area convenience stores, and other locations. It was first used to support JD.com's renowned two-hour express delivery service and will be rolled out across JD's deliver network to be used for a wider range of applications. Autonomous driving vehicles will be loaded at delivery stations and will travel to pick-up points designated in advance by consumers. The recipient can collect the products they order simply by pressing a button on JD's mobile app. JD's delivery vehicle can also recognize the customer using face identification and deliver the product accordingly.

JD conducted its first trial in autonomous driving vehicles for last-mile delivery on June 18, 2017 at Renmin University, Beijing. The vehicle delivered about 10 packages in approximately six hours. JD subsequently deployed approximately 60 autonomous driving vehicles for last-mile delivery at Beijing, Xian, and Hangzhou for pilot AI-based package delivery. The city of Xian has been selected as the headquarters for JD's fleet of vehicles. In December 2017, JD Group CEO Qiangdong Liu announced last-mile parcel delivery plans for 100 universities.

Figure 3. The deployment progress of JD's autonomous driving vehicle.

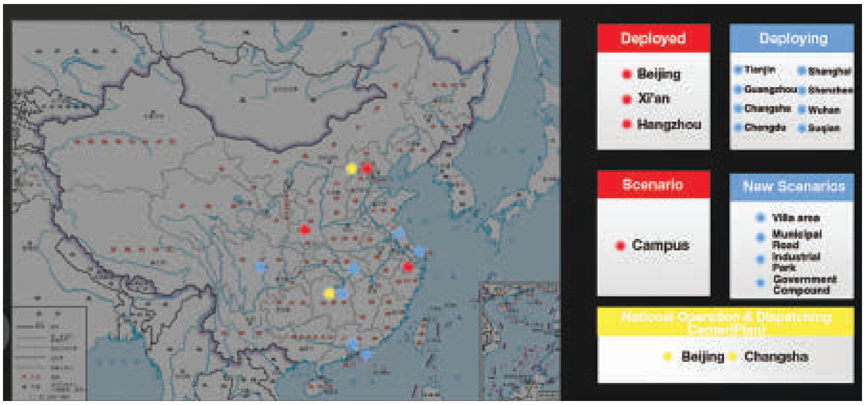
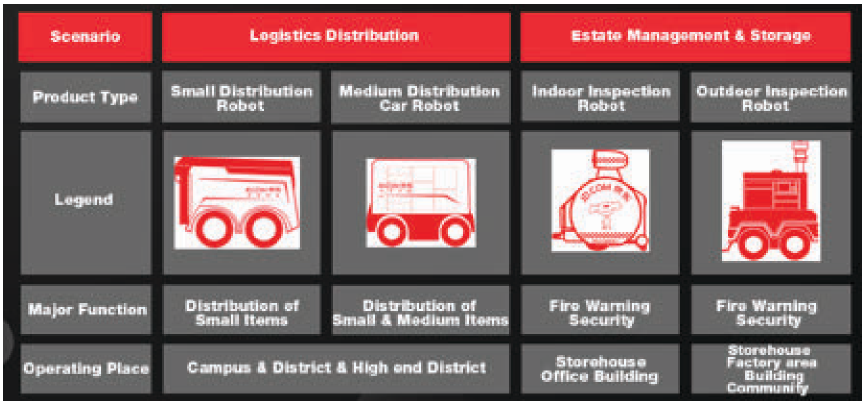


Figure 4. The use cases of the JD autonomous driving vehicle.



Scenario	Logistics Distribution		Estate Management & Storage	
Product Type	Small Distribution Robot	Medium Distribution Car Robot	Indoor Inspection Robot	Outdoor Inspection Robot
Legend				
Major Function	Distribution of Small Items	Distribution of Small & Medium Items	Fire Warning Security	Fire Warning Security
Operating Place	Campus & District & High end District		Storehouse Office Building	Storehouse Factory area Building Community

To be effective, last-mile delivery must determine the best route to dispense the most parcels. In a city, the best route is often *not* the shortest route, and road conditions constantly change over time.

As Figure 3 illustrates, the autonomous driving vehicles have been deployed and well tested in Beijing, Xian, and Hangzhou. Cities recently added include Tianjin, Guangzhou, Shanghai, Shenzhen, Changsha, Chengdu, Wuhan, and Suqian. User scenarios have also expanded from university campuses, to village areas, municipal areas, as well as industrial and institutional parks. Due to this wider reach, operations centers will be setup in Beijing to serve the north, and Changsha to serve the south.

However, there are still many challenges for future fleet expansion; for example, deployment is currently not available to truly open, more rural environments and human interaction is still required to periodically adjust the autonomous driving path. In addition, the effort to build a fully autonomous driving vehicle network is heavily depended on how the technology evolves, moving from cloud computing to edge computing, the capability of sensor networks, and the level of intelligence in artificial intelligence. Moreover, laws and regulations covering autonomous vehicles on the open road can vary dramatically.

There are many pilot deployments extending the last-mile delivery programs, and as a result many customized last-mile delivery vehicles have joined the general-purpose fleet, such as the smart shopping cart to enhance the supermarket shopping experience; an model used for datacenter inspections; moving demo center vehicles used at conferences, and so on. As shown in Figure 4, JD does not intend to solve all the technology, business, and regulation limitations of these vehicles; instead, the focus is on carefully designing user scenarios that best fit the strengths of the current technology.

Last June, JD's autonomous driving vehicles hit the open road in Beijing's Haidian District (Figure 5). These courier robotic vehicles can perform 360-degree environmental monitoring, automatically avoid roadblocks and pedestrians, and can react to traffic lights. It can independently stop at a distribution point, send delivery information to the user, and the user can pick up the packages through face recognition, input code, or clicking on the JD app link from their mobile phone. The vehicle can store up to 30 containers, and can travel 15k/h. As a result, last-mile deliveries can upgrade from 100–200 orders a day, to 1,000 orders per day.

Production difficulties still persist: the fault tolerance of the object recognition technology, the business processing for package returns from the end user to the warehouse, and the reliability of the vehicle (especially rounding corners) are three major challenges that must be resolved. In the early production period, human interaction may still be necessary. However, we

Figure 5. A JD courier robot on a road in Haidian District, Beijing.

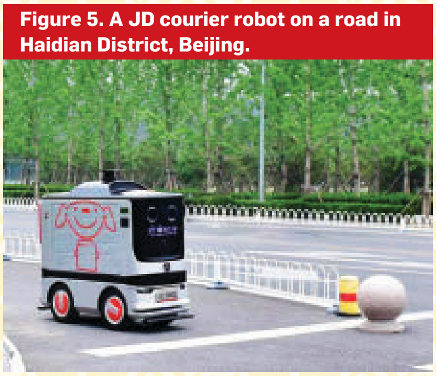
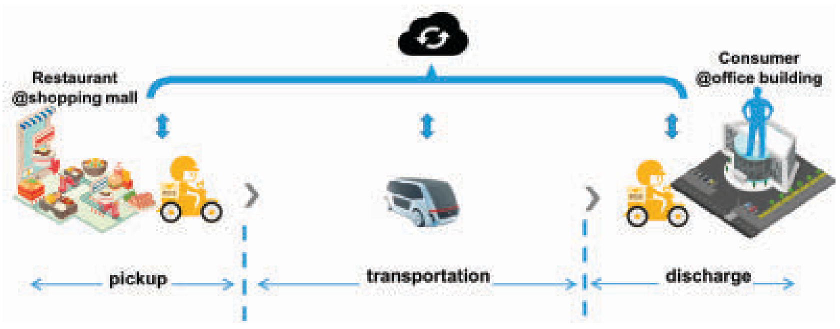


Figure 6. Diagram of an on-demand delivery for Meituan's autonomous vehicle.



believe when more road test data is accumulated and analyzed, the accuracy and efficiency of these vehicles will be fully achieved.

How Meituan Uses the Vehicles in Delivery Scenarios

Meituan is the world's largest e-commerce platform for local services. Meituan's service covers over 200 categories, including catering, on-demand delivery, car sharing, bicycle sharing, hotel and travel, movie, entertainment and lifestyle, and spreads over 2,800 counties, districts, and cities in China. In 2017, Meituan served 310 million active consumers and 4.37 million active merchants on the platform.

One of Meituan's services is its on-demand food delivery known as Meituan Waimai. By May 2018, Meituan Waimai was delivering 21 million orders per day and had hired 600,000 food-delivery riders. The service is usually within three kilometers, with tight time limits of 30 minutes. The fulfillment process includes three phases: 1. The courier goes to the restaurant to pick up the food, usually by walking through a shopping mall to get to the restaurant; 2. The courier transports the food next to consumer's building; 3. The courier walks or takes an elevator to the consumer (See Figure 6). In practice, each phase takes approximately one-third of the total delivery time.

Different phases need different types of vehicles. In phase 1 and phase 3, we use an indoor robot, as shown in Figure 7. This robot is 0.5m by 0.8m; its small size allows for easy entrance to shopping malls and office buildings. It does localization based on WiFi fingerprint and vision SLAM. It can also communicate through Zigbee to an elevator control module, thus can go up and down the buildings. The robot receives order information from the cloud scheduling system, runs to the merchant following the scheduled route, opens the top cover automatically so the merchant can put the food inside. When approaching its destination, the robot sends a text message to the user's mobile app, and then the user can pick up the food using the password code included in the text message.

In phase 2, a larger and faster autonomous delivery vehicle is used for street transportation, as shown in Figure 8. The vehicle measures one-meter wide and two-meters long, with maximum speed of 40km/h and maximum load of 10 orders. It uses the same technology as an autonomous passenger vehicle, including lidar, camera, and GNSS receiver. It can detect pedestrians, bicycles, automobiles, and other obstacles, and can also react to traffic lights.

Figure 7. Meituan's indoor delivery robot.

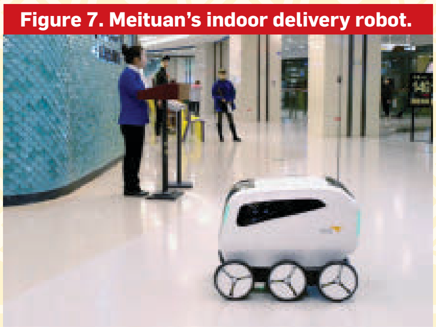
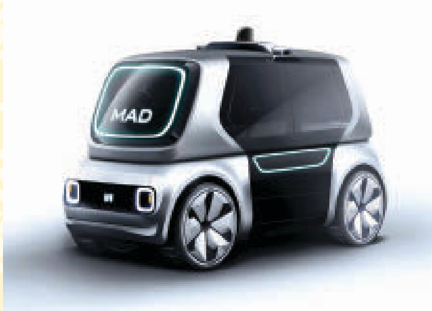


Figure 8. Meituan's outdoor delivery vehicle.



Challenges and Opportunities

There are many challenges for the large-scale deployment of autonomous delivery vehicles: The technology is not mature yet, the entire ecosystem must be further developed to make it more reliable, and the costs much shrink.

Moreover, our living infrastructure is not yet ready for autonomous driving. Many communities have locked or gated entrances, which require manual operation using a key or access card. Many buildings have revolving or swing doors, which are easy for humans to use but very difficult for robots. Elevators are rarely robot-ready. In fact, we must talk to building owners to get a permit to install a communication module in every elevator. These frustrations must be handled before we can fully enjoy autonomous vehicle deployment.

Government regulation of autonomous vehicle is a critical concern. While these delivery vehicles run at a fairly slow speed, most regulators consider "slow" a grey area, fitting between high-speed passenger vehicles and bicycles. Government regulations are not ready to handle pure level 4 (driver-free) vehicles like autonomous delivery vehicles. What happens if the autonomous vehicle is involved in an accident or a traffic violation? Who is responsible? Closed environments such as common in last-mile delivery can be used as pilot scenarios for learning that enable more complex, open road scenarios. This suggests two autonomous driving vehicles development methodologies: find a killer-level use case to drive the business model; or find the best technology and build the ecosystem.

Fortunately, this somewhat immature technology is acceptable for slow-speed delivery vehicles. The lack of infrastructure support may prevent us from mass deployment in some situations, but there are still many suitable scenarios for first-stage deployment. As for government regulation, the Chinese government is among the most supportive for high-tech innovations like autonomous vehicles.

The Future of Autonomous Delivery Vehicles

China's e-commerce boom brings a huge volume of logistics demand, both for express package delivery and for on-demand food delivery. The last-mile delivery is a perfect use case for autonomous driving technology.

Large-scale deployment of autonomous vehicles still depends on technology maturity and governing regulations. Nevertheless, these issues are not showstoppers. There are many pilot scenarios with government support, helping the industry step into the water of autonomous driving in order to accumulate the data and real-world experience needed to improve its technology. C

References

1. Levine, S. What it really costs to turn a car into a self-driving vehicle; https://qz.com/924212, 2017.
2. Xinhua News. Chinese express firms deliver over 40 bln parcels in 2017, 2018; https://bit.ly/2QnioCi
3. Xinhua News. China's express delivery sector prepares for post-holiday bonanza, 2018; http://www.xinhuanet.com/english/2018-02/24/c 136996545.htm.

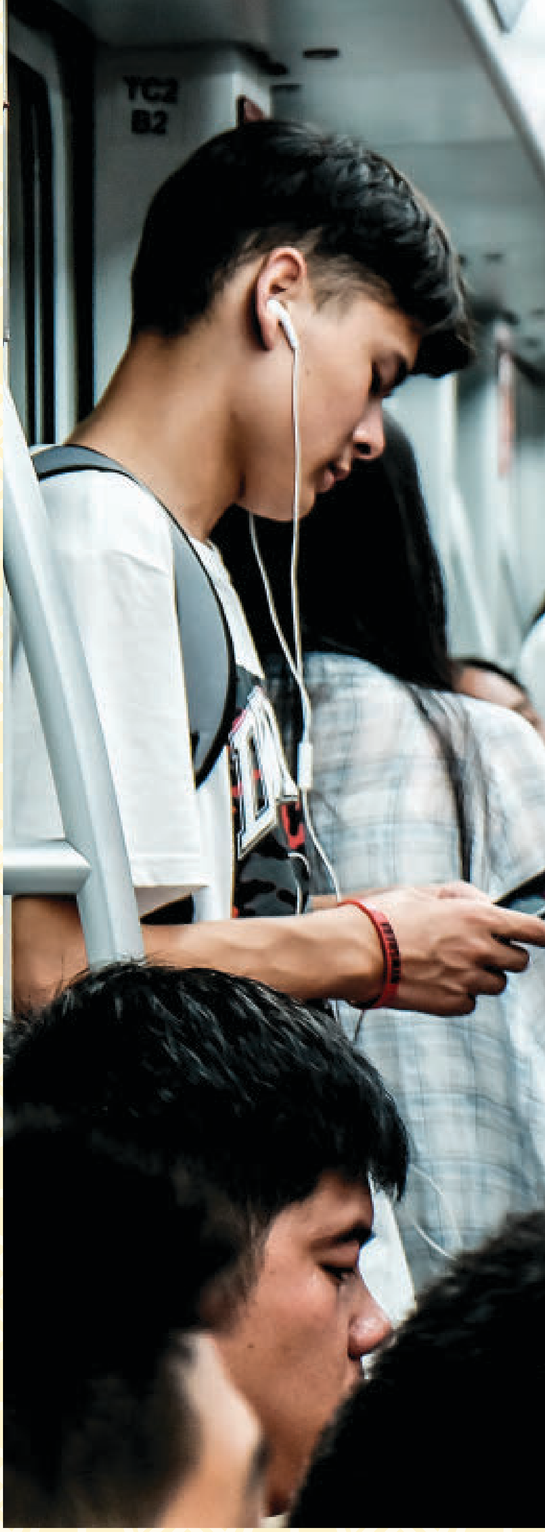
Huaxia Xia is Scientist and General Manager of the Autonomous Delivery Department at Meituan, Beijing.

Haiming Yang is Chief Architect at JD CTO Group, Beijing.

DOI:10.1145/3239554

BY YUE ZHUGE/HULU BEIJING

Video Consumption, Social Networking, and Influence



REVENUE FROM CHINA'S online entertainment market reached approximately $200 billion this year.[a] It is not surprising that China's video market is comparable to the U.S.;[b,c] in fact the number of online video users in China is 2.5 times more than that of the U.S. (that is, 212 million U.S.-based users[d] compared to 579 million users in China).[e] Due to advancements in broadband and mobile technology, online video is the fastest growing area for China's Internet, with a growth of around 50% over the past five years.[f]

The landscape of China's online video industry has as many similarities as differences with the U.S., presenting extremely interesting observations and insights. This article provides an overview of the market, dominant players, and business models, as well as presents intriguing product nuances and technical advances in this area.

Like the U.S., there are two major categories of services for online video: the head and the tail. The heads are the premium players that stream

[a] Statista; https://www.statista.com/statistics/237772/value-of-the-chinese-entertainment-and-media-market/

[b] Statista; https://www.statista.com/statistics/278574/revenue-of-chinese-online-video-industry/

[c] Statista; https://www.statista.com/statistics/459396/digital-video-revenue-digital-market-outlook-usa/

[d] Statistia; https://www.statista.com/topics/1137/online-video/

[e] Statistia; https://www.statista.com/statistics/279537/number-of-online-video-users-in-china/

[f] Revenue Growth of China's Online Video Industry; http://www.iresearchchina.com/content/details7_44334.html

PHOTO BY CHUTHARAT KAMKHUNTEE/SHUTTERSTOCK.COM

copyrighted shows and movies. They are the Netflixes and Hulus of China. The tails present professionally generated content (PGC) and user-generated content (UGC) for different market segments. They are the YouTubes and Snapchats of China.

The Head: Premium Video Platforms

The top three players in premium online video are iQiyi, Tencent Video, and Youku. These companies are affiliated with Baidu, Tencent, and Alibaba, respectively. Statistics from QuestMobile, China's big data services provider, shows that both Tencent Video and iQiyi recorded around 500 million monthly active users by the end of 2017, and around 300 million for Youku. The iQiyi video platform, that went public on NASDAQ last March, also leads in total watch hours.[g]

Unlike the premium video services in the U.S., the major players in China all started with free services supported by advertising. Total ad revenue for online video was approximately $10 billion, catching up to ad revenue generated by commercial television.

However, we have seen a huge take off in subscriptions over the past two years, when users started gravitating toward (and paying for) platforms that were ad-free and offered additional features such as access to higher-quality video and member-only original content. iQiyi counted 60 million subscribers as of Feb.

[g] Prospectus iQIYI Inc; https://bit.ly/2ouHhzO

Due to the prevalence and popularity of video streaming, both infrastructure and application companies in China have invested heavily in video technology.

2018, and Tencent Video has over 40 million. By comparison, Netflix had about 55 million U.S. subscribers and 63 million international subscribers as of January 2018.[h]

Payment practices have been forming rapidly among the middle-class and young Internet users in China over the past few years. The major driving forces behind this movement include a concerted crackdown on pirated content, affordable prices, and, most importantly, the ease of online payment.[i]The monthly subscription price is between 20 RMB ($3.16) to 40 RMB ($6.32) for each of the three services. The total market size of Internet video subscription services has increased dramatically, from about $63 million in 2012, to $2.1 billion in 2016, and an estimated $11.5 billion in 2022.

Unlike premium content distributions in the U.S., many of the TV dramas and movies are non-exclusive, available on all three major services and elsewhere once they aired on TV or in movie theaters. Exclusive content is usually far more costly. All services invest heavily in copyrighted movies and TV shows, resulting in very high production prices for this content. Platforms often must make a calculated bet on what shows will prove popular, and make an offer before production. According to its prospectus, iQiyi's annual content cost is about $1.9 billion, while the other two platforms spent nearly double that. These purchases are far beyond subscription fees and advertising income. According to its public filing, iQiyi lost $169 million Q1 2018,[j] and the other two services also lost similar sums.[k] This situation will continue for the next few years.

To reduce costs, and to stand out among their peers, all three premium services have started to make original content exclusive on their own platform.[l] The lead player for original shows is iQiyi, although all three had different hits. Unlike U.S. platforms where TV dramas tend to reign supreme, variety shows garner a greater audience in China. According to a recent *Wall Street Journal* article,[m] "The Rap of China," a 12-episode hip-hop rap competition reality series created and shown by iQiyi targeting younger audience, has become "China's most popular entertainment program in 2017." The show attracted 2.7 billion views during its run from late June to early September. Short videos clips gleaned from the show were watched eight billion times on the social media platform Weibo.

Another fact about China's premium online video services that differs from their U.S. counterparts is that they all participate in PCG and UGC short video markets. But, as we will discuss, other emerging players are increasingly dominating these segments.

The Tail: UGC and PGC Video Platforms

With more than 100 players and 400 million users in 2017, the short video landscape in China is hugely dynamic, and far from settling down. The user base is huge, fast growing, and extremely active. There are several major players, and most of them came into prominence over the last year or two.

Unlike the U.S. market, the initial dominant short video platforms were the premium players, like iQiyi and Youku. For example, YouKu claims to have invested approximately $1.6 billion in user-generated content since 2015.[n] They are modeled after YouTube and had a large number of viewers watching a mixture of premium and UGC content. Their short-form videos include movie clips and music videos as well as free-form user-created content, and they provide channels created for professional content producers.

[h] Recode: Netflix now has nearly 118 million streaming subscribers globally; https://www.recode.net/2018/1/22/16920150/netflix-q4-2017-earnings-subscribers

[i] CNN: China's big streaming shift: Paying instead of pirating; http://money.cnn.com/2018/01/24/technology/china-streaming-music-video/index.html

[j] iQiyi First Quarter Financial Results; https://bit.ly/2LfXGRK

[k] Forbes; https://bit.ly/2KOhQ9v

[l] iResearch: 2017 Report on Original Video Productions in China; http://report.iresearch.cn/report_pdf.aspx?id=3088

[m] *Wall Street Journal*; https://on.wsj.com/2hGe7Nk

[n] Youku making $1.6 billion investment in UGC; http://www.chinadaily.com.cn/business/2015-08/07/content_21525850.htm

However, in the past two years, we have witnessed the phenomenal growth of several mobile short video apps not associated with the premium players. Thanks to their ease of use, these apps became super popular and prevalent, penetrating a huge number of users in massive areas of China (see Figure 1). We saw a 311% increase in short video traffic in Q3 2017 compared to one year earlier.[o] Analysts labeled the China Internet era of 2017 as "the year of short videos."

New popular short video services include Kuaishou, Huoshan, Xigua, Douyin, Miaopai, Meipai, Weishi, and many more. The most popular of them have monthly active users in the 100 million–200 million range. The app experience is mostly a flow of mobile video feeds, with videos running seconds to minutes long. The app provides good tools for users to shoot, edit, beautify, and add special effects to the videos. These video apps can be divided into two categories: those more like the "dubsmash" mobile app, where users can record their own video dubbing over music, and those more like Snapchat with free-form videos. All video apps provide strong discovery and follow functionalities, encouraging interaction and social connections among users.

More specifically, videos in Xigua are mainly PGC content and video clips running 1–5 minutes long. Content in both Huoshan and Kuaishou are short UGC videos of about 15 seconds. Many everyday users, especially those in the rural area and small cities, record short videos and share them via these apps.[p] The apps provide a method for users to present themselves, compete to gain fans and eyeballs, and eventually profit from the viewership. Kuashou, with daily active users of close to 100 million, is the current leader in this category. Using a slightly different format, Douyin has recently received considerable attention and gained a significant user base, claiming more than 150 million monthly active users. Douyin, as well as competing apps like Weishi, allows users to create and share short music videos using provided templates.

Another large set of online video services focuses on end user live-streaming: they provide live-broadcasting capabilities for consumers, popular pop idols, and the general public. According to Pandaily, eight live-streaming platforms in China raised approximately $11.6 billion in the first half of 2018. The leading platforms—Huya Inc. and Douyu TV—account for nearly 70% of the total.[q]

Just like all premium players provide UGC, all short-video players also working on live streaming. The situation is highly dynamic; the landscape was quite different several months ago, and it is likely to be very different a few months from now.

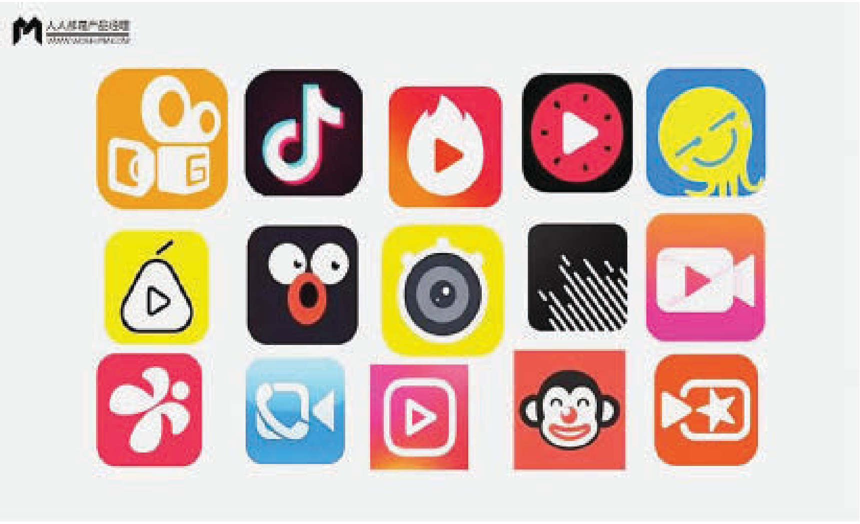
Still experimenting with business models, the new mobile video apps make money through advertising, affiliate marketing to e-commerce, gifting, and other creative methods. The top "VIPs" on these platforms may have millions of fans, and they profit by advertising or selling goods. Brands also start to create channels on these platforms. According to the China's Short Video Industry Report from iResearch, the revenue from the short video arena reached $860 million in 2017, and is expected to exceed $4.5 billion by 2020.[r]

We are just beginning to see the great potential of China's short video market; it will evolve in the coming years. There is a trend of going vertical, with different players specialized in a domain or a demographic. With fierce competition within China, many of these short video apps made their ways overseas, in particular, to other Asian countries. For example, Kuaishou was the Number One video app in the Korean app store in November 2017. According to an April 2018 report from 36kr, almost half of the popular short video apps in Asia were "made in China."[s]

Mobile and Social via Video

Large screens, including TV and other OTT television devices, are heavily regulated in China. Due to the high penetration of mobile devices, the usage of online video tilts strongly toward mobile. Even for the premium users, the dominant media preference is mobile and personal computers. According to iQiyi, it has on average 421 million monthly active users (MAU) on mobile and 424 million MAU on PCs in Q4 2017, while TV and OTT devices are negligible. Many short videos applications are

Figure 1. A sample of logos for popular short video apps in China.



[o] Short Video Report from iFeng; http://tech.ifeng.com/a/20180104/44831545_0.shtml

[p] QuestMobile: China Lower Tier City Post-90s' Mobile Life; https://www.questmobile.com.cn/blog/en/blog_138.html

[q] Pandaily: 2018 Report on Live-streaming; https://bit.ly/2mbdhHt

[r] iResearch: 2017 China's Short Video Industry Report; https://bit.ly/2zA2Mac

[s] 36kr: Made in China; http://36kr.com/p/5130958.html

Analysts have labeled the China Internet era of 2017 as "the year of short videos."

designed for and used exclusively on mobile devices.

Most people using online video—premium or UGC—are quite young. According to iResearch, about 83% of iQiyi's mobile users in August 2017 were younger than 35. The demographic is similar for short videos, with more than 80% of users younger than 35.

A major difference between the premium video products in China and in the U.S. is the emphasis on social features. If you look at the video discovery or display pages carefully, you will find sharing buttons on every page, linking to every possible social network. Excerpts from the longer shows, like songs or jokes, are often very popular clips for users to share. These services also provide interaction such as screen bullets, comments and love buttons, to improve the social experience while watching (Figure 2).

Another type of social interaction for the premium video service is with media stars. iQiyi PaoPao is a place for celebrities to interact with their fans. Hundreds of movie and TV stars have their home pages set there, with 20 million fans following the top stars. iQiyi claims 600 million active users in PaoPao, who are proven to be more sticky, watching 20% more videos on average than other users.

For short video apps, a significant emphasis is on social functions. These short video apps provide many ways for interaction, including discovering people with similar interests, follows, claps, love, comments, and gifting. The app platforms compete by paying large sums of money to the popular "VIPs," encouraging them to set up channels on their platform.

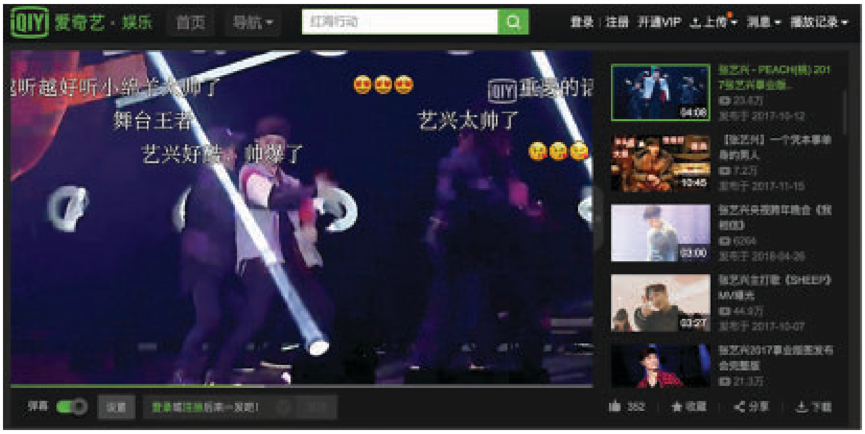
For a short video app user, the number of fans defines success, and the top "VIPs" have more than 30 million fans. To become a "VIP," users compete to create attractive, frequent content, and stimulate excitement with their fans. A slogan popular among the top users is "300 clips a day!"

Technical Opportunities and Challenges

With the prevalence and popularity of video streaming, both infrastructure and application companies in China have invested heavily in video technology. Generally, a video startup time of less than two seconds is considered "good;" of course, one needs to consider the video quality, device type, network situation, and other factors to make a specific judgment. According to a 2017 report from China Broadband Development Alliance, the average video startup time (VST on broadband) was between 0.6 to 1 second, which is better than the world standard.

What is unique about China's Internet market that fosters technical advancements and innovation? We can call out a few examples: mobile dominance, the huge number of users and available user data, the massive scale of user-generated content, regulatory requirements, ever-changing

Figure 2. A typical page from the iQiyi website with bullet screen.



听越好听小绵羊太帅了

舞台王者

艺兴好酷，帅爆了

艺兴太帅了

user interests, and extremely fierce competition.

Features more suitable for mobile viewers, like usage scenarios that support watching movies while commuting on public transit, add to the success of these platforms. For example, all premium video platforms provide offline viewing as a default feature, while short video apps provide offline information such as news for people to view while not connected.

All video providers rely on personalized recommendation; it is especially important for short videos as the main method to discover videos of interest. A commercial recommendation system employs large-scale machine learning on real-time streaming data and tries to optimize metrics such as the click-through rate, time spent, and user retention. With the availability of large amounts of data from both first- and third-party vendors, one thing that distinguishes recommendation systems on China's Internet from the rest of the world is its sheer scale: Alibaba claims its machine-learning platform—eXtreme Parameter Sever (XPS)—processes 10B samples and 100B features daily,[t] while the Toutiao platform claims tens of billions of features and billions of vectors.[u] By the same token, strong recall technology is developed to select the top few thousand results from millions of potentially low quality or redundant user-generated content.

In general, machine learning is used to annotate, classify, and analyze video content, and to build user profiles based on the user's geological location and browsing history. It then uses such information to match a user to videos that reflect their interests. Instead of a 'pull' or 'subscribe' model, many of the Chinese short video apps 'pushes' the relevant content to viewers' home pages. Since most Chinese viewers are quite receptive to pushed information, recommendation technology is very effective, and users can indulge in content they like for hours a time.

For short forms, users upload videos to platforms every day, which may contain inappropriate content such as pornography, content that infringes on copyrights, and duplications. Although many of the media companies have thousands of human editors, it is difficult to manually examine all contents in real time. To react quickly to the market, video companies developed adaptive machine models that work together with human editors to prescreen and filter out potentially problematic videos. Necessity also prompted the rise of high-tech providers such as SenseTime that specialize in image and video reviewing technology.[v]

Another important application of AI is to add special effects to user-generated videos. For example, Chinese users often want features such as beautifying faces, adding special costumes, and changing backgrounds. To do so, one needs to detect facial key points and perform highly accurate face and hair segmentation. The related technologies, such as style transfer and object segmentation, are active areas of research in computer vision. Significant progress has been made in recent years using deep learning. In addition, augmented reality (AR) also has a lot of interesting usage on video, like blending fake objects with video backgrounds. These technologies help users create more interesting videos that are better fit for sharing.

To monetize the contents, all video platforms use computational advertising to display ads targeting users based on their personalized interests. In addition, many innovative video technologies are used in video ads, for example, ads overlaid on the videos, ads inserted in live broadcasts, ads on bullet screens, 360 ads integration with direct sell, and more. For example, one video ad service provider—Video++—claims to have more than 9,000 clients using their technology to integrate direct sell ads into streaming video.[w]

Mobile video apps have also presented interesting technical challenges to machine learning and computer vision. Instead of inventing new algorithms, the practical solution was to apply the PC applicable algorithm to mobile. For example, to improve speed, or to use local data, a ML model inference for object recognition may need to be done on a mobile phone. In such circumstances, the size of the ML model must be small enough to fit, and even better, optimized to the phone hardware.

China's streaming video companies continue to explore many other ways to innovate with technology. For example, Tencent tried to use robots to write articles for live news, and Youku tested auto caption translation. Technology is also used to predict user reactions and suggest content investment in media. iQiyi, for example, claimed to have casted actors in their original shows based on AI predictions.

Summary

Online video, or online media in a larger setting, is one of the best places where technology creatively meets user experience. Through online media, there are infinite ways to connect hundreds of millions of users to billions of pieces of information. This is an exciting time for China's online video market, as it has just realized mass-market adoption. We look forward to the new technological innovations to come.

[t] Alibaba's eXtreme Parameter Server; http://m.sohu.com/a/210104407_473283

[u] CSDN: Recommendation System in Toutiao; https://bit.ly/2NJVS5n

[v] Sensetime: Video Content Review; https://www.sensetime.com/core#2

[w] Buying while watching videos; https://www.huxiu.com/article/161897.html

Yue Zhuge is Vice President of Research and Development and General Manager at Hulu, Beijing.

DOI:10.1145/3239556

BY YUTONG LU/SUN YAT-SEN UNIVERSITY, NSCC-GZ,
DEPEI QIAN/BEIHANG UNIVERSITY, SYSU,
HAOHUAN FU/TSINGHUA UNIVERSITY, NSCC-WX,
WENGUANG CHEN/TSINGHUA UNIVERSITY

Will Supercomputers Be Super-Data and Super-AI Machines?

HIGH-PERFORMANCE COMPUTING (HPC) plays an important role in promoting scientific discovery, addressing grand-challenge problems, and promoting social and economic development. Over the past several decades, China has put significant effort into improving its own HPC through a series of key projects under its national research and development program. Development of supercomputing systems has advanced parallel applications in various fields in China, along with related software and hardware technology, and helped advance China's technological innovation and social development.

To meet the requirements of multidisciplinary and multidomain applications, new challenges in architecture, system software, and application technologies must be addressed to help develop next-generation exascale supercomputing systems.

Supercomputer Development in China

The first supercomputer developed in China was Yinhe-I in 1983, with 1MFlops peak performance, by the National University of Defense Technology (NUDT). China has since continued its supercomputer development.

Three major teams in China—Tianhe, Sunway, and Sugon, like IBM, Cray, and Intel in the U.S.—have developed a series of domestic supercomputing systems, including Dawning 4000A (2005, 11.2TFlops); Tianhe-1A (2011, 4.7PFlops, number one in the TOP500); Sunway BlueLight (2011, 1PFlops); Tianhe-2 (2013, number one in the TOP500 six times); and Sunway TaihuLight (2016, number one in the TOP500 four times). Chinese supercomputers have adapted multiple architectures, including vector, SMP, ccNUMA, MPP,

cluster, heterogeneous-accelerated, and many-core. Their developers have thus acquired rich knowledge of supercomputing hardware and software and trained a large number of engineers along the way.

In the years since the Yinhe-1 system in 1983, China has achieved the leading position in supercomputer development worldwide. For example, Tianhe-2 and Sunway TaihuLight held the top position in the TOP500 from 2013 to 2017. At the same time, the number of HPC systems in China increased dramatically, exceeding the number of HPC systems in the U.S., as of the June 2018 TOP500 ranking. And Chinese HPC manufacturers Lenovo, Sugon, Inspur, and others have claimed significant shares of the market for HPC systems and high-end servers.

China's leading-class supercomputer systems. Two 100PF computers—Tianhe-2 and Sunway TaihuLight—were developed with support from the National High-Tech R&D Program in the country's 12th Five Year Plan. The first stage of Tianhe-2 was completed in early 2013, delivering peak performance of 55 petaflops and Linpack performance of 33.9 petaflops. It was a hybrid system consisting of Intel Xeon processors and Xeon Phi accelerators and claimed the top position in the TOP500 six consecutive times, from June 2013 to November 2015. In 2017, NUDT deployed the new 128-core Matrix 2000 processor and applied it to upgrading Tianhe-2. The upgraded system, called Tianhe-2A, delivered 100.68 petaflops peak performance and approximately 61 petaflops Linpack performance.

The second 100 petaflops system is the many-core-based Sunway TaihuLight that delivers 125 petaflops peak performance and 93 petaflops Linpack performance. Sunway TaihuLight was implemented with the homegrown many-core processor SW26010—a 3Tflops chip with 260 cores—ranking in first place four times, from 2016 to 2017, in the TOP500. Key technologies adopted by the TaihuLight system include highly scalable heterogeneous architecture, high-density system integration, a high-bandwidth multi-level network, highly efficient DC power supply, and customized water cooling. It also represents a new milestone in China's HPC history for being a 100PF system implemented completely with homegrown processors.

Two 100PF systems have been installed at two national supercomputing centers, with Tianhe-2A at the National Supercomputing Center in Guangzhou and Sunway TaihuLight at the National Supercomputing Center in Wuxi, respectively. Moreover, extra effort has gone toward increasing the user population and developing applications.

Efficient HPC software stack. With development of domestic supercomputing systems, China has established a self-controllable system software stack covering basic drivers, operating system, compilers, communication software, basic library, parallel program-

China must rely on self-controllable technologies, especially for basic hardware components like processors, memory, and interconnect networks, to build an exascale system.

ming environment, parallel file system, resource management, and scheduling system, thus providing a comprehensive capability for large-scale system construction and performance tuning.

The Tianhe-2 software stack consists of four components: a system environment, an application-development environment, a runtime environment, and a management environment. The system environment consists of the 64-bit Kylin OS and H2FS parallel file system and a resource-management system. Various job-scheduling policies and resource-allocation strategies have been implemented so system throughput and resource utilization can be enhanced. The application-development environment supports multiple programming languages, including C, C++, Fortran 77/90/95, a heterogeneous programming model called OpenMC, and the traditional OpenMP and MPI programming models. The THMPI is an updated version based on MPICH over the Tianhe Net communication protocols, able to deliver 12GB/s P2P bandwidth at the user level. The runtime environment consists of the parallel numerical toolkit for scientific applications, a scientific data-visualization system, and an HPC application service and cloud-computing platform. It provides runtime support to multiple fields, including scientific and engineering computing, big data processing, and high-throughput information service.

To support both HPC and big data applications, the Sunway TaihuLight also includes highly efficient scheduling and management tools and a rich set of parallel programming languages and development environments for application research and development. A two-level "MPI+X" approach helps devise the right parallelization scheme for mapping the target application onto the processes and threads that utilize more than 10 million of the system's cores. The 260-core SW26010 processor consists of four core groups (CGs), with each CG including one management processing element (MPE) and one computing processing element (CPE) cluster with eight-by-eight CPEs. Each CG usually corresponds to one MPI process. Within each CG, the system has two options: one is Sunway OpenACC, a customized parallel compilation tool that supports OpenACC 2.0 syntax targeting the CPE cluster; the other is a high-performance yet lightweight thread library called Athread that exploits fine-grain parallelism. With a byte-to-flop ratio five to 10 times less than other top-five systems, the system needs extraordinary memory-related innovation to deal with the memory wall to scale its simulation capability with 125 Pflops computing performance. It also needs software migration to such an architecture, with radical changes in both compute and memory hierarchy. For each CPE, instead of hardware L1 cache, the system includes user-controlled 64-KB local data memory (LDM) that completely changes the memory perspective for programmers.

Effect on HPC industry. The rapid development of China's supercomputing systems has benefited from the continuous support of several national five-year plans, as well as the country's economic development and national strength. The systems support scientific research, technological breakthroughs, and an industrial revolution while promoting development and expansion of the IT server sector. Vendors Inspur, Lenovo, Huawei, and others have taken advantage of research and development of domestic HPC kernel technologies, including systems integration, storage architecture, interconnection technology, optimization techniques, system testing and benchmarks, and application technologies. At the same time, a large number of HPC hardware and software engineers have been trained for IT companies in China, including Alibaba, Baidu, and Tencent. The Chinese IT industry has also benefited from the technological innovation resulting from supercomputer development, including high-performance clusters, HPC-enabled cloud computing, distributed computing, and application optimization.

Key Applications and Beyond

Along with the rapid development of hardware systems, major breakthroughs have been made in application development based on the new supercomputers, covering both traditional HPC domains like climate, seismology, computational fluid dynamics, and fusion and relatively new applications like big data and artificial intelligence (AI).

Atmospheric modeling. Large-scale simulation of the global atmosphere is one of the most computationally challenging problems in scientific computing. The Tianhe-1A hybrid CPU-GPU system launched a continuous development effort toward highly scalable atmospheric dynamic solvers on heterogeneous supercomputers, achieving sustained double-precision performance of 581 Tflops on Tianhe-1 by efficiently using the CPU and GPU resources on 3,750 nodes. The work was later extended to the Tianhe-2 system, scaling to the 8,644 nodes of Tianhe-2, achieving 3.74 Pflops performance with CPUs and MICs.

In 2016, the solver effort migrated to the Sunway TaihuLight supercomputer, with a highly scalable fully implicit solver for cloud-resolving atmospheric simulations. The solver supports fully implicit simulations with large time steps at extreme-scale resolutions and encapsulates novel domain decomposition, multigrid, and ILU factorization algorithms for massively parallel computing. With both algorithmic and optimization innovations, the solver scales to 10.5-million heterogeneous cores on Sunway TaihuLight at an unprecedented 488-m resolution with 770-billion unknowns, sustaining 7.95 PFLOPS performance in double-precision with 0.07 simulated-years-per-day. Considered a major breakthrough, it won the ACM Gordon Bell Prize in 2016, the first time in 29 years Chinese researchers were so recognized.

Earthquake simulation. Earthquake simulation is another traditional major challenge for supercomputers. Starting with AWP-ODC and CG-FDM codes, Chinese researchers have developed nonlinear earthquake simulation software on Sunway TaihuLight, winning the ACM Gordon Bell Prize in 2017. While TaihuLight delivers an unprecedented level of computing power (three times that of Tianhe-2 and five times that of Titan), its memory system is relatively modest. Total memory size is similar to other systems (such as Piz Daint and Titan, two GPU-based systems), with a significantly lower byte-to-flop ratio, as compared to 1/5 in other heterogeneous systems and 1/10 in the K Computer. Such a system represents both high potential and notable challenges for scaling scientific applications. Especially for earthquake simulation, which requires both a large amount of memory and high memory bandwidth, breaking the memory wall becomes the top challenge. To resolve this bandwidth constraint, Chinese researchers have performed three notable optimizations: a customized parallelization scheme that employs the 10-million cores efficiently at both the process level and the thread level (to address the scale challenge); an elaborate memory scheme that integrates on-chip halo exchange through register communication, optimized blocking configuration guided by an analytic model, and coalesced DMA access with array fusion (to alleviate the memory constraint); and on-the-fly compression that doubles the maximum problem size and further improves performance by 24% (to further address the memory wall). The extreme cases demonstrate sustained performance greater than 18.9 Pflops, enabling simulation of the Tangshan earthquake through an 18Hz scenario with eight-meter resolution.

The Tianhe-2 supercomputer is installed at the National Supercomputer Center in Guangzhou.

Drug design. Virtual high-throughput screening is an established computational method for identifying drug candidates from a large collection of compound libraries, accelerating the drug-discovery process. When diseases and unknown viruses appear, it is especially useful for screening as many molecules as possible to help identify an effective treatment. The kernel algorithm is the Lamarckian Genetic Algorithm, a combined local search and genetic algorithm for efficient global-space coverage and local-search optimization. A typical data scale of 40 million molecules requires more than 800TB. With the need to handle approximately 40 million small files, the optimized design on Tianhe-2 takes advantage of the high throughput of the H2FS file system, I/O-congestion control, multi-stage task scheduling, task-pool management, asynchronous I/O, and communication to improve application scalability. The design is able to screen 35 million candidate drug molecules against the Ebola virus in 20 hours. The parallel efficiency from 500 to 8,000 nodes (1.6 million hybrid cores) is over 84%. Such computational capability demonstrates how Tianhe-2 is able to screen all known 40 million drug molecules against an unknown virus in a single day.

Large-scale graph computing. With increasing demand for graph processing, both Sunway TaihuLight and Tianhe-2 have earned Graph500 breadth-first-search (BFS) scores. Sunway TaihuLight ranks second at 23,755.7 giga-traversed edges per second (GTEPS), and Tianhe-2 ranks tenth.

In addition, the graph-processing framework ShenTu was developed on the Sunway TaighuLight, allowing users to write vertex-centric graph-processing programs and scale out the computing to the whole Sunway TaihuLight machine. The framework can support such graph algorithms as PageRank, Shortest Path, BFS, and K-Core with just 20 lines of code. It can process graphs with 10 trillion edges in tens of seconds. For example, ShenTu can complete one round of page ranking in 21 seconds on a 12-trillion-edge real-world Web graph, an order-of-magnitude performance improvement on graphs that are one order of magnitude larger than prior work.

Deep-learning applications. In addition to traditional applications, efforts are under way to explore the potential of training complex deep neural networks (DNNs) on these heterogeneous supercomputers. For example, there is a highly efficient library on swDNN on Sunway TaihuLight for accelerating deep-learning applications. By identifying the most suitable approach for mapping the convolutional neural networks (CNNs) onto the 260 cores within the chip, swDNN achieves double-precision performance greater than 1.6Tflops for the convolution kernel, which is over 54% of the theoretical peak of the SW26010 processor. Parallel training is supported through swCaffe, a redesigned version of Caffe, for large-scale training on up to 1,000 Sunway nodes.

Some deep-learning applications run on Tianhe-2, including for tumor diagnosis, video analysis, and intelligent transportation. One application called "trade business of Guanghzou" supports 900 million deals annually.

The Chinese government is encouraging development of the kernel technologies, including high-performance processor/accelerator, novel memory devices, and interconnect networks.

Toward Next-Generation Systems

As of July 2018, the Summit supercomputer (powered by IBM POWER9 and Nvidia V100 processors) was ranked number one in the TOP500, achieving 122PF LINPACK performance and 3.3Exaops for data processing and AI applications at half precision. However, a number of planned systems will soon surpass it. In the past few years, several countries have targeted exscale computing, including ECP in the U.S., Post K in Japan, and EuroHPC in the E.U., aiming for breakthroughs in key technologies, including novel architecture, high energy efficiency, system software, and exascale applications. These efforts lead the way toward next-generation supercomputing systems.

China's exascale project. The key HPC project in China's 13th five-year research and development program was launched two years ago to pursue a two-step strategy for developing exaflops supercomputing. The first step aims to deploy three prototype exascale computers by the Tianhe team, Sunway team, and Sugon teams, respectively, pursuing novel architectures, kernel technology breakthroughs, and possible technical approaches for implementing future exascale systems. Carried out from 2016 to 2018, the projects were completed by the end of June 2018. The second step is to select two of the three to develop exascale systems by the end of 2020.

The project aims to develop self-dependent and controllable kernel technologies for exascale computing and maintain China's leading position in global HPC; develop a number of critical HPC applications and build a national software center, establishing an HPC-application ecosystem; and build a national HPC environment with world-leading resources and services.

The Chinese exascale system will aim to achieve the following specification: peak performance of 1EFlops, node performance greater than 10TFlops, memory capacity greater than 10PB, storage capacity of 1EB, interconnection network bandwidth greater than 500Gbps, Linpack efficiency over 60%, and energy efficiency greater than 30GFlops/W. Moreover, the system should include an easy-to-use parallel programming environment, monitoring and fault-tolerance management, and support for large-scale applications.

Our approach. Exascale computing must address unprecedented technical challenges worldwide, including the memory wall, communication wall, reliability wall, energy-consumption wall, and programming wall. A strategy of hardware and software co-design will thus be required. For example, new algorithms will be proposed and implemented with the target hardware features in mind. Resilience will be addressed through fault-tolerant hardware design and fast failure detection and recovery enabled by software.

China must also rely on self-controllable technologies, especially for basic hardware components like processors, memory, and interconnect networks, to build an exascale system. The Chinese microelectronics and IC industry is still relatively weak, thus calling for more basic research and technology development. Also, China must satisfy various complex application needs and deal with a huge and highly diverse market, thus calling for multiple design and development approaches. The current key HPC project relies on architectural innovation, technology breakthroughs, and hardware and software coordination to address these challenges. Novel architectures will be explored to address the requirement of the various applications. Engineering trade-offs will be necessary to balance metrics in power consumption, performance, programmability, and resilience. Technology breakthroughs will be pursued through comprehensive research efforts. Special attention will target application software.

The Chinese government is encouraging development of the kernel technologies, including high-performance processor/accelerator, novel memory devices, and interconnect networks. The effort toward self-controllable processor technologies include Sunway's SW many-core processor, NUDT's FT series CPU and Matrix series accelerator, and Sugon's X86 AMD-licensed processor. NUDT has developed its propriety interconnect network TH-Net with high bandwidth and low latency, making the TH-2 system efficient and scalable. The Sunway system also includes its own self-designed large-scale network, enabling the TaihuLight system to run efficiently on 10 million cores. More new technologies breakthroughs are still needed to support successful development of exascale systems.

The key HPC project also targets applications focusing on climate change, ocean simulation, combustion, electromagnetic-environment simulation, oil exploration, material science, astrophysics, and life science. A new computational model and algorithm will be designed, and the efficiency, scalability, reliability of the applications will be evaluated for future exascale systems.

The pervasive use of HPC has promoted development of large-scale parallel software. Chinese researchers are strengthening development of system software and application software for domestic hardware systems, aiming to establish the country's own HPC ecosystem.

Emerging big data and AI applications have also gained the attention of Chinese HPC research programs. The National Natural Science Foundation of China, the counterpart of the U.S. National Science Foundation, has launched an initiative in big-data science to research computational models, algorithms, and platforms for data analytics and processing. Related projects focus on such big-data-related fields as video processing, health and medicine, intelligent transport, finance, government administration, and intelligent education. And an upcoming national research initiative on AI will call for HPC support for AI applications. The scope of HPC applications will definitely broaden in the future.

The TaihuLight supercomputer is installed at the National Supercomputer Center in Wuxi.

Conclusion

Parallel computers and parallel applications have cross-pollinated each other in China for the past 15 years. The availability of leading-class supercomputers has stimulated the growth of parallel applications in a number of fields, an application- and technology-driven growth trend that will continue into the future.

How to maintain sustainable development toward the next generation of supercomputing in China is an open question. Though significant progress has been made in recent years, China is still behind Western countries in HPC in many respects. A long-term national plan on HPC is needed that would allow more systematic deployment of HPC research. A mechanism that would ensure sustainable development of the national HPC infrastructure must be established so the supercomputing centers do not have to struggle to find the money needed to run the supercomputers.

Exascale computing projects are being implemented in the U.S., Japan, and Europe, aiming to deliver exaflops computers in three to five years. Their effort is like mountain climbing. Climbers can enjoy the magnificent scenery only when they get to the top following their arduous journey. The Chinese HPC community is willing to work with the international HPC community to pursue the goal of exascale computing, sharing the experience and jointly attacking the grand challenges. HPC should not be a new kind of arms race but technology that benefits all people.

Chinese researchers also need to be aware of new technologies and applications. The emergence of big data and AI brings new challenges and opportunities to HPC. Supporting big data and AI with HPC while being rewarded by big-data- and AI-enabled technologies for HPC should drive coordinated and converged development of all three. All should take this opportunity to embrace this new exciting era of supercomputing. C

Yutong Lu is a professor of data and computer science at Sun Yat-Sen University and Director of the National Computing Center in Guangzhou.

Depei Qian is a professor and Dean of Data and Computer Science at Sun Yat-Sen University, Guangzhou and a professor of computer science and engineering at Beihang University, Beijing.

Haohuan Fu is professor in the Department of Earth System Science at Tsinghua University, Beijing, and Deputy Director of the National Supercomputing Center in Wuxi.

Wenguang Chen is a professor in the Department of Computer Science and Technology at Tsinghua University, Beijing.

DOI:10.1145/3233233

Article development led by acmqueue
queue.acm.org

How Google moved its virtual desktops to the cloud.

BY MATT FATA, PHILIPPE-JOSEPH ARIDA, PATRICK HAHN, AND BETSY BEYER

Corp to Cloud: Google's Virtual Desktops

OVER ONE-FOURTH OF Googlers use internal, datacenter-hosted virtual desktops. This on-premises offering sits in the corporate network and allows users to develop code, access internal resources, and use GUI tools remotely from anywhere in the world. Among its most notable features, a virtual desktop instance can be sized according to the task at hand, has persistent user storage, and can be moved between corporate datacenters to follow traveling Googlers.

Until recently, our virtual desktops were hosted on commercially available hardware on Google's corporate network using a homegrown open source virtual cluster-management system called Ganeti (http://www.ganeti.org/). Today, this substantial and Google-critical workload runs on Google Cloud Platform (GCP). This article discusses the reasons for the move to GCP, and how the migration was accomplished.

While Ganeti is inexpensive to run, scalable, and easy to integrate with Google's internal systems, running a do-it-yourself full-stack virtual fleet had some notable drawbacks. Because running virtual desktops on Ganeti entailed managing components from the hardware up to the VM manager, the service was characterized by:

- Long lead times to expand fleet capacity;
- Substantial and ongoing maintenance overhead;
- Difficulty in staffing the team, given the required breadth and depth of technologies involved;
- Resource waste of underlying hardware, given working hours for a typical Googler are 8–10 hours a day; and
- Duplication of effort in the virtualization space at Google across multiple divisions.

Taken together, these issues reduced the time and resources available to the team tasked with improving the offering. To tackle these problems, the team began migrating Google's top corporate workload to GCP in early 2016.

Planning

Planning for the migration consisted of several discrete stages. In aggregate, the planning phases described here took approximately three to four months and involved the participation of approximately 15 subject-matter-expert groups.

Vision. This phase articulated the core business and engineering case for replatforming our virtual desktops to GCP. The top reasons for pursuing virtual desktops on Cloud included:

- Large reduction in engineering toil;
- Improved user experience;
- Reduced total cost of platform ownership; and
- Desire to improve GCP.

Customer user journeys. The next step was to study the users and their needs. The primary users—the virtual desktop owners, fleet Site Reliability

IMAGE BY EVERYTHING POSSIBLE

85.3.66.1
85.3.66.1
85.3.66
85.3.66.1
85.3.66.1
85.3.66.1
85.3.66.1
85.3.66.1

Engineers (SREs), and the fleet security managers—were identified and surveyed to determine their typical workflows. Using this information, the migration team wrote implementation-agnostic user journeys. To perform effective gap analysis, and to reduce bias during the design phase, the team made a conscious effort to describe user journeys in a purely functional fashion.

Production milestone definition. Based on the survey responses and usage patterns collected, the team grouped customer user journeys (by both technology area and user type) and prioritized them into bands of features labeled alpha, beta, and general availability.

Workstream definition. In parallel to milestone definition, the team grouped requirements into seven streams of related work such as networking and provisioning. Each workstream was assigned a technical lead, a project lead, and a skeleton staff. Each team was virtual, recruited from across reporting lines as needed to address the work domain. The flexibility provided by this form of organization and associated matrix management turned out to be essential as the project evolved.

Engineering prototyping gap analysis, and design proposals. Once formed, each workstream examined the critical user journeys in their domains and researched the feasibility of implementing these stories on GCP. To do so, the team performed a gap analysis for each user journey by reading product GCP documentation and running "fail-fast" prototyping sprints. Throughout this process, possible implementations were collected and rated according to complexity, feasibility, and (most importantly) how easily a customer external to Google could implement this solution.

Whenever the migration team arrived at a "Google-only" solution, it filed a feature request to the GCP team requesting a solution that would work for customers outside of Google as well, especially if another enterprise customer would be interested in such functionality. In this way, the team sought to "act like a customer" in an effort to make the platform enterprise-ready. Where the GCP product teams could not deliver a feature in time for a release milestone, they implemented bridging solutions that favored solutions the public could use (for example, Forseti Security) above Google-only workarounds.

Workstream work breakdown and staffing. With design proposals in place and implementation directions decided, the team created detailed work plans for each workstream in a central project management tool. The work was organized by customer user journey, and tasks were broken down by objective, key results, and quarter. Drilling down to this level of detail provided enough information to estimate the staffing required for each workstream, to understand interdependencies between streams, and to fine-tune the organization as needed.

Technical Implementation Details

Once planning was complete, the team was ready to begin implementing the technical details of the migration. This section describes the three main buckets of work.

Background: Networking and BeyondCorp. Many of the networking challenges of running a desktop service on Google Compute Engine (GCE) were at least partially solved by the BeyondCorp program (https://cloud.google.com/beyondcorp/). In a BeyondCorp model, access controls are based on known information about a given device and user rather than on the location in a privileged network. When network trust no longer factors into access-control decisions, many services become readily accessible from outside of the corporate network—usually via a corporate laptop, but now also from appropriately managed and inventoried hosts on GCE.

Enterprises that leverage traditional virtual private networks (VPNs) for remote access to applications will have a different networking experience when moving desktops or other services. A typical strategy is to set up a Cloud VPN (https://cloud.google.com/compute/docs/vpn/overview) in a cloud project and peer with on-premises equipment to bridge the networks together.

Host authentication and authorization. Device authentication is usually performed using client certificates deployed on the host. When a user receives a physical machine (or even a virtual machine on privileged corporate networks), the user can initially log in and request a certificate because the corporate network retains the level of privilege needed to sync login policies. Extending this level of network privilege to public cloud IP ranges is undesirable for security reasons.

To bridge this gap, Google developed a (now-public) API to verify the identity of instances (https://cloud.google.com/compute/docs/instances/verifying-instance-identity). The API uses JWTs (JSON Web tokens; JSON is the JavaScript Object Notation data-interchange format) to prove that an instance belongs to a preauthorized Google Cloud project. When an instance first boots, one of the JWTs provided by that API can be exchanged for a client certificate used to prove device identity, unblocking nearly all of the normal communication paths (including syncing login policies for user authorization).

Once the client certificate is in place, Google applications can be accessed via the BeyondCorp/identity-aware proxy as if they were any other Internet-facing service. In order for cloud desktops to reach the proxies (and other Internet endpoints), the team set up network address translation (NAT) gateways (https://cloud.google.com/compute/docs/vpc/special-configurations) to forward traffic from the instances to targets outside of the cloud project. In combination, these approaches allow users to access internal resources and the public Internet seamlessly, without requiring each instance to have a publicly routed IP address.

Provisioning. The first step in designing a provisioning scheme was to map out everything necessary to deliver an end product that met users' needs. Compute Engine instances needed levels of trust, security, manageability, and performance for users to perform their jobs—developing, testing, building, and releasing code—as normal. Working from these requirements, the team used the following specific principles to guide the rest of the design.

Users should interact with a cloud desktop similarly to how they interact with hosts on the corporate network. Users should be able to use their normal authentication mechanisms. They should also be able to use the same tools to check machine statistics or report issues that they would use for physical desktops or Google's legacy virtual desktop platform.

Instances must be securely inventoried. As a first step in the provisioning process, host inventory is bootstrapped. As the hosts are used, further inventory data is collected, both for reliability monitoring and to inform access-control decisions.

Google's corporate network uses multiple inventory systems, cross-referenced to validate such access requests (for more context on BeyondCorp's inventory process, see https://research.google.com/pubs/pub43231.html and https://research.google.com/pubs/pub44860.html). Therefore, corporate systems need some metadata during the provisioning process to indicate a privileged virtual desktop creation request. This metadata is then cross-referenced with inventory data pulled from Google Cloud APIs in order to evaluate compliance with security policy and assert trust.

Instances must be securely managed. A host must be able to securely download the information needed to construct its authorization policies such that only permitted users have access. Hosts must also be able to update packages and configurations driven by the operating system installation and must be able to send logs to a central location so irregular behavior or installation-related issues can be detected.

Instances must be created to user specifications, with constraints. Instances should be provisioned with enough resources for users to do their work effectively but should also have caps to prevent users from gratuitously creating high-specification/high-cost devices. Users can choose where to build their instances (typically, in regions close to their physical location; disaster-recovery reasons might dictate a different location).

Because of the need to manage and inventory the devices in corporate systems, there were two options for creating new instances once desktop was migrated to cloud:

- Creating cloud desktop instances on behalf of the requester; or
- Allowing users to interface natively with cloud to create and manipulate their instances. In this scenario, it would be necessary to observe these changes and make corresponding updates in the corporate tools.

Enterprises that leverage traditional virtual private networks for remote access to applications will have a different networking experience when moving desktops or other services.

The team decided on the first approach and integrated existing virtual-machine management tools with GCE. As a result, they could enforce more complex business logic in line with the user's request.

The workflows built around the GCE provisioning process focus mostly on translating data provided by the requesting user, plus data known about the user (group membership, job role), into a request that can be passed to GCE. The instance can then be tracked through the creation process, as well as during any first-boot operating system updates and configuration, to make sure a usable machine is delivered to the user.

Operating system. Google uses an internally managed Linux distribution based on Debian, called gLinux, for corporate hosts. On its corporate network, gLinux is installed by loading a bootstrap environment on first boot. Large parts of the root file system are unpacked from a tarball, and the Debian installer then performs the actual installation.

On Cloud, this process starts with an image created by gLinux release tooling, which is uploaded to Cloud storage and imported into GCE (https://cloud.google.com/compute/docs/images/import-existing-image). Creating a disk from this image results in a fully runnable and bootable file system. You can boot directly from the imaged disk and only need to perform some small modifications on first boot before it's fully usable: the file system grows to fill the full span of the disk, the hostname updates, and a few other GCE-specific modifications are performed.

To avoid the burden of maintaining and testing separate behavior on different platforms, the team needed to minimize specific customizations to gLinux on Compute Engine. Fortunately, this effort required very few modifications, most of which focused on DNS (Domain Name System) name resolution. For example, corporate DNS zones, which many applications running on Google's corporate network require, are not available off-network. To address this need, the team introduced a DNS resolver that runs inside each instance to proxy requests for internal DNS zones back to the corporate servers over HTTP through a BeyondCorp proxy.

Alpha and Beta Rollouts

Before beginning the migration to GCP in earnest, the team conducted alpha and beta rollouts as initial sets of features were ready. The alpha release targeted roughly 100 users, while the beta release targeted roughly 1,000 users.

To evaluate the success of both releases, the following metrics were tracked and compared with Google's existing corporate fleet statistics:

- *Pager load.* There was a 95% drop in pager load once we migrated virtual desktops to Google Cloud, in large part because of the platform abstraction provided by GCP. While the team maintained a large fleet of physical servers, storage units, network equipment, and support software to run virtual desktops on the corporate network, GCE removed all these concerns.
- *Interrupts load.* Initially, the migration to cloud led to an increase in interrupts as a result of the novelty of the system (which also resulted in a corresponding increase in product bug reports). After this initial surge, interrupts load dropped to just 20% of the volume experienced when virtual desktops were hosted on the corporate network.
- *Login rates.* Seven- and 30-day login rates before and after the migration were comparable.

To inform the roadmap for future milestones, the team also collected data during alpha and beta releases via two main avenues:

- *User-reported feedback.* User feedback in the form of tickets, bug reports, emails, and word of mouth provided a list of items to fix. The team filed bugs to track each list item, prioritized by severity, number of users impacted, and aggregated customer preference. The last metric was made measurable by offering power users 100 "feature points" to apportion across features as they wished. These metrics could then be used to inform development priorities.
- *Surveys.* Surveys measured subjective impressions from the user base with the goal of improving marketing. For example, subjective feedback on the relative performance of virtual desktops on Cloud versus the corporate network was split (when, in fact, performance on Cloud was superior). In response, the team emphasized and more heavily promoted benchmarking results to customers, with the hope of prompting more objective valuations. Surveys also helped quantify fears about the transition, most notably around performance, user-data migration, and the ability to roll back to the corporate network-hosted offering when a given user workflow was not supported.

A host must be able to securely download the information needed to construct its authorization policies such that only permitted users have access.

Based on survey feedback, the team emphasized the following aspects to make the migration to Google Cloud attractive to users:

- *Improved VM specs.* Users were offered large increases in standard CPU, RAM, and disk specs.
- *One-click personal data migration.* The migration process was easy to begin with, and it automated the most time-consuming part of users' workflow: copying over personal data.
- *Easy rollback.* The migration process allowed users to roll back to their Corp-hosted instances, which were simply shut down before the migration to Cloud. Unused Corp-hosted instances were deleted after a grace period of 90 days.
- *Impact on the company.* Clearly articulating the cost reduction to Google helped reassure users they were doing the right thing for the company.

Migrating Users: Technical and Process Details

After collecting data and feedback from alpha and beta rollouts, the team was ready to proceed with the general migration to Cloud. This section details the main features of the migration process.

Trade-in. Once the provisioning system was ready for new users, approximately 20,000 virtual desktop users had to be moved to the new product. The team briefly considered a naive strategy of simply moving each user's disks into a new Cloud instance, but experience in managing the existing platform pointed in a slightly different direction.

Occasionally, the old platform experienced a fault that required significant work to return a user's instance to full functionality. This fault became more common as incremental, automatic modifications to an instance accrued. To spare unnecessary toil, the team's default first response in these scenarios was to ask users if they needed the data on the disk in question. Much of the time, the answer was no, as users didn't have any important data on their disks. Following this strong (but anecdotal) signal from corporate network-hosted virtual desktops, the team based the move to Cloud around the concept of user-involved *trade-ins*, as opposed to a traditional behind-the-scenes *migration*.

To carry out such a trade-in program, the team crafted two workflow pipelines: one to handle cases where users explicitly indicated that they did not need their data moved (or could move it themselves); and one for outliers who needed all of their data moved. The former (and by far, most common) case required simply performing two straightforward tasks: powering off the original instance and creating a new instance with a similar name. This approach required minimal engineering effort; used minimal compute, bandwidth, and storage resources; and provided a strong signal for how much benefit users gathered from traditional stateful disks.

Exceptional cases. Google employs a great many engineers; if you give an engineer a Linux machine, there's a good chance that the machine will be customized within an inch of its life. For these cases, the team wanted to provide a path to use the new platform that did not force all users to migrate their own data. However, the team was wary of setting a "magical migration" expectation for edge cases that could not possibly be fulfilled.

For users who requested data migration, the best option was to move their home directories in-place to the new cloud instances and provide an on-disk backu p of their entire operating system. While this strategy duplicated a significant amount of operating system data already present on the gLinux system, it meant the team could proceed without worrying that important files were not transferred, only to be noticed months later.

Moving bits. The actual uploading proceeded in a straightforward manner. For each user request, a job execution system crafted a signed Google Cloud Storage URL entitling the bearer to perform an HTTP PUT request to a bucket for a period of time. To ensure that long-running upload jobs were not interrupted as the workflow system deployed, an Upstart script on the old virtualization platform processed the upload. Upon upload completion, a cloud worker instance fired up to create a new disk image by merging the user's data onto a copy of the golden master image.

Push vs. pull. The beta phase revealed a high demand for virtual desktops hosted on Cloud, so the team wanted to make sure the general launch adequately anticipated demand. To avoid overwhelming the network links on the old virtualization platform and the team's capacity for toil in handling a surge of requests, users were not allowed to request trade-ins themselves. Instead, trade-ins were first offered to a population of users who were most likely to need them: those with low disk usage who likely wouldn't need to move their entire disk, and users whose instances were hosted on old hardware. In this way, the team could both balance capacity limits and precisely target user populations whose machine locations would buy the most reduction in toil. Once users actually started using the platform, high demand for cloud desktops meant that users wanted to opt in earlier. Attempts to create new cloud desktop instances without an explicit invitation to do so were a signal to send those users a trade-in invitation.

Early Known Issues and Limitations

While initial reception of the new service was positive, a few barriers caused mild inconvenience to users and some completely broken use cases.

Most of these pain points affected the provisioning process and were largely caused by misunderstanding the various SLOs (service-level objectives) and delays within Google's inventory and trust pipeline. The team required an independent signal of user intent in order to grant trusted access to a new desktop instance, and once a user provided this signal by "enrolling" the instance (post-creation), the user still needed to wait several hours before the instance became fully trusted by all systems. Provisioning and testing an instance therefore took three user interactions over a period of three to five hours. Once the team became aware of this issue, they made changes to the trust pipeline and folded the enrollment step into the initial user request, thereby eliminating the hours of waiting. If the team had considered the timing of provisioning and other operations as building requirements, they could have made these improvements earlier.

Broken use cases ended up flushing out many bad assumptions that various applications and groups of users made about network access. For example, instead of using a library to check if a user is on the corporate network or the production network, some suboptimal implementations instead depended on hostname or IP space. These issues were typically addressed by updating the code of individual applications to remove the bad assumptions.

There were also issues with some workflows that technically violated security policy but had been granted exceptions. Cloud desktop enforces these policies by default; it encourages users to fix their workflows rather than carry forward bad practices. For example, on Google's corporate network, users can get an exception to connect directly to some application databases. This is practical because the database server and the user are on the same network. These sorts of use cases should be steered toward BeyondCorp gateways.

Technical and Non-Technical Lessons Learned

As with any complicated launch, the development team learned a number of lessons along the way, both technical and nontechnical.

Push, don't pull. Being able to control the flow of traffic to your system makes operating it infinitely simpler. Find a bug? Stop sending invites. Everything humming along? Turn up the volume. Even if integrating this functionality is difficult, it's worth implementing from square one if possible.

Be explicit about trade-offs and costs with your users. If you offer two options, and one seems like less work, everyone will choose that easier option. To offset this impulse, if one option is much more costly, expose that cost at the user decision point. For example, moving disks into the cloud is convenient for users but much more time-consuming (and costly) than the alternative. The team exposed the cost of moving disks as a 24-hour duration, which was much less convenient than the one-hour duration for a simple exchange of a corporate network-hosted instance for a Cloud-hosted instance. Simply exposing this information when users had to choose between the two options saved an estimated 1.8 petabytes of data moves.

Never waste an opportunity to gather data. Before the migration, the team didn't know what proportion of users depended heavily on the contents of their local disks. It turns out that only about 50% of users cared enough about preserving their disks to wait 24 hours for the move to complete. That's a valuable data point for future service expansions or migrations.

Don't be tempted to make a special case out of a "one-time" migration." Your future self will be thankful if you take the opportunity to homogenize when making lasting changes. Previous generations of the corporate network-hosted virtual desktop system had a slightly different on-disk layout than the current models used for testing. Not only was this an unpleasant surprise in production, but it was also almost impossible to test since no existing tools would create the old disk type. Fortunately, during the design phase the team had resisted the urge to "simplify" the data-copying phase by putting user data on a second GCE disk—doing so would have made these instances special snowflakes for the lifetime of the Cloud-hosted platform.

Keep the organization flexible. Organizing the team into virtual workstreams has multiple benefits. This strategy allowed the team to quickly gather expertise across reporting chains, expand and contract teams throughout the project, reduce communication overhead between teams, assign singular deliverable objectives to work groups, and reduce territoriality across teams.

This is an opportunity to "get it right." The migration to Google Cloud allowed the team to reconsider certain implementations that had ossified over time within the team and organization.

Applying This Experience Elsewhere

Since a cloud desktop is composed of a GCE instance running a custom image (production of which is fairly cheap and well documented; https://cloud.google.com/compute/docs/images#custom_images), the infrastructure scales extraordinarily well. Very little changed when piloting with a dozen instances versus running with thousands, and what Google has implemented here should be directly applicable to other, smaller companies without requiring much specialization to the plan detailed in this article.

Future Plans

While the migration of virtual desktops to Cloud wasn't painless, it has been a solid success and a foundation for further work. Looking to the future, the Google Corporate Cloud Migrations team is engaged in two primary streams of work: improving the virtual desktop experience and enabling Google corporate server workloads to run on Cloud.

In the desktop space, the team plans to improve the service management experience by developing various tools that supplement the Google Cloud platform to help manage the fleet of cloud desktops. These add-ons include a disk-inspection tool and a fleet-management command-line tool that integrates and orchestrates actions between Cloud and other corporate systems.

There are several possibilities for improving fleet cost effectiveness. On the simple end of the spectrum, cloud desktop could automatically request that owners of idle machines delete instances they don't actually need.

Finally, the end-user experience could be improved by implementing a self-serve VM cold migration between datacenters, allowing traveling users to relocate their instances to a nearby datacenter to reduce latency to their VM. Note that these plans are scoped to cloud desktop as part of the customer/application-specific logic, as opposed to features Google as a company is planning for Compute Engine in general.

As for server workloads, the team is building on lessons learned from cloud desktop to provide a migration path. The main technical challenges in this space include:

- Cataloging and characterizing the corporate fleet;
- Creating scalable and auditable service and VM lifecycle management frameworks;
- Maintaining multiple flavors of managed operating systems;
- Extending BeyondCorp semantics to protocols that are hard to proxy;
- Tackling a new set of security and compliance requirements;
- Creating performant-shared storage solutions for services requiring databases;
- Creating migration tools to automate toilsome operations; and
- Implementing a number of service-specific requirements.

Migrating server workloads also has the added organizational complexity of a heterogeneous group of service owners, each with varying priorities and requirements from the departments and business functions they support. C

Related articles on queue.acm.org

Titus: Introducing Containers to the Netflix Cloud
Andrew Leung, Andrew Spyker, and Tim Bozarth
https://queue.acm.org/detail.cfm?id=3158370

Reliable Cron across the Planet
Štepán Davidovič, Kavita Guliani
https://queue.acm.org/detail.cfm?id=2745840

Virtualization: Blessing or Curse?
Evangelos Kotsovinos
https://queue.acm.org/detail.cfm?id=1889916

Matt Fata is a Site Reliability Manager at Google, where he works on corporate virtualization solutions. He has previously worked as a network engineer and as an IT support desk manager.

Philippe-Joseph Arida is a Technical Program Manager at Google, where he works on making GCP the best platform for enterprise workloads. He previously worked as a PM at Microsoft on desktop, server, and search products.

Patrick Hahn is a Site Reliability Engineer at Google and the Technical Lead of the cloud desktop project. He has previously worked as a sysadmin in the Web development, managed IT, and quantitative finance industries.

Betsy Beyer is a technical writer for Google Site Reliability Engineering in NYC, and the editor of *Site Reliability Engineering: How Google Runs Production Systems* and the *Site Reliability Workbook*.

DOI:10.1145/3233243

Article development led by acmqueue
queue.acm.org

Three critical design points: Joint learning, weak supervision, and new representations.

BY ALEX RATNER AND CHRIS RÉ

Research for Practice:

Knowledge Base Construction in the Machine-Learning Era

THIS INSTALLMENT OF Research for Practice features a curated selection from **Alex Ratner** and **Chris Ré**, who provide an overview of recent developments in Knowledge Base Construction (KBC). While knowledge bases have a long history dating to the expert systems of the 1970s, recent advances in machine learning have led to a knowledge base renaissance, with knowledge bases now powering major product functionality including Google Assistant, Amazon Alexa, Apple Siri, and Wolfram Alpha. Ratner and Ré's selections highlight key considerations in the modern KBC process, from interfaces that extract knowledge from domain experts to algorithms and representations that transfer knowledge across tasks. Please enjoy!

—*Peter Bailis*

Peter Bailis is an assistant professor of computer science at Stanford University. His research in the Future Data Systems group (futuredata.stanford.edu) focuses on the design and implementation of next-generation data-intensive systems.

More information is accessible today than at any other time in human history. From a software perspective, however, the vast majority of this data is unusable, as it is locked away in *unstructured* formats such as text, PDFs, Web pages, images, and other hard-to-parse formats. The goal of KBC (knowledge base construction) is to extract structured information automatically from this "dark data," so that it can be used in downstream applications for search, question-answering, link prediction, visualization, modeling and much more. Today, knowledge bases (KBs) are the central components of systems that help fight human trafficking,[19] accelerate biomedical discovery,[9] and, increasingly, power web-search and question-answering technologies.[4]

KBC is extremely challenging, however, as it involves dealing with highly complex input data and multiple connected subtasks such as parsing, extracting, cleaning, linking, and integration. Traditionally, even with machine learning, each of these subtasks would require arduous *feature engineering* (that is, manually crafting attributes of the input data to feed into the system). For this reason, KBC has traditionally been a months- or years-long process that was approached only by academic groups (for example, YAGO,[8] DBPedia,[7] KnowItNow,[2] DeepDive,[18] among others) or large, well-funded teams in industry and government (for example, Google's Knowledge Vault, IBM Watson, and Amazon's Product Graphs).

Today, however, there is a renewed sense of democratized progress in the area of KBC, thanks to powerful but easy-to-use deep-learning models that largely obviate the burdensome task of feature engineering. Instead, modern deep-learning models operate directly over raw input data such as text or images and get state-of-the-art performance on KBC sub-tasks such as parsing, tagging, classifying, and linking. Moreover, standard commodity architectures are often suitable for a wide range of domains and tasks such as the "hegemony"[11] of the bi-LSTM (bidirectional long short-term memory) for text, or the CNN (convolutional neural network) for images. Open source implementations can often be downloaded and run in several lines of code.

For these emerging deep-learning-based approaches to make KBC faster and easier, though, certain critical design decisions need to be addressed—such as how to piece them together, how to collect training data for them efficiently, and how to represent their input and output data. This article highlights three papers that focus on these critical design points: *joint-learning* approaches for pooling information and coordinating among subcomponents; more efficient methods of *weakly supervising* the machine-learning components of the system; and, new ways of representing both inputs and outputs of the KB.

Joint Learning: Sharing Information and Avoiding Cascaded Errors

T.M. Mitchell et al.
Never-ending learning. In *Proceedings of the Conference on Artificial Intelligence*, 2015, 2302–2310.

KBC is particularly challenging because of the large number of related subtasks involved, each of which may use one or more ML (machine-learning) models. Performing these tasks in disconnected pipelines is suboptimal in at least two ways: it can lead to cascading errors (for example, an initial parsing error may throw off a downstream tagging or linking task); and it misses the opportunity to pool information and training signals among related tasks (for example, subcomponents that extract similar types of relations can probably use similar representations of the input data). The high-level idea of what are often termed *joint inference* and *multitask learning*—which we collectively refer to as *joint learning*—is to learn multiple related models jointly, connecting them by logical relations of their output values and/or shared representations of their input values.

Never-Ending Language Learner (NELL) is a classic example of the impact of joint learning on KBC at an impressive scale. NELL is a system that has been extracting various facts about the world (for example, `ServedWith(Tea, Biscuits)`) from the Internet since 2010, amounting to a KB containing (in 2015) more than 80 million entries. The problem setting approached by NELL consists of more than 2,500 distinct learning tasks, including categorizing noun phrases into specific categories, linking similar entities, and extracting relations between entities. Rather than learning all these tasks separately, NELL's formulation includes known (or learned) *coupling constraints* between the different tasks, which Mitchell et al. cite as critical to training NELL. These include logical relations such as subset/superset (for example, `IsSandwhich(Hamburger)` ⇒ `IsFood(Hamburger)`) and mutual-exclusion constraints, which connect the many disparate tasks during inference and learning.

In other systems, the importance of connecting or coupling multiple tasks is echoed in slightly different contexts or formulations: for example, as a way to avoid cascading errors between different pipeline steps such as extraction and integration (for example, DeepDive[18]), or implemented by sharing weights or learned representations of the input data between tasks as in multitask learning.[3,17] Either way, the decision of how to couple different subtasks is a critical one in any KBC system design.

Weak Supervision: Programming ML with Training Data

A.J. Ratner, S.H. Bach, H. Ehrenberg, J. Fries, J., S. Wu, and C. Ré
Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the Very Large Database (VLDB) Endowment 11*, 3 (2017), 269–282.

In almost all KBC systems today, many or all of the critical tasks are performed by increasingly complex machine-learning models, such as deep-learning ones. While these models indeed obviate much of the feature-engineering burden that was a traditional bottleneck in the KBC development process, they also require large volumes of labeled *training data* from which to learn. Having humans label this training data by hand is an expensive task that can take months or years, and the resulting labeled data set is frustratingly static: if the schema of a KB changes, as it frequently does in real production settings, the training set must be thrown out and relabeled. For these reasons, many KBC systems today use some form of *weak supervision*:[15] noisier, higher-level supervision provided more efficiently by a domain expert.[6,10] For example, a popular heuristic technique is distant supervision, where the entries of an existing knowledge base are heuristically aligned with new input data to label it as training data.[1,13,16]

Snorkel provides an end-to-end framework for weakly supervising machine-learning models by having domain experts write LFs (labeling functions), which are simply black-box functions that programmatically label training data, rather than labeling any training data by hand. These LFs subsume a wide range of weak supervision techniques and effectively give non-machine-learning experts a simple way to "program" ML models. Moreover, Snorkel automatically learns the accuracies of the LFs and reweights their outputs using statistical modeling techniques, effectively denoising the training data, which can then be used to supervise the KBC system. In this paper, the authors demonstrate that Snorkel improves over prior weak supervision approaches by enabling the easy use of many noisy sources, and comes within several percentage points of performance using massive hand-labeled training sets, showing the efficacy of weak supervision for making high-performance KBC systems faster and easier to develop.

Embeddings: Representation and Incorporation of Distributed Knowledge
S., Riedel, L. Yao, A. McCallum and B.M. Marlin
Relation extraction with matrix factorization and universal schemas. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies*: 2013, 74–84.

Finally, a critical decision in KBC is how to represent data: both the input unstructured data and the resulting output constituting the knowledge base. In both KBC and more general ML settings, the use of dense vector embeddings to represent input data, especially text, has become an omnipresent tool.[12] For example, word embeddings, learned by applying PCA (principal component analysis) or some approximate variant to large unlabeled corpora, can inherently represent meaningful semantics of text data, such as synonymy, and serve as a powerful but simple way to incorporate statistical knowledge from large corpora. Increasingly sophisticated types of embeddings, such as hyperbolic,[14] multimodal, and graph[5] embeddings, can provide powerful boosts to end-system performance in an expanded range of settings.

In their paper, Riedel et al. provide an interesting perspective by showing how embeddings can also be used to represent the knowledge base itself. In traditional KBC, an output schema (that is, which types of relations are to be extracted) is selected first and fixed, which is necessarily a manual process. Instead, Riedel et al. propose using dense embeddings to represent the KB itself and learning these from the union of all available or potential target schemas.

Moreover, they argue that such an approach unifies the traditionally separate tasks of extraction and integration. Generally, *extraction* is the process of going from input data to an entry in the KB—for example, mapping a text string `X` likes `Y` to a `KB` relation `Likes(X,Y)`—while *integration* is the task of merging or linking related entities and relations. In their approach, however, both input text and `KB` entries are represented in the same vector space, so these operations become essentially equivalent. These embeddings can then be learned jointly and queried for a variety of prediction tasks.

KBC Becoming More Accessible

This article has reviewed approaches to three critical design points of building a modern KBC system and how they have the potential to accelerate the KBC process: coupling multiple component models to learn them jointly; using weak supervision to supervise these models more efficiently and flexibly; and choosing a dense vector representation for the data. While ML-based KBC systems are still large and complex, one practical benefit of today's interest and investment in ML is the plethora of state-of-the-art models for various KBC subtasks available in the open source, and well-engineered frameworks such as PyTorch and TensorFlow with which to run them. Together with techniques and systems for putting the pieces all together like those reviewed, high-performance KBC is becoming more accessible than ever.

References

1. Bunescu, R.C., Mooney, R.J. Learning to extract relations from the Web using minimal supervision. In *Proceedings of the 45th Annual Meeting Assoc. Computational Linguistics*, 2007, 576–583.
2. Cafarella, M.J., Downey, D., Soderland, S., Etzioni, O. KnowItNow: Fast, scalable information extraction from the Web. In *Proceedings of Conf. on Human Language Tech. Empirical Methods in Natural Language Processing*, 2005, 563–570.
3. Caruana, R. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the 10th Intern. Conf. Machine Learning*, 1993, 41-48.
4. Dong, X. et al. Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining*, 2014, 601–610.
5. Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining*, 2016, 855–864.
6. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Assoc. Computational Linguistics–Human Language Technologies*, 1, 2011, 541–550.
7. Lehmann, J. et al. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web 6*, 2 (2014), 167–195.
8. Mahdisoltani, F., Biega, J. and Suchanek, F.M. YAGO3: A knowledge base from multilingual wikipedias. In *Proceedings of the 7th Biennial Conf. Innovative Data Systems Research*, 2013.
9. Mallory, E.K., Zhang, C., Ré, C. and Altman, R.B. Large-scale extraction of gene interactions from full-text literature using DeepDive. *Bioinformatics 32*, 1 (2015), 106–113.
10. Mann, G.S. and McCallum, A. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *J. Machine Learning Research 11* (Feb 2010), 955–984.
11. Manning, C. Representations for language: From word embeddings to sentence meanings. Presented at Simons Institute for the Theory of Computing, UC Berkeley; https://nlp.stanford.edu/manning/talks/Simons-Institute-Manning-2017.pdf.
12. Mikolov, T., Chen, K., Corrado, G. and Dean, J. Efficient estimation of word representations in vector space, 2013; arXiv preprint arXiv:1301.3781.
13. Mintz, M., Bills, S., Snow, R. and Jurafsky, D. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conf. 47th Annual Meeting of the Assoc. Computational Linguistics* and *the 4th Conf. Asian Federation of Natural Language Processing*, 2009, 1003–1011.
14. Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems 30* (2017), 6341–6350.
15. Ratner, A., Bach, S., Varma, P. and Ré, C. Weak supervision: the new programming paradigm for machine learning. Hazy Research; https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html.
16. Ren, X., He, W., Qu, M., Voss, C. R., Ji, H., Han, J. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining*, (2016), 1825–1834.
17. Ruder, S. An overview of multi-task learning in deep neural networks, 2017; arXiv preprint arXiv: 1706.05098.
18. Zhang, C., Ré, C., Cafarella, M., De Sa, C., Ratner, A., Shin, J., Wang, F., Wu, S. DeepDive: Declarative knowledge base construction. *Commun. ACM 60*, 5 (May 2017), 93–102.
19. Zhang, C., Shin, J., Ré, C., Cafarella, M. and Niu, F. Extracting databases from dark data with DeepDive. In *Proceedings of the Intern. Conf. Management of Data*, 2016, 847–859.

Alex Ratner is a Ph.D. candidate in computer science at Stanford University, advised by Chris Ré, where his research focuses on weak supervision—using higher-level, noisier input from domain experts to train complex state-of-the-art models where limited hand-labeled training data is available. He leads the development of the Snorkel framework for weakly supervised ML, which has been applied to KBC problems in domains such as genomics, clinical diagnostics, and political science. He is supported by a Stanford Bio-X SIGF fellowship.

Christopher Ré is an associate professor of computer science at Stanford University. His work focuses on enabling users and developers to build applications that more deeply understand and exploit data. Work from his group has been incorporated into major scientific and humanitarian efforts, including the IceCube neutrino detector, PaleoDeepDive, and MEMEX in the fight against human trafficking, and into commercial products from major Web and enterprise companies.

DOI:10.1145/3267118

Article development led by acmqueue
queue.acm.org

Dependency management is a crucial part of system and software design.

BY SILVIA ESPARRACHIARI GHIROTTI, TANYA REILLY, AND ASHLEIGH RENTZ

Tracking and Controlling Microservice Dependencies

IN SEARCH OF a cappuccino, cheese bread, and a place to check her email, Silvia walked into a coffee shop. Upon connecting to the Wi-Fi hotspot, a captive portal prompted her to log in and offered a few third-party authentication options. When she clicked on one of the access token providers, her browser showed a "No Internet Connection" error. Since she didn't have access to the network, she could not get an OAuth token—and she couldn't access the network without one.

This anecdote illustrates a critical detail of system design that can easily go unnoticed until an outage takes place: cyclic dependencies.

Dependency cycles are familiar to you if you have ever locked your keys inside your house or car. You cannot open the lock without the key, but you can't get the key without opening the lock. Some cycles are obvious, but more complex dependency cycles can be challenging to find before they lead to outages. Strategies for tracking and controlling dependencies are necessary for maintaining reliable systems.

Reasons to Manage Dependencies

A lockout, as in the story of the cyclic coffee shop, is just one way that dependency management has critical implications for reliability. You cannot reason about the behavior of any system, or guarantee its performance characteristics, without knowing what other systems it depends on. Without knowing how services are interlinked, you cannot understand the effects of extra latency in one part of the system, or how outages will propagate. How else does dependency management affect reliability?

SLO. No service can be more reliable than its critical dependencies.[8] If dependencies are not managed, a service with a strict service-level objective (SLO)[1] might depend on a back end that is considered best-effort. This might go unnoticed if the back end has coincidentally high availability or low latency. When that back end starts performing exactly to its SLO, however, it will degrade the availability of services that rely on it.

High-fidelity testing. Distributed systems should be tested in environments that replicate the production environment as closely as possible.[7] If noncritical dependencies are omitted in the test environment, the tests cannot identify problems that arise from their interaction with the system. This can cause regressions when the code runs in production.

Data integrity. Poorly configured production servers may accidentally depend on their development or quality assurance (QA) environments. The reverse may also be true: A poorly configured QA server may accidentally leak fake data into the production environment. Experiments might inadvertently send requests to production servers

IMAGE BY VERSHININ89

and degrade production data. Dependency management can expose these problems before they become outages.

Disaster recovery/isolated bootstrap. After a disaster, it may be necessary to start up a company's entire infrastructure without having anything already running. Cyclic dependencies can make this impossible: a front-end service may depend on a back end, but the back-end service could have been modified over time to depend on the front end. As systems grow more complex over time, the risk of this happening increases. Isolated bootstrap environments can also provide a robust QA environment.

Security. In networks with a perimeter-security model, access to one system may imply unfettered access to others.[9] If an attacker compromises one system, the other systems that depend on it may also be at risk. Understanding how systems are interconnected is crucial for detecting and limiting the scope of damage. You may also think about dependencies when deploying denial of service (DoS) protection: One system that is resilient to extra load may send requests downstream to others that are less prepared.

Dependency Cycles

Dependency cycles are most dangerous when they involve the mechanisms used to access and modify a service. The operator knows what steps to take to repair the broken service, but it is impossible to take those steps *without* the service. These control cycles commonly arise in accessing remote systems. An error that disables sshd or networking on a remote server may prevent connecting to it and repairing it. This can be seen on a wider scale when the broken device is responsible for routing packets: The whole network might be offline as a result of the error, but the network outage makes it impossible to connect to the device and repair it. The network device depends on the very network it provides.

Dependency cycles can also disrupt recovery from two simultaneous outages. As in the isolated bootstrap scenario, two systems that have evolved to depend upon each other cannot be restarted while neither is available. A job-scheduling system may depend on writing to a data-storage system, but that data-storage system may depend on the job-scheduling system to assign resources to it.

Cycles may even affect human processes, such as oncall and debugging. In one example, a source-control system outage left both the source-code repository and documentation server unavailable. The only way to get to the documentation or source code of the source-control system was to recover the same system. Without this key information about the system's internals, the oncall engineer's response was significantly obstructed.

Microservices and External Services

In the era of monolithic software development, dependency management was relatively clear-cut. While a monolithic binary may perform many functions, it generally provides a single failure domain containing all of the binary's functionality. Keeping track of a small number of large binaries and storage systems is not difficult, so an owner of a monolithic architecture can easily draw a dependency diagram, perhaps like that in Figure 1.

The software industry's move toward the microservices model makes dependency management much more difficult. As Leslie Lamport said in 1987, "A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."[5] Large binaries are now frequently broken into many smaller services, each one serving a single purpose and capable of failing independently. A retail application might have one service for rendering the storefront, another for thumbnails, and more for currency conversion, checkout, address normalization, and surveys. The dependencies between them cross failure domains.

Figure 1. Sample dependency diagram.

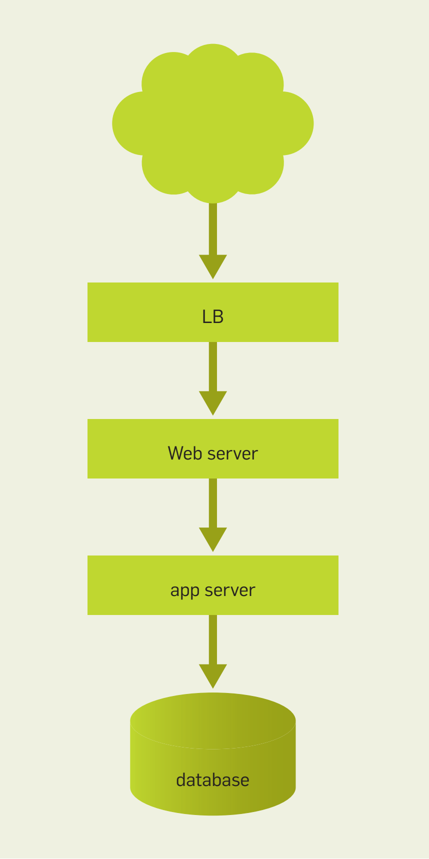
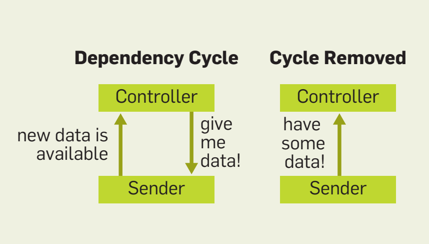


Figure 2. Cycle removal.



In her 2017 Velocity NY Conference talk, Sarah Wells of the *Financial Times* explained how her development teams manage more than 150 microservices—and that is for just one part of the *Financial Times*'s technical estate. Squarespace is in the process of breaking down its monolith[4] and already has more than 30 microservices. Larger companies such as Google, Netflix, and Twitter often have thousands of microservices, pushing the problem of dependency management beyond human capabilities.

Microservices offer many advantages. They allow independent component releases, smoother rollbacks, and polyglot development, as well as allow teams to specialize in one area of the codebase. However, they are not easy to keep track of. In a company with more than 100 microservices, it is unlikely that employees could draw a diagram and get it right, or guarantee they are making dependency decisions that will not result in a cycle.

Both monolithic services and microservices can experience bootstrapping issues caused by hidden dependencies. They rely on access to decryption keys, network, and power. They may also depend on external systems such as DNS (Domain Name System). If individual endpoints of a monolith are reached via DNS, the process of keeping those DNS records up to date may create a cycle.

The adoption of SaaS (software as a service) creates new dependencies whose implementation details are hidden. These dependencies are subject to the latency, SLO, testing, and security concerns mentioned previously. Failure to track external dependencies may also introduce bootstrapping risks. As SaaS becomes more popular and as more companies outsource infrastructure and functionality, cyclic dependencies may start to cross companies. For example, if two storage companies were to use each other's systems to store boot images, a disaster that affected both companies would make recovery difficult or impossible.

Directed Acyclic Graphs

At its essence, a service dependency is the need for a piece of data that is remote to the service. It could be a configuration file stored in a file system, or a row for user data in a database, or a computation performed by the back end. The way this remote data is accessed by the service may vary. For the sake of simplicity, let's assume all remote data or computation is provided by a serving back end via remote procedure calls (RPCs).

As just described, dependency cycles among systems can make it virtually impossible to recover after an outage. The outage of a critical dependency propagates to its dependents, so the natural place to begin restoring the flow of data is the top of the dependency chain. With a dependency cycle, however, there is no clear place to begin recovery efforts since every system is dependent on another in the chain.

One way to identify cycles is to build a dependency graph representing all services in the system and all RPCs exchanged among them. Begin building the graph by putting each service on a node of the graph and drawing directed edges to represent the outgoing RPCs. Once all services are placed in the graph, the existing dependency cycles can be identified using common algorithms such as finding a topological sorting via a depth-first search. If no cycles are found, that means the services' dependencies can be represented by a directed acyclic graph (DAG).

What happens when a cycle is found? Sometimes, it's possible to remove a cycle by inverting the dependency, as shown in Figure 2. One example is a notification system where the senders notify the controllers about new data, and the controller then pulls data from the senders. The cycle here can be easily removed by allowing the senders only to push data into the controller. Cycle removal could also be accomplished by splitting the functionality across two nodes—for example, by moving the new data notification to a third system.

Some dependencies are intrinsically cyclic and may not be removed. Replicated services may periodically query their replicas in order to reinforce data synchronization and integrity.[3] Since all replicas represent a single service, this would be represented as a self-dependency cycle in the graph. It's usually okay to allow self-dependencies as long as they do not prevent the isolated bootstrapping of the system and can properly recover from a global outage.

Another intrinsically cyclic dependency occurs in data-processing pipelines implemented as a workers-controller system.[2] Workers keep the controller informed about their status, and the controller assigns tasks to the workers when they become idle. This cyclic dependency between workers and controllers may not be removed without completely changing the processing model. What can be done in this case is to group workers and controllers into a supernode representing a single service. By repeating this edge contraction for all strongly connected components of the graph, taking into account their purpose and practical viability, you may achieve a DAG representation of the original graph.

Dependency cycles among systems can make it virtually impossible to recover after an outage.

Tracking vs. Controlling

In some environments, you can derive great benefit from just understanding the existing dependency graph. In others, determining the existing state is not sufficient; mechanisms are needed for preventing new undesirable dependencies. The two approaches examined here—dependency tracking and dependency control—have different characteristics:

- *Tracking dependencies is a passive approach.* You use logging and monitoring to record which services contact each other, then look back at that data in the future. You can understand the dependencies by creating data structures that can be queried efficiently or by representing the relationships visually.
- *Controlling dependencies is an active approach.* There are several points during design and implementation where you can identify and avoid an undesirable dependency. Additionally, you can prevent connections from being made while the code is running in production. If you wait until the dependency has already been used and monitored, it will be too late to prevent the issues it may cause.

These approaches overlap (for example, data collected during dependency control can certainly be used for tracking), but let's look at them separately.

Dependency tracking. Initially, dependency tracking often takes the form of information stored in engineers' heads and visualized in whiteboard drawings. This is sufficient for smaller environments, but as the system becomes more complex, the map of services becomes too complicated for any one person to memorize. Engineers may be surprised by an outage caused by an unexpected dependency, or they may not be able to reason about how to move a service and its associated back ends from one data center to another. At this stage, organizations begin to consider programmatically generated views of the system.

Different environments may use different ways of collecting information about how services are interconnected. In some, a firewall or network device might record logs of which services are contacting each other, and these logs can be mined for dependency data. Alternatively, a set of services built on a common framework might export standard monitoring metrics about every connection; or distributed tracing might be used to expose the paths a request takes through the system, highlighting the connections.

You can aggregate whatever sources of information are available to you and create a dependency graph, processing the data into a common structure and optimizing it for running queries over it. From there, you can use algorithms on the graph to check whether it is a DAG, visualize it using software such as Graphviz and Vizceral, or expose information for each service, perhaps using a standard dashboard with a page for each service.

Continually monitoring traffic between systems and immediately integrating it into the graph may see new dependencies shortly after they reach production. Even so, the information is available only after the new dependency has been created and is already in use. This is sufficient for dependency *tracking*, where you want to describe the interconnections of an existing system and become aware of new ones. Preventing the dependency, however, requires dependency *control*.

Dependency control. Just like dependency tracking, dependency control typically starts as a manual process using information stored in engineers' heads. Developers might include a list of proposed back ends in all design documentation and depend on their colleagues' knowledge of the existing systems to flag dangers. Again, this may be enough for a smaller environment. As services are born, grow, change, and are deprecated, the data can quickly become stale or unwieldy. Dependency control is most effective if enforced programmatically, and there are several points to consider in adding it.

When working on dependency management at Google, we found it best to think about controlling dependencies from the client side of a client-server connection (that is, the service that is about to depend on another service). By owning the code that initiates the connections, the owner of the client has the most control and visibility over which dependencies exist and can therefore detect potential problems earlier. The client is also most affected by ill-considered dependencies.

Although the owner of a server may want to control who its clients are for reasons such as capacity planning or security, bad dependencies are much more likely to affect the client's SLO. Because the client requires some functionality or data from the server for its own functionality or performance, it needs to be prepared for server-side outages. The server, on the other hand, is unlikely to notice an outage of one of its clients.

One approach to dependency control is to analyze the client's code and restrict dependencies at build time. The behavior of the binary, however, will be influenced by the configuration and environment it receives. Identical binaries might have very different dependencies in different situations, and the existence of code inside a binary is not a reasonable predictor for whether that binary has a dependency on another service. If a standard mechanism is used for specifying back ends or connection types—for example, if all back ends are provided in configuration and not in code—this might be an area worth exploring.

One approach to dependency control is to analyze the client's code and restrict dependencies at build time.

Restrictions are most effective if applied at runtime. By intercepting and potentially blocking connections as they are being made, you can be certain that you are inspecting the actual behavior of the running system, rather than speculating based on code or configuration. To avoid wasted engineering effort, restrictions on the back ends that services may contact should be implemented as early in the development life cycle as possible. Changing a system's architecture after it is already live and in use is far more expensive.[6] By applying the same set of restrictions at all stages of software development—during development, testing, canarying, and running live—any unwelcome dependency can be identified early.

There are several options for runtime enforcement. Just as with dependency tracking, existing infrastructure could be repurposed for dependency control. If all interservice connections pass through a firewall, network device, load balancer, or service mesh, those infrastructure services could be instrumented to maintain a list of acceptable dependencies and drop or deny any requests that do not match the list. Silently dropping requests at a point between the client and server may complicate debugging, though. A request that is dropped for being an unapproved dependency may be indistinguishable from a failure of the server or the intermediate device: the connections may seem to just disappear.

Another option is to use a dedicated external dependency-control service that the client can query before allowing each new back-end connection. This kind of external system has the disadvantage of adding latency since it requires extra requests to allow or deny each back end. And, of course, the dependency-control service itself becomes a dependency of the service.

At Google, we had the most success adding restrictions into the code at the point where the connection is made.

Since Google had a homogenous environment with a standard RPC mechanism used for all connections, we were able to modify the RPC code to match each new back end against a "dependency control policy"—an extra configuration option provided to every binary.

Figure 3. Pseudocode for policy authorization.

```
func isAllowedByPolicy(rpc, acl):
  foreach expectedRPC in acl:
    if rpc == expectedRPC:
      # If the RPC is listed in the ACL,
      # it should be allowed.
      return true

  # If the RPC didn't match any item on the ACL,
  # it should be rejected.
  return false
```

Authorizing RPCs

The dependency-control policy consists of an access control list (ACL) of the RPC names a service is expected to initiate. For performance reasons, the policy is serialized and loaded by the service during startup. If the policy is invalid (because of syntax errors or data corruption), it's not used and the dependency control is not activated. If the policy is correct, it becomes active, and all outgoing RPCs are matched against it. If an RPC is fired but is not present in the policy, it will be flagged as rejected. Rejected RPCs are reported via monitoring so that service owners can audit them and decide on the correct course of action: remove the RPC from the binary if it's not a desired dependency or add it to the ACL if it's indeed a necessary new dependency.

The pseudocode depicted in Figure 3 shows how the authorization of RPCs could be implemented.

To prevent production outages, service owners are allowed to choose whether to enforce the policy and drop rejected RPCs or to soft-apply the policy and allow rejected RPCs to go through. Most service owners choose to enforce the policy in their test and QA environments so they catch new dependencies before they reach production, then soft-apply the policy in production. Even when soft-applying the policies, monitoring and alerting are still available for RPCs that would be rejected.

As mentioned earlier, the ACL can be based on historical information about RPCs fired by the binary, but that implies allowing the binary to run and serve production data for some time without dependency controls. Also, depending on the variability and diversity of the outgoing traffic, some RPCs might be rare or fired only under special circumstances, such as turn-down or crash. In this case, they might not show up in the historical data and would have to be manually added to the ACL. Because of this, service owners would be well advised to run in soft-apply mode for a long time before enforcing dependency controls in production.

Isolating Groups of Servers

Authorizing RPCs by name is a good way to control RPCs that are uniquely served by a single back end, but this will not cover all of the dependency issues highlighted earlier. One example is a data-integrity case where both production and test servers employ the same RPC. Unless the policy can distinguish between the serving back ends, you cannot block production RPCs from reaching test instances. Additionally, ACLs can offer a tight lock around dependencies, but they are not singly sufficient to prevent dependency cycles.

To prevent these other dependency issues, RPCs can be isolated within a group of servers. The first decision to be made is choosing a DAG model that will dictate the communication between the sets of servers. One simple model that prevents cycles is a graph that represents the expected turnup sequence of the servers. This is a *stacked* or *layered* model of the system, shown in Figure 4. The layered model reinforces that servers will never have dependencies on layers higher than the ones they live on, enabling the sequential bootstrap of the servers from bottom to top. Servers can depend on servers only at the same layer or below.

Services at the bottom layer can rely only on local static data in order to bootstrap, never on data served by another service. At Google, the full set of static data that is necessary to bootstrap a system (for example, compiled binaries, configuration files, system documentation, and root keys) is called the *backpack*.

Figure 4. Stacked model.

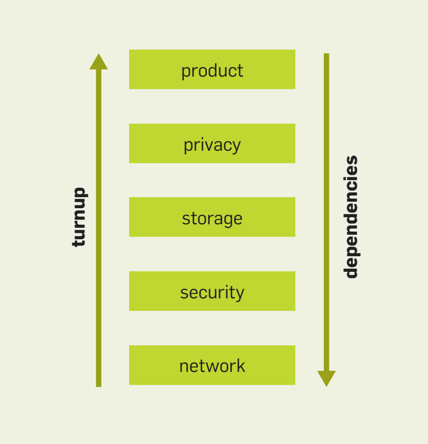
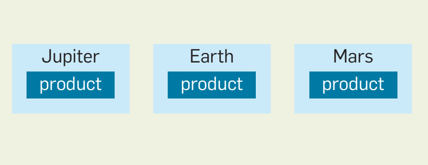


Figure 5. Isolated model.



A layer can have sublayers in order to prevent dependency cycles between servers in the same layer. Sublayering is a smaller problem and can often be handled by a single team without coordination with other teams.

The layered model works well for enabling the bootstrap of a single system, but it still doesn't solve the problem of isolating production and test environments, or restricting communications among different geographic regions. To tackle this kind of problem, you must be able to group servers into disconnected sets; this is called the *isolated model*. Unlike the layered model, where dependencies are allowed in the downward direction, the isolated model disallows dependencies among different components. In the example illustrated in Figure 5, the products Jupiter, Earth, and Mars are not allowed

Figure 6. Pseudocode for model authorization.

```
func isAllowedByModel(rpc, model):
  clientNode = model.resolveNode(rpc.sender)
  serviceNode = model.resolveNode(rpc.receiver)
  return model.hasTransitiveConnection(clientNode, serviceNode)
```

to exchange RPCs with each other. Thus, they are not allowed to depend on each other.

One way to generalize dependency authorization in a DAG model is to let oriented edges represent *can-send-to* relations. Each node on the graph has a self-referencing edge (that is, they can send RPCs to themselves). Also, the can-send-to relation is transitive: if A can send RPCs to B, and B can send RPCs to C, then A can send RPCs to C. Note that if B can send RPCs to A, and B can send RPCs to C, that does not imply that A can send RPCs to C or vice versa. Can-send-to is a directed relation. If there were a can-send-to relation in both directions (from A to B and from B to A), this would constitute a cycle and the model wouldn't be a DAG.

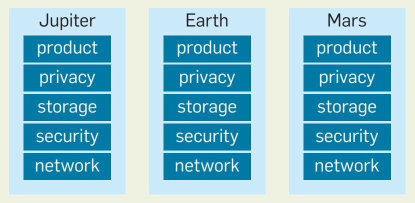
Figure 6 shows how the pseudocode for authorizing RPCs in a DAG model could be written.

The isolated model can be combined with the layered model, allowing the isolated bootstrap of each region to be reinforced, as illustrated in Figure 7.

Figure 8 shows the pseudocode for combining different models.

Be careful when combining models that you do not isolate critical components by combining mutually exclusive models. Usually, simple models are easier to understand and to predict the results of combining, like the layered and isolated models described here. It can be challenging to predict the combined logic for two or more complex models. For example, suppose there are two models based on the geographical locality of machines. It's straightforward to see that assigning locality "Tokyo" from one model and locality "London" from the other model will result in an empty set, since no machine can be physically located in London and Tokyo at the same time. Meanwhile, if there are two tree models based on locality—such as one for city, time zone, and country, and another for metro, voting zone, and country—it might be difficult to verify which combinations of values will return non-empty sets.

Figure 7. Combined model.



Conclusion

With the growth of massive interdependent software systems, dependency management is a crucial part of system and software design. Most organizations will benefit from tracking existing dependencies to help model their latency, SLOs, and security threats. Many will also find it useful to limit the growth of new dependencies for data integrity and to reduce the risk of outages. Modeling infrastructure as a DAG will make it easier to be certain there are no dependencies that will prevent isolated bootstrapping of a system.

Dependencies can be tracked by observing the behavior of a system, but preventing dependency problems before they reach production requires a more active strategy. Implementing dependency control ensures each new dependency can be added to a DAG before it enters use. This gives system designers the freedom to add new dependencies where they are valuable, while eliminating much of the risk that comes from the uncontrolled growth of dependencies.

Figure 8. Pseudocode for multiple model authorization.

```
func isAllowedByAllModels:
  foreach model in modelCollection:
    # Checks if the RPC is allowed by the model.
    if !isAllowedByModel(rpc, model):
      return false

  # If no models reject the RPC, then it should be allowed.
  return true
```

Related articles on queue.acm.org

The Hidden Dividends of Microservices
Tom Killalea
https://queue.acm.org/detail.cfm?id=2956643

A Conversation with Werner Vogels
https://queue.acm.org/detail.cfm?id=1142065

Fail at Scale
Ben Maurer
https://queue.acm.org/detail.cfm?id=2839461

References

1. Beyer, B., Jones, C., Petoff, J., Murphy, N.R. (Eds.). *Site Reliability Engineering: How Google Runs Production Systems.* O'Reilly Media, 2016, 37–40.
2. Beyer, B., Jones, C., Petoff, J., Murphy, N.R. (Eds.). *Site Reliability Engineering: How Google Runs Production Systems.* Chapter 25: Data processing pipelines. O'Reilly Media, 2016.
3. Chang, F. et al. Bigtable: A distributed storage system for structured data, 2006; https://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf.
4. Kachouh, R. The pillars of Squarespace services. Squarespace Engineering; https://engineering.squarespace.com/blog/2017/the-pillars-of-squarespace-services.
5. Lamport, L. Email message sent to a DEC SRC bulletin board, 1987; https://www.microsoft.com/en-us/research/publication/distribution/.
6. Saini, A. How much do bugs cost to fix during each phase of the SDLC? *Synopsis*, 2017; https://www.synopsys.com/blogs/software-security/cost-to-fix-bugs-during-each-sdlc-phase/.
7. Seaton, N. Why fidelity of environments throughout your testing process is important. Electric Cloud; http://electric-cloud.com/blog/2015/09/why-fidelity-of-environments-throughout-your-testing-process-is-important/.
8. Treynor, B., Dahlin, M., Rau, V. and Beyer, B. The calculus of service availability. *acmqueue 15*, 2 2017); https://queue.acm.org/detail.cfm?id=3096459.
9. Ward, R. and Beyer, B. BeyondCorp: A new approach to enterprise security. *;login: 39*, 6 (2014), 6–11; https://ai.google/research/pubs/pub43231.

Silvia Esparrachiari Ghirotti has been at Google for eight years, working in the areas of social products, user data privacy, and fighting abuse. She currently leads the team developing tools for dependency control.

Tanya Reilly is the principal engineer for infrastructure at Squarespace. She previously spent 12 years improving the resilience of low-level services at Google, including introducing a layered model for dependency control.

Ashleigh Rentz is a technical writer whose interests include blameless post mortems and wearable technology. She spent 14 years at Google, most recently producing internal documentation for SRE and Google Cloud Platform.

ACM Welcomes the Colleges and Universities Participating in ACM's Academic Department Membership Program

ACM now offers an Academic Department Membership option, which allows universities and colleges to provide ACM Professional Membership to their faculty at a greatly reduced collective cost.

The following institutions currently participate in ACM's Academic Department Membership program:

- Amherst College
- Appalachian State University
- Armstrong State University
- Ball State University
- Bellevue College
- Berea College
- Binghamton University
- Boise State University
- Bryant University
- Calvin College
- Colgate University
- Colorado School of Mines
- Creighton University
- Cuyahoga Community College
- Edgewood College
- Franklin University
- Gallaudet University
- Georgia Institute of Technology
- Governors State University
- Harding University
- Harvard University
- Hofstra University
- Howard Payne University
- Indiana University Bloomington
- Kent State University
- Klagenfurt University, Austria
- La Sierra University
- Messiah College
- Missouri State University
- Montclair State University
- Mount Holyoke College
- New Jersey Institute of Technology
- Northeastern University
- Ohio State University
- Old Dominion University
- Pacific Lutheran University
- Pennsylvania State University
- Regis University
- Roosevelt University
- Rutgers University
- Saint Louis University
- San José State University
- Shippensburg University
- St. John's University
- Stanford University
- State University of New York at Fredonia
- Trine University
- Trinity University
- Union College
- Union University
- Univ. do Porto, Faculdade de Eng. (FEUP)
- University of Alabama
- University of California, Riverside
- University of California, San Diego
- University of California, Santa Cruz
- University of Colorado Boulder
- University of Colorado Denver
- University of Connecticut
- University of Houston
- University of Illinois at Chicago
- University of Jamestown
- University of Liechtenstein
- University of Maryland, Baltimore County
- University of Memphis
- University of Nebraska at Kearney
- University of Nebraska Omaha
- University of New Mexico
- University of North Carolina at Charlotte
- University of North Dakota
- University of Puget Sound
- University of Southern California
- University of the Fraser Valley
- University of Victoria, BC Canada
- University of Wisconsin–Parkside
- University of Wyoming
- Virginia Commonwealth University
- Wake Forest University
- Wayne State University
- Wellesley College
- Western New England University
- Worcester State University

Through this program, each faculty member receives all the benefits of individual professional membership, including *Communications of the ACM*, member rates to attend ACM Special Interest Group conferences, member subscription rates to ACM journals, and much more.

For more information: www.acm.org/academic-dept-membership

Association for Computing Machinery

DOI:10.1145/3185336

Skill recommendations must be provided when users need them most, without being obtrusive or distracting.

BY RYEN W. WHITE

Skill Discovery in Virtual Assistants

VIRTUAL ASSISTANTS LIKE Amazon Alexa, Microsoft Cortana, Google Assistant, and Apple Siri employ conversational experiences and language-understanding technologies to help users accomplish a range of tasks, from reminder creation to home automation. Voice is the primary means of engagement, and voice-activated assistants are growing in popularity; estimates as of June 2017 put the number of monthly active users of voice-based assistant devices in the U.S. at 36 million.[a] Many are "headless" devices that lack displays. Smart speakers (such as Amazon Echo and Google Home) are among the most popular devices in this category. Speakers are tethered to one location, but there are other settings where voice-activated assistants can be helpful, including automobiles (such as for suggesting convenient locations to help with pending tasks[5]) and personal audio (such as for providing private notifications and suggestions[18]).

Virtual assistant capabilities are commonly called "skills." Skill functionality ranges from basic (such as timers, jokes, and reminders) to more advanced (such as music playback, calendar management, and home automation). Assistant skillsets include both first-party skills and third-party skills. First-party skills comprise the aforementioned basic skill functionality found in many assistants, as well as skills that leverage assistant providers' strengths in such areas as electronic commerce (Amazon Alexa), productivity (Microsoft Cortana), and search (Google Assistant). All major assistants also provide development kits that empower third-party developers to create their own skills for inclusion. Skills can be invoked independently, linked together within a single voice command to invoke a preprogrammed routine, or in a sequence of related skills arranged as required for complex task completion. Despite the significant value virtual assistants can offer, discovery of their capabilities remains a challenge.

Skill Discovery

Skill discovery is a challenge for two primary reasons: the "affordances," or capabilities, of virtual assistants are often unclear and the number of

» key insights

- Virtual assistants in headless devices (such as smart speakers) do not fully convey their rapidly expanding capabilities, sometimes causing users to struggle to understand when and how to best utilize assistant skills.
- This article evaluates methods for recommending relevant virtual-assistant skills using the current task context, identifying complementary contributions from personal and contextual features.
- It also provides design recommendations for how to offer contextual skill recommendations in a timely and unobtrusive manner.

[a] https://www.emarketer.com/Article/Smart-Home-Speakers-Possible-New-Competitor/1015961

IMAGE BY AND4ME

skills available in virtual assistants is increasing rapidly. It is not easy to communicate all that these assistants can do. Users may develop an expectation from prior assistant use or device marketing that assistants will perform certain tasks well. Discovering new skills, especially those that could help with the task at hand, is considerably more difficult.[b] The number and variety of skills available in virtual assistants is also accelerating rapidly, especially with the advent of third-party skill creation through tools (such as the Alexa Skills Kit[c] and the Cortana Skills Kit[d]). Amazon Alexa, the most established skills platform, had more than 26,000 skills available as of December 2017. Figure 1 reports the dramatic increase in the Alexa skillset over time. The pace of growth is such that users often struggle to keep track as new skills are released.

Despite the increase in the number of Alexa skills, it is not clear that the ones being added are actively being utilized. To help determine if they are, I ran an offline experiment. Although usage logs from Alexa were unavailable for this study, it was possible to examine Alexa skill popularity on the http://www.bing.com/ Web search engine. Bing search logs show that over the 18 months between July 2016 and December 2017 inclusive (the maximum time horizon of the logs), the 100 most popular Alexa skills (0.4%) comprised two-thirds of the skill-related search clicks. It is unlikely the 99.6% of skills clicked one-third of the time have little or no utility, addressing only highly specific needs; moreover, there was no correlation between the explicit skill rating on the Alexa skill store and skill use (Pearson $r = -0.05$). A more likely explanation is that users need help to fully understand the capabilities of their virtual assistants and technology to support salient skill discovery is necessary to help them make the best use of these assistants.

[b] Although the focus is on skill discovery, it is also worth acknowledging there are other factors that can affect the use of virtual assistants, including reliance on far-field speech recognition in smart speakers that is typically less accurate than its near-field counterpart,[25] as well as lack of privacy with the broadcast audio in these devices and concern about the social acceptability of using voice-activated assistants in general.[3]

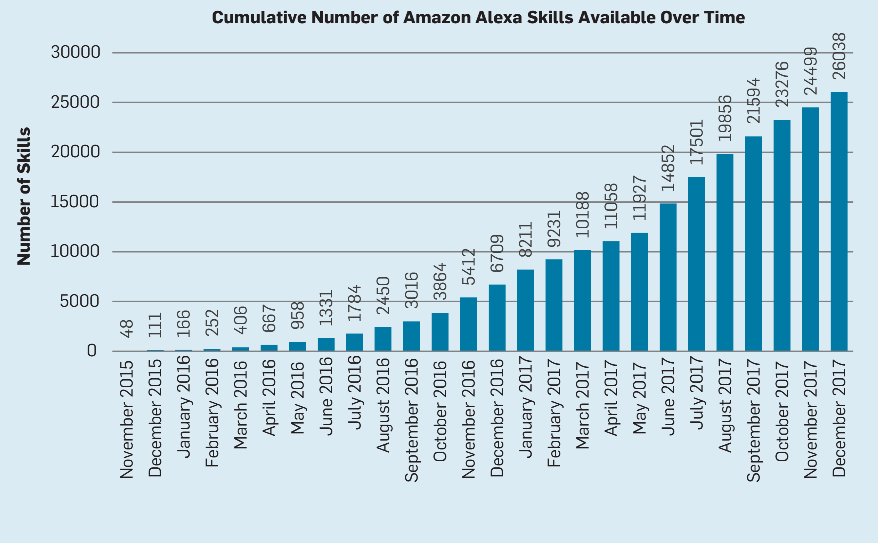
[c] https://developer.amazon.com/alexa-skills-kit

[d] https://developer.microsoft.com/en-us/Cortana

Skill Search

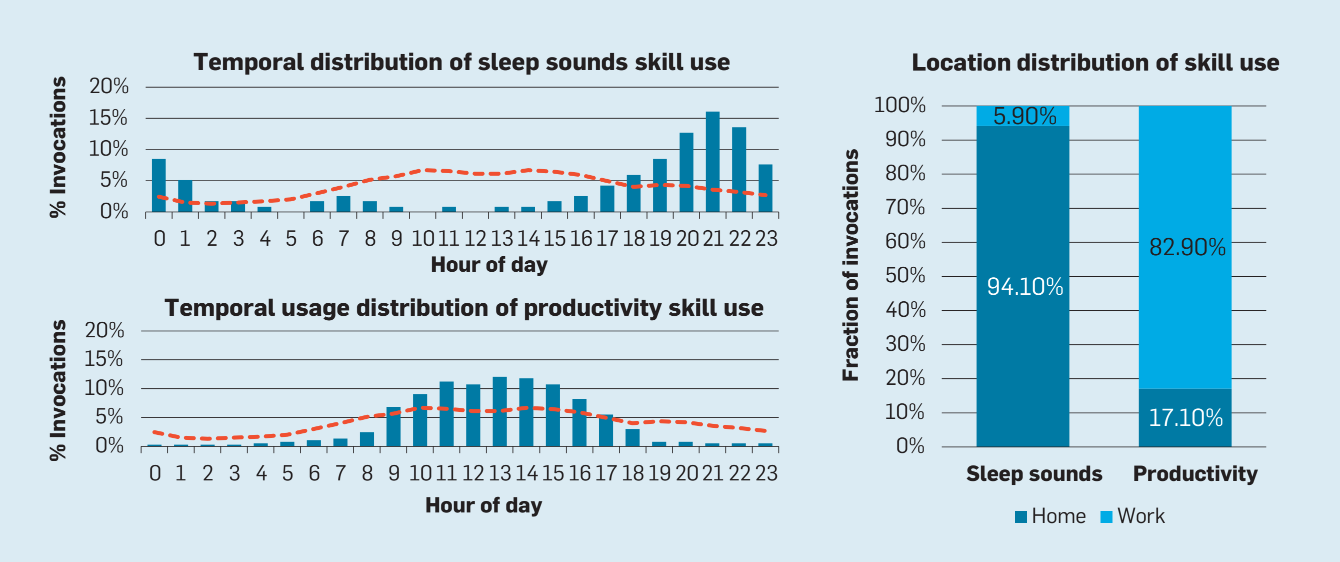
While assistants may support skill searching to locate skills of interest (such as through a dedicated skill like "SkillFinder" on Alexa), these searches may be frustrating and fruitless since smart speakers do not present sufficient clues about their capabilities; many people simply do not know what they can even ask. Although a virtual assistant can help order food from a local restaurant or reserve transportation, unless users are aware of such options, they are unlikely to invoke the skills except by accident or through trial and error. Unclear affordances have long been highlighted by the design community as a reason for inaccurate mental models and the sparse or incorrect use of technology.[14] To address this limitation, assistants support voice prompts (such as "things to try" or "skill of the day") or answer questions (such as "What are your new skills?"). Both are inefficient ways to access new capabilities that require user input and present results through an audio list that is difficult for a typical user to peruse. Moreover, skills discovered through such mechanisms are less likely to relate to users' current tasks since the active context is ignored.

Figure 1. Growth in the number of available skills for Amazon Alexa from November 2015 to December 2017 inclusive; https://www.alexaskillstore.com/. Alexa is a good subject for this analysis given the maturity of Amazon's third-party skills platform.



When users do search for specific skills, they encounter a different experience from what they may be familiar with through Web search engines; search engines are designed to handle general-purpose queries and provide rich visual feedback (a list of search results) that can help people understand what worked well in their query and help them refine their search as needed. In contrast, virtual assistants have a fixed set of capabilities, and smart speakers provide users limited information about what worked well in their search. Failure messages (such as "Sorry, I cannot do this for you right now" or "I am sorry, I do not understand the question") are common but uninformative. If they cannot handle the request and have a display (if invoked through, say, a mobile device), some assistants may resort to presenting Web search results, creating a disjointed experience. Users may be affected by "functional fixedness,"[2] a cognitive bias whereby expectations about what the assistant can do limits the breadth of users' requests. Also, complex answers or multiple search results are difficult to convey through audio output alone. Virtual assistants may elect to delegate results presentation to a companion device (such as a smartphone) when they cannot present the results via audio, but such devices are not always available.

Figure 2. Plots of the time of day at which sleep sounds and productivity skills are invoked. The red dashed line denotes temporal distribution across all skills. Also shown are percentages of all invocations for each skill at two user-defined locations: home and work.



Design Challenges

Developers of virtual assistants face a two-fold challenge: how to set user expectations and educate users about what their assistants can do; and how to help these users while they are learning or even afterward as the assistant adds new skills. Existing onboarding methods provide written instructions in the retail packaging with examples of the types of requests assistants can support, along with periodic email messages that highlight new capabilities over time. These methods mostly promote only first-party skills, yet the power of virtual assistants (and much of the challenge with skill discovery) resides in the silent emergence of tens of thousands (and ultimately many more) third-party skills. Although users can develop effective mental models of products through prolonged use,[20] ever-expanding assistant capabilities make it difficult.

From a consumer-learning perspective, knowledge of virtual-assistant capabilities falls into the domain of "declarative knowledge."[22] Instruction manuals packaged with these products may outline sample functionality, but following the instructions too closely can hinder exploration of product capabilities.[8] Periodic (weekly) email messages may help reveal new skills and reinforce existing skills, but they are shown on a different device from the one(s) used for skill invocation and a different time from when they are needed. The in-situ recommendation of skills (based on the current task) could be an effective way to help ensure users of virtual assistants discover available skills and receive help when it is most needed and welcome.

Although a virtual assistant can help order food from a local restaurant or reserve transportation, unless users are aware of such options, they are unlikely to invoke the skills except by accident or through trial and error.

Role of Context

Context is an important determinant of skill utility. The capabilities people want to employ differ based on contextual factors (such as time and location). For example, consider two skills, one offering ambient relaxing sounds to help promote sleep and one focusing on work productivity. Analyzing usage logs of these two skills from their internal deployment in Cortana with Microsoft employee volunteers reveals notable differences by time of day and location—home and work—both user-specified.

Figure 2 reports that, as expected, the sleep-sounds skill is more likely to be used in the evening and at night than during the day, and much more likely to be used at home than at work. Use of the productivity skill exhibits a wholly different time-and-location profile. Though just one example, it shows that even for an immobile device like a smart speaker, there are still important contextual factors that should influence skill-recommendation priors. The use of context is even more pertinent in mobile scenarios where the context is more dynamic and user tasks are more context-dependent.

Just-in-time information access has been studied extensively.[16] To recommend the right skills at the "right time," or when they are most useful for the current task, it is understood that developers of skills software need rich models of user context. Fortunately, virtual assistants already employ myriad sensors to collect and model context, including physical location, calendar, interests, preferences, search and browse activity, and application activity, all gathered with explicit user consent.

Skill Recommendation

Models for contextual skill recommendation can leverage a range of signals available to the virtual assistant to recommend relevant skills based on the current context. Recommender systems have been studied extensively,[1] and developers of virtual assistants can draw on lessons from that community to assist in the recommendation of skills to users. Salient skills could be suggested in response to an explicit request for assistance from users (or a "cry for help" in more time-critical scenarios,[12] where the need for assistance is more urgent) or be based on external events. For example, a virtual assistant running on a smart speaker deployed in a meeting room can use the commencement of the meeting as a trigger to suggest ways to help make the meeting more productive (such as by taking notes or identifying action items).

Given a set of rich contextual signals, I have been exploring the use of machine-learned skill-recommendation algorithms (in this case, multiple additive regression trees[4]) to recommend skills that are useful in the current context. The models are trained using historic skill usage data. This experimental setup resembles click prediction in search and advertising[17] but reflects several differences, including prediction target (skill used vs. search result or advertisement clicked), setting (open-ended assistant engagement vs. search-engine result page examination), and context (richer and more varied contextual signals available for skill usage prediction).

This research uses records with five months of skill invocations from an internal deployment of a smart speaker powered by Cortana with Microsoft employee volunteers (the same dataset as in Figure 2) and data on the context in which those skills were used as collected by Cortana. The data is split temporally and the first 16 weeks are used for training and the last two weeks for testing. Training and test data is stratified by user. The core principal in this specific instantiation of contextual skill recommendation is that if this or a similar context is observed again, then the skills used previously in that context—by the current user, one or more cohorts, or the population of users—are more likely to be relevant and hence used again. Since the use of historical data puts the focus largely on predicting already-used skills, the study also investigated prediction performance for the subset of test cases where Cortana first observed users trying a skill.

The study further examined the effect of three classes of features used in the learned-skill-recommendation model: popularity, or general popularity of a skill across all users (using only historic usage data from before the skill was invoked); context, or rich contextual signals describing when the skill is used; and personalization, or features corresponding to the user who invoked the skill (such as the popularity of the specific skill for that specific user). These features resemble some that are commonly used in search and recommendation,[1,12] although there are differences (such as lack of an explicit query and desire to focus on suggestion utility) rather than relevance or "interestingness" as the primary measure of model effectiveness. The table here provides more detail on the feature classes.

Classes of features used in the skill-recommendation task. Text features with * are first represented in a continuous semantic space (300-dimension concept vector),[6] and the cosine similarities with both skill name and skill description are then computed. Each cosine measure (such as cosine similarity between the vectors for recent queries and for skill name) becomes a feature in the contextual skill-recommendation model.

Feature class	Example features
Popularity	Historic skill usage count, historic skill user count
Context	Calendar (meeting duration, subject*), visited venues (type, duration, name*), local time, to-do-list items*, search queries (in past 30 mins*, in past 30 days*), applications used*
Personal	Historic skill usage by current user within the past 30 mins, within the past 30 days, and all time

Figure 3. Performance curves for contextual skill recommendation. Results are shown for all skill instances in the test data for several feature classes: popularity, popularity plus context, and popularity plus context and personal (full model).

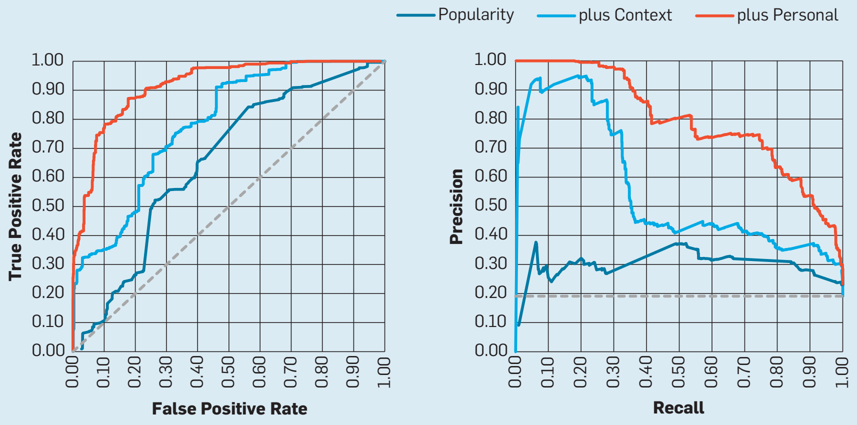


Figure 3 reports the receiver operating characteristic curves and precision-recall curves for the skill-usage prediction task across all skill instances in the test data. The feature contribution analysis starts with popularity features (area under the receiver operating characteristic, or ROC, curve or AUROC is 0.651), adds context features (AUROC increases to 0.786), and then adds personal features (AUROC increases further to 0.918), as in Figure 3a. All three models outperform a baseline of always predicting skill utilization, which reflects 19% precision at 100% recall, as traced by the dotted line in Figure 3b. The model that uses only historic skill-usage frequency performs worst. The results also show that algorithm performance improves considerably, given contextual features (yielding gains in precision) and personal features (yielding gains in recall), as in Figure 3b.[e]

Inspecting the feature weights in the model containing all features (full model) reveals the features with the greatest discriminatory values are those associated with skill popularity (for both the current user and globally), calendar, and short-term interests, in this case, recent Web search queries. While the performance of the recommendation model is promising, reliance on historic data and the importance of such data in the model means there could be limitations on when it can be applied; for example, it could perform worse for new skills for which virtual-assistant developers have little data. To better understand the role of usage data, I reran the experiment, limiting test-data skill invocations to cases where the invoked skill was used by the user for the first time. The results resemble those reported earlier (AUROC = 0.894 for the full model), suggesting the approach may well generalize to unseen

[e] Similar trends are were observed when the order was reversed, to first add personal features and then to add contextual features.

skills. Regardless, this usage-based method is meant only as an illustrative example, and many extensions are possible. Complementary methods from recommender-systems research specifically tailored for cold-start scenarios (such as by Schein et al.[19]) may be helpful in tandem with the usage-based approach.

Although the focus in here is on contextual-skill recommendation, the results show that personal features contribute significantly to the quality of the recommendations generated. Personalization differs from contextualization because it is unique to the user, whereas contextualization could apply to all users in the same context, perhaps in the same meeting. More studies are needed on the use of personal signals, as well as how best to apply them to devices (such as smart speakers) that may be used in social settings (such as a meeting room or a family residence) where there could be simultaneous users of the virtual assistant, some known to the assistant and some unknown. More broadly, use of virtual assistants in social situations raises corporate product-development policy questions around whose virtual assistant should be employed at any given time in such multi-user settings. Speaker-identification technology[15] can help distinguish speakers in these settings to help decide what user profile or even what virtual assistant to apply. A centralized group assistant tied to collective activities (such as meetings[21]) can also help serve as a broker to coordinate tasks between individuals and their virtual assistants, even across assistant brands.

Using Recommendations

Despite the plentiful opportunities around developing more accurate contextual skill-recommendation algorithms, generation of skill recommendations is not the only challenge developers of virtual assistants face when working in this area. They need to also consider how to present the recommendations to users at the right time and in a manner that is not too obtrusive or distracting. The detection of trigger events and selection of the appropriate notification strategy are both particularly important.

As noted, the trigger can be user-initiated and intrinsic, as when a user says, “Hey Cortana, help me,” or event-driven and extrinsic, as when a weather report warns of an impending severe weather event. Proactive scenarios on headless devices can cause frustration and distraction if a device reaches out with an audio message at an inopportune moment. While such intrusion could annoy users, it also has privacy implications tied to sharing potentially sensitive data with a wide audience, as when, say, accidentally notifying all meeting attendees about an upcoming private appointment. Methods have been proposed to better understand the situation and choose a suitable notification strategy.[7] The need for intelligent notifications is not lost on designers of smart speakers; for example, Google Home and Amazon Echo both support subtler approaches for notifying users (such as illuminating an indicator light on the device as an alert regarding a pending notification). It is only when users notice the notification and engage with the device that the notification is provided. However, this delay reduces notification utility considerably.

Context and User Consent

The performance of the contextual-skill-recommendation algorithms is strongly dependent on the signals that are accessible and the degree of consent users are willing to provide to get access to them. Contextual-skill recommendation focuses on suggesting skills to help people when they are in a context where those skills could be useful. Communicating clearly to users the connection between the provision of consent for data access and the provision of useful recommendations is likely to increase the chances users would be willing to grant data access for skill-recommendation purposes.[10]

As virtual assistants begin to manifest in other applications and devices, the range of contextual signals available to them will expand; for example, Facebook’s artificial intelligence assistant, M, indeed chimes in during instant-messaging conversations to suggest relevant content and capabilities.[f] Virtual assistants can leverage signals about the conversation (such as topics discussed and people involved) for contextual modeling. Running skill recommendation within conversations highlights an interesting social dimension to the task of skill recommendations, where the skills suggested could vary based on who is spoken to and that person’s relationship with the speaker, in addition to the topic of the conversation. Skills are also not used in isolation; many scenarios involve skills that are interconnected within a task, as in the restaurant-plus-transportation scenario mentioned earlier. Skill-invocation logs can be mined by virtual assistants for evidence of the co-utilization of these skills within a single task (similar to how guided tours and trails can be mined from historical user-activity data[23]) to help generate relevant skill recommendations. Such recommendations can then be presented proactively, immediately following the use of a related skill.

New Horizons

Virtual assistants have traditionally served users independently. In the past few years, assistant providers have started to partner to leverage their complementary strengths (such as the collaboration announced in August 2017 between Amazon and Microsoft on their Alexa and Cortana personal assistants). Although the focus of this article is generally on assistants recommending their own capabilities, opportunities are emerging to recommend skills among multiple assistants in ways where users could have several assistants, each helping them with one or more aspects of their lives. For example, Cortana might aim to capitalize on Microsoft’s many productivity assets to excel in the personal-productivity domain. In the partnership between Microsoft and Amazon, Alexa could recommend Cortana for productivity-related tasks, and Cortana could recommend Alexa for more consumer-related scenarios, especially in e-commerce. Such partnerships allow developers of virtual assistants to focus more of their resources on strengthening their differentiating capabilities and less on keeping pace with competitors in other areas. Beyond strategic partnerships between well-known major corporate assistant providers, the virtual-assistant-using public is also

[f] https://www.theverge.com/2017/4/6/15200836/facebook-messenger-m-suggestions-ai-assisant

likely to see increased collaborations among multiple skill developers to create compelling new skills and skill combinations. Such partnerships can capitalize on complementary technologies, shared domain knowledge, and other assets (such as data and human capital) that can unlock significant skill differentiation and utility for users. Services that offer skill federation across multiple assistants—much like search engine recommenders (such as Switcheroo,[24] which directs searchers to the optimal search engine for their current query)—will also emerge for virtual assistants, guiding users to the assistant best able to handle the current task or their tasks in general. Interoperability among multiple assistants might also yield considerable user benefit; for example, assistants could share contextual signals to offer skill recommendations and other services of greater utility than any individual virtual assistant alone.

Helping people understand how their assistants can help them is an important step in driving their uptake at scale. This is especially important in smart speakers and similar devices, where capabilities are not immediately obvious given limited display capacity. Looking ahead, I offer the following eight recommendations for virtual assistant developers:

Be proactive. The effectiveness of search (reactive) experiences for the skill-discovery task is influenced by users' expectations regarding affordances in virtual assistants. Proactive skill recommendation methods that understand the current context are a necessary complement to user-initiated skill discovery. Proactive methods may eventually supersede reactive methods as the primary means of engaging with assistant skills, contingent on the emergence of intelligent notification strategies. Virtual assistants could offer proactive support when certain criteria are met, including the availability of rich contextual signals, high confidence scores from recommendation algorithms, and low cost of interruption (such as when the user is assumed by the assistant to not be engaged in another task on the speaker or companion device, as in the recommendation on leveraging companion devices).

Skill-development kits should allow developers to specify for each skill during skill creation the context(s) in which the skill should be recommended.

Timing is everything. Surfacing salient skills means users can more fully leverage the range of support virtual assistants can provide. Presenting users with skill suggestions at the right moment (when they need them) means assistant capabilities are more likely to be remembered in the future.[9] Assistant providers could start by offering support for easily detectable events (such as the start of a scheduled meeting, receipt of a severe-weather alert, or following use of a related skill) and broaden trigger-event coverage thereafter based on task models built from contextual signals, users' contact preferences, and implicit and explicit feedback data.

Use contextual and personal signals. Skills are relevant in one or more contexts. The results of the study showed both contextual and personal signals are important in skill recommendation. A combination of the current context and long-term user activities and interests should be used for this task if that data is available. In addition, skill-development kits should allow developers to specify for each skill during skill creation the context(s) in which the skill should be recommended.

Examine additional signals. There is a range of contextual and personal signals virtual assistants do not have access to today (such as conversations in the room where a smart speaker is located, food being consumed, and television shows being watched) that could correlate with the invocation of skills and enable more targeted recommendations. Virtual-assistant developers should investigate what subset of these contexts is most likely to yield the best improvements in the accuracy of skill recommendations and explore the feasibility of collecting these signals at a large scale. They also need to engage with users to understand what new signals they are comfortable sharing with their assistant.

Consider privacy and utility. User privacy is paramount. If developers and their employers expect users to provide access to the contextual and personal signals required by skill-recommendation algorithms, they must clearly show signal value. Offering the right help at the right moment and attributing it to the permissioned data access via recommendation explanations could serve to demonstrate the utility that can be derived from data sharing. Virtual assistants could offer explanations for each skill recommendation to help users understand how and why it was generated.

Permit multiple recommendations. The focus in this article is the task of predicting the single skill that users would be most likely to use in a given context. Regardless of the richness of any contextual model, the model is often incomplete and lacking in some information about the current task. Having only limited information could thus affect recommendation quality. When confidence in the recommendation model is below a threshold at which a definitive skill would typically be suggested, the assistant should recommend multiple (most-relevant) skills. This process accommodates less-relevant recommendations and meets other requirements of the recommendation task (such as showing the breadth of relevant skills available and supporting serendipitous skill discovery).

Leverage companion devices. Devices without displays may still have access to many screens through WiFi or Bluetooth connectivity, whether on a smartphone, tablet, or desktop PC. Signals from such devices that may not be available on smart speakers (such as recent smartphone apps used) would help enrich the context and assist in providing more-relevant skill suggestions. Given limitations in users' working memory,[13] evaluating result lists is considerably easier if a device has a display. If not, only the top few options can reasonably be vocalized by the assistant for consideration by users. Virtual assistants running on headless devices could use proximal devices with screens to better understand user tasks and display additional content to augment voice-only interaction.

Support continuous learning. Recommendations are needed when users are new to their virtual assistant. However, since skill volume grows silently and quickly over time (see Figure 1 for an example of such growth on Alexa), I foresee there will always be a requirement for assistants to offer suggestions to their users on how they can best help them with their current task. To help improve user understanding, virtual assistants could occasionally suggest new skills based on their users' past skill usage. An appropriate format and time for such suggestions could be through an instructive tip at what may represent a teachable moment immediately following the use of a related skill. As mentioned, developing a notification strategy needs careful attention, given the need to balance the intrusiveness of alerting (especially audio alerting) vs. guiding users toward skills when they need them most.

Conclusion

Learning all that virtual assistants can do or relying on periodic skill-update email messages from their developers is insufficient for a user to make the most of such skills. Unlike apps, which are popular on smartphones and tablets, assistant skills are most likely to be invoked on headless devices that lack displays, increasing dependence on skill finding and limiting skill discovery. The limitations of browsing to discover new knowledge are well understood.[11] Even devices with screens, including Amazon's Echo Show, are limited in the number of recommendations they can present to users and would benefit from algorithms that leverage contextual and personal cues for skill recommendation. Looking ahead, the user-perceived utility of virtual assistants, especially as they manifest in smart speakers and other headless devices (such as personal audio), will depend largely on their ability to proactively identify and share skills that help their users at the moment they need that help the most.

References

1. Adomavicius, G. and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17*, 6 (June 2005), 734–749.
2. Duncker, K. and Lees, L.S. On problem solving. *Psychological Monographs 58*, 5 (1945), i.
3. Easwara Moorthy, A. and Vu, K.P.L. Privacy concerns for use of voice-activated personal assistant in the public space. *International Journal of Human-Computer Interaction 31*, 4 (2015), 307–335.
4. Friedman, J., Hastie, T., and Tibshirani, R. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics 28*, 2 (2000), 337–407.
5. Horvitz, E. and Krumm, J. Some help on the way: Opportunistic routing under uncertainty. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (Pittsburgh, PA, Sept. 5–8). ACM Press, New York, 2012, 371–380.
6. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. Learning deep-structured semantic models for Web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management* (San Francisco, CA, Oct. 27–Nov. 1). ACM Press, New York, 2013, 2333–2338.
7. Horvitz, E. and Apacible, J. Learning and reasoning about interruption. In *Proceedings of the Fifth International Conference on Multimodal Interfaces* (Vancouver, BC, Canada, Nov. 5–7). ACM Press, New York, 2003, 20–27.
8. Lakshmanan, A. and Krishnan, H.S. The Aha! experience: Insight and discontinuous learning in product usage. *Journal of Marketing 75*, 6 (Nov. 2011), 105–123.
9. Kester, L., Kirschner, P.A., van Merriënboer, J.J., and Baumer, A. Just-in-time information presentation and the acquisition of complex cognitive skills. *Computers in Human Behavior 17*, 4 (July 2001), 373–391.
10. Krause, A. and Horvitz, E. A utility-theoretic approach to privacy in online services. *Journal of Artificial Intelligence Research 39* (Nov. 2010), 633–662.
11. Marchionini, G. and Shneiderman, B. Finding facts vs. browsing knowledge in hypertext systems. *Computer 21*, 1 (Jan. 1988), 70–80.
12. Mishra, N., White, R.W., Ieong, S., and Horvitz, E. Time-critical search. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Gold Coast, QLD, Australia, July 6–11). ACM Press, New York, 2014, 747–756.
13. Miyake, A. and Shah, P., Eds. *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control.* Cambridge University Press, New York, 1999.
14. Norman, D.A. Affordance, conventions, and design. *Interactions 6*, 3 (May 1999), 38–43.
15. Reynolds, D.A., Quatieri, T.F., and Dunn, R.B. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing 10*, 1-3 (Jan. 2000), 19–41.
16. Rhodes, B.J. and Maes, P. Just-in-time information retrieval agents. *IBM Systems Journal 39*, 3.4 (2000), 685–704.
17. Richardson, M., Dominowska, E., and Ragno, R. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, AB, Canada, May 8–12). ACM Press, New York, 2007, 521–530.
18. Sawhney, N. and Schmandt, C. Nomadic radio: Speech and audio interaction for contextual messaging in nomadic environments. *ACM Transactions on Computer-Human interaction 7*, 3 (Sept. 2000), 353–383.
19. Schein, A.I., Popescul, A., Ungar, L.H., and Pennock, D.M. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Tampere, Finland, Aug. 11–15). ACM Press, New York, 2002, 253–260.
20. Schilling, M.A. A 'small-world' network model of cognitive insight. *Creativity Research Journal 17*, 2-3 (2005), 131–154.
21. Tur, G., Stolcke, A., Voss, L., Peters, S., Hakkani-Tur, D., Dowding, J., Favre, B., Fernández, R., Frampton, M., Frandsen, M., and Frederickson, C. The CALO meeting assistant system. *IEEE Transactions on Audio, Speech, and Language Processing 18*, 6 (Aug. 2010), 1601–1611.
22. Van Osselaer, Stijn M.J., and Janiszewski, C. Two ways of learning brand associations. *Journal of Consumer Research 28*, 2 (Sept. 2001), 202–223.
23. Wexelblat, A. and Maes, P. Footprints: History-rich tools for information foraging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada, Apr. 13–18). ACM Press, 1999, 270–277.
24. White, R.W., Richardson, M., Bilenko, M., and Heath, A.P. Enhancing Web search by promoting multiple search engine use. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Singapore, July 20–24). ACM Press, New York, 2008, 43–50.
25. Wölfel, M. and McDonough, J. *Distant Speech Recognition.* John Wiley & Sons, Inc., New York, 2009.

Ryen W. White (ryenw@microsoft.com) is a Partner Researcher and Research Manager at Microsoft Research AI, Redmond, WA, USA.

DOI:10.1145/3186277

Simplicity, small size, portability, and embeddability set Lua apart from other scripting languages.

BY ROBERTO IERUSALIMSCHY, LUIZ HENRIQUE DE FIGUEIREDO, AND WALDEMAR CELES

A Look at the Design of Lua

LUA IS A scripting language developed at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio) that has come to be the leading scripting language for video games worldwide.[3,7] It is also used extensively in embedded devices like set-top boxes and TVs and in other applications like Adobe Photoshop Lightroom and Wikipedia.[14] Its first version was released in 1993. The current version, Lua 5.3, was released in 2015.

Though mainly a procedural language, Lua lends itself to several other paradigms, including object-oriented programming, functional programming, and data-driven programming.[5] It also offers good support for data description, in the style of JavaScript and JSON. Data description was indeed one of our main motivations for creating Lua, some years before the appearance of XML and JavaScript.

Our motto in the design of Lua has always been "mechanisms instead of policies." By policy, we mean a methodical way of using existing mechanisms to build a new abstraction. Encapsulation in the C language provides a good example of a policy. The ISO C specification offers no mechanism for modules or interfaces.[9] Nevertheless, C programmers leverage existing mechanisms (such as file inclusion and external declarations) to achieve those abstractions. On top of such basic mechanisms provided by the C language, policy adds several rules (such as "all global functions should have a prototype in a header file" and "header files should not define objects, only declare them"). Many programmers do not know these rules (and the policy as a whole) are not part of the C language.

Accordingly, in the design of Lua, we have replaced addition of many different features by creating instead only a few mechanisms that allow programmers to implement such features themselves.[6] The motto leads to a design that is economical in concepts. Lua offers exactly one general mechanism for each major aspect of programming: tables for data; functions for abstraction; and coroutines for control. On top of these building blocks, programmers implement several other features, including modules, objects, and environments, with the aid of minimal additions (such as syntactic sugar) to the language. Here, we look at how this motto has worked out in the design of Lua.

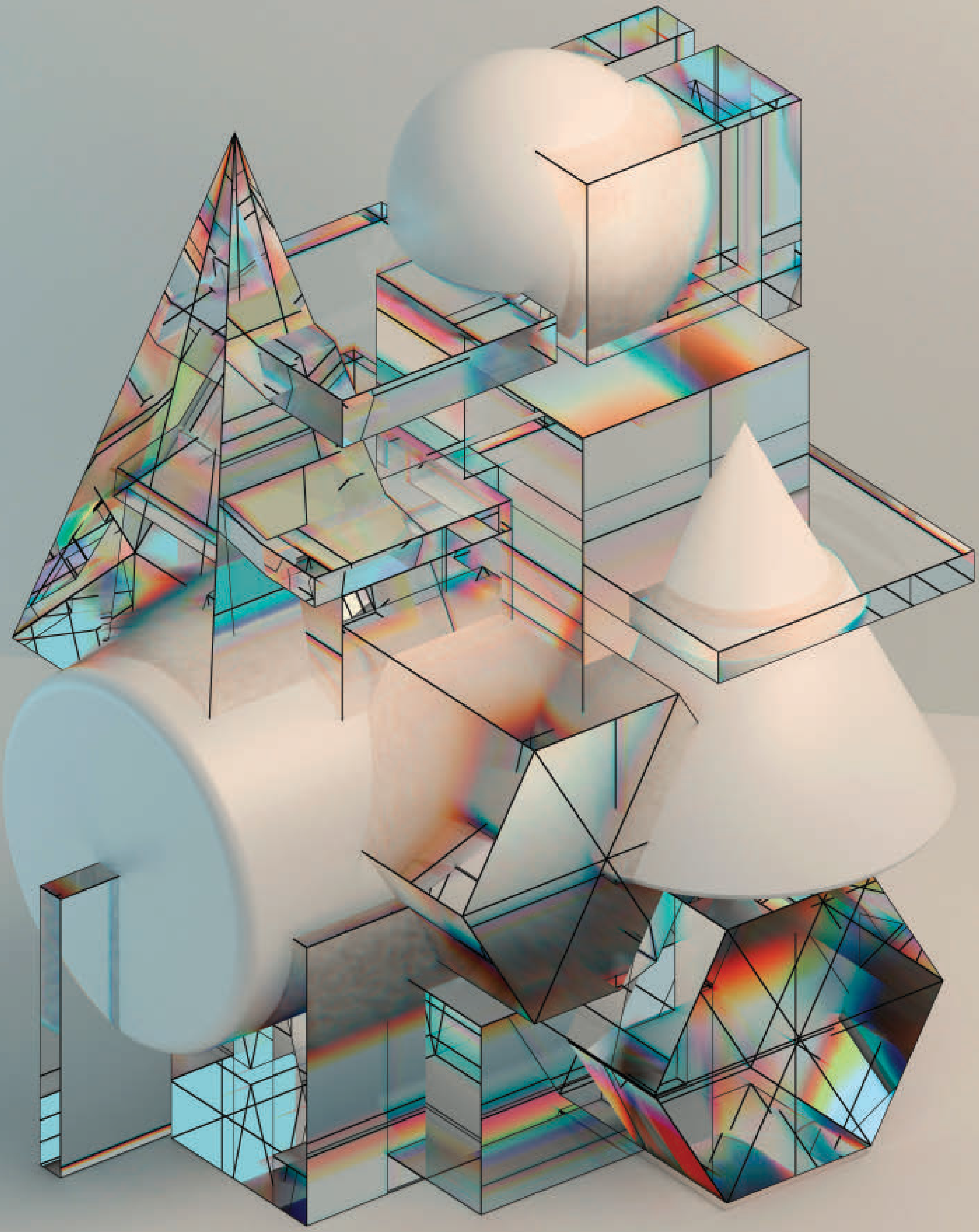
Design Goals

Like other scripting languages, Lua has dynamic types, dynamic data structures, garbage collection, and an `eval`-like functionality. Consider Lua's particular set of goals:

» **key insights**

- **What sets Lua apart from other scripting languages is its particular set of goals: simplicity, small size, portability, and embeddability.**
- **The entire implementation of Lua has 25,000 lines of C code; the binary for 64-bit Linux has 200k bytes.**
- **Since its inception, Lua was designed to interoperate with other languages.**

IMAGE BY BUG FISH



Simplicity. Lua aims to offer only a few powerful mechanisms that can address several different needs, instead of myriad specific language constructs, each tailored for a specific need. The Lua reference manual is small, with approximately 100 pages covering the language, its standard libraries, and the API with C;

Small size. The entire implementation of Lua consists of 25,000 lines of C code; the binary for 64-bit Linux has 200k bytes. Being small is important for both portability, as Lua must fit into a system before running there, and embedding, as it should not bloat the host application that embeds it;

Portability. Lua is implemented in ISO C and runs in virtually any system with as little as 300k bytes of memory. Lua runs in all mainstream systems and also on mainframes, inside OS kernels (such as the NetBSD kernel), and on “bare metal” (such as NodeMCU running on the ESP8266 microcontroller); and

Embeddability. Lua was designed since its inception to interoperate with other languages, both by extending—allowing Lua code to call functions written in a foreign language—and by embedding—allowing foreign code to call functions written in Lua.[8] Lua is thus implemented not as a standalone program but as a library with a C API. This library exports functions that create a new Lua state, load code into a state, call functions loaded into a state, access global variables in a state, and perform other basic tasks. The standalone Lua interpreter is a tiny application written on top of the library.

These goals have had a deep impact on our design of Lua. Portability restricts what the standard libraries can offer to what is available in ISO C, including date and time, file and string manipulation, and basic mathematical functions. Everything else must be provided by external libraries. Simplicity and small size restrict the language as a whole. These are the goals behind the economy of concepts for the language. Embeddability has a subtler influence. To improve embeddability, Lua favors mechanisms that can be represented naturally in the Lua-C API. For instance, Lua tries to avoid or reduce the use of special syntax for a new mechanism, as syntax is not accessible through an API. On the other hand, mechanisms exposed as functions are naturally mapped to the API.

Following the motto “mechanisms instead of policies” has a clear impact on simplicity and small size. It also affects embeddability by breaking complex concepts into simpler ones that are easier to represent in the API.

Lua supports eight data types: nil, boolean, number, string, userdata, table, function, and thread, which represents coroutines. The first five are no surprise. The last three give Lua its flavor and are the ones we discuss here. However, given the importance of embeddability in the design of Lua, we first briefly introduce the interface between Lua and its host language.

Figure 1. A C program using the Lua library.

```
#include <stdio.h>
#include “lauxlib.h”
#include “lualib.h”

int main (int argc, char **argv) {
  // create a new state
  lua_State *L = luaL_newstate();
  // load the standard libraries
  luaL_openlibs(L);
  // try to load the given file and then
  // call the resulting function
  if (luaL_loadfile(L, argv[1]) != LUA_OK ||
      lua_pcall(L, 0, 0, 0) != LUA_OK) {
    // some error occurred; print the error message
    fprintf(stderr, “lua: %s\n”, lua_tostring(L, -1));
  }
  else {  // code ran successfully
    lua_getglobal(L, “pi”);
    printf(“pi: %f\n”, lua_tonumber(L, -1));
  }
  lua_close(L);   // close the state
  return 0;
}
```

The Lua–C API

To illustrate the concept of embedding in Lua, consider a simple example of a C program using the Lua library. Take this tiny Lua script, stored in a file

```
pi = 4 * math.atan(1)
```

Figure 1 shows a C program that runs the script and prints the value of pi. The first task is to create a new state and populate it with the functions from the standard libraries (such as math.atan). The program then calls luaL _ loadfile to load (precompile) the given source file into this state. In the absence of errors, this call produces a Lua function that is then executed by lua _ pcall. If either loadfile or pcall raises an error, it produces an error message that is printed to the terminal. Otherwise, the program gets the value of the global variable pi and prints its value.

The data exchange among these API calls is done through an implicit stack in the Lua state. The call to luaL _ loadfile pushes on the stack either a function or an error message. The call to lua _ pcall pops the function from the stack and calls it. The call to lua _ getglobal pushes the value of the global variable. The call to lua _ tonumber projects the Lua value on top of the stack to a double. The stack ensures these values remain visible to Lua while being manipulated by the C code so they cannot be collected by Lua’s garbage collector.

Besides the functions used in this simple example, the Lua–C API (or “C API” for short) offers functions for all kinds of manipulation of Lua values, including pushing C values (such as numbers and strings) onto the stack, calling functions defined by the script, and setting variables in the state.

Tables

“Table” is the Lua term for associative arrays, or “maps.” A table is just a collection of entries, which are pairs ⟨key, value⟩.

Tables are the sole data-structuring mechanism in Lua. Nowadays, maps are available in most scripting languages, as well as in several non-scripting ones, but in Lua maps are ubiquitous. Indeed, Lua programmers use tables not only for all kinds of data structures (such as records, arrays, lists, sets, and sparse matrices) but also for higher-level constructs (such as modules, objects, and environments).

Programmers implement records using tables whose indices are strings representing field names. Lua supports records with syntactic sugar, translating a field reference like `t.x` to a table-indexing operation `t["x"]`.

Lua offers constructors, expressions that create and initialize tables. The constructor `{}` creates an empty table. The constructor `{x=10,y=20}` creates a table with two entries, one mapping the string `"x"` to the integer 10, the other mapping `"y"` to 20. Programmers see this table as a record with fields `"x"` and `"y"`.

Programmers implement arrays with tables whose indices are positive integers. Constructors also support this usage. For example, the expression `{10,20,30}` creates a table with three entries, mapping 1 to 10, 2 to 20, and 3 to 30. Programmers see the table as an array with three elements.

Arrays have no special status in the semantics of Lua; they are just ordinary tables. However, arrays pervade programming. Therefore, implementation of tables in Lua gives special attention to their use as arrays. The internal representation of a table in Lua has two parts: an array and a hash.[7] If the array part has size N, all entries with integer keys between 1 and N are stored in the array part; all other entries are stored in the hash part. The keys in the array part are implicit and do not need to be stored. The size N of the array part is computed dynamically, every time the table has to rehash as the largest power of two such that at least half the elements in the array part will be filled. A generic access (such as `t[i]`) first checks whether `i` is an integer in the range $[1, N]$; this is the most common case and the one programmers expect to be fast. If so, the operation gets the value in the array; otherwise, it accesses the hash. When accessing record fields (such as `t.x`) the Lua core knows the key is a string and so skips the array test, going directly to the hash.

Lua offers exactly one general mechanism for each major aspect of programming: tables for data; functions for abstraction; and coroutines for control.

An interesting property of this implementation is that it gives sparse arrays for free. For instance, when a programmer creates a table with three entries at indices 5, 100, and 3421, Lua automatically stores them in the hash part, instead of creating a large array with thousands of empty slots.

Lua also uses tables to implement weak references. In languages with garbage collection, a weak reference is a reference to an object that does not prevent its collection as garbage.[10] In Lua, weak references are implemented in weak tables. A weak table is thus a table that does not prevent its contents from being collected. If a key or a value in an entry is collected, that entry is simply removed from the table; we discuss later how to signal that a table is weak. Weak tables in Lua also subsume ephemerons.[4]

Weak tables seem to contradict the motto "mechanisms instead of policies" because weak reference is a more basic concept than weak table. Weak tables would then be a policy, a particular way of using weak references. However, given the role of tables in Lua, it is natural to use them to support weak references without introducing yet another concept.

Functions

Lua supports first-class anonymous functions with lexical scoping, informally known as closures.[13] Several non-functional languages nowadays (such as Go, Swift, Python, and JavaScript) offer first-class functions. However, to our knowledge, none uses this mechanism as pervasively as Lua.

All functions in Lua are anonymous. This is not immediately clear in the standard syntax for defining a function

```
function add (x, y)
  return x + y
end
```

Nevertheless, this syntax is just syntactic sugar for an assignment of an anonymous function to a variable

```
add = function (x, y)
  return x + y
end
```

Most dynamic languages offer some kind of `eval` function that evaluates a piece of code produced at runtime. Instead of `eval`, Lua offers a `load` function that, given a piece of source code, returns a function equivalent to that code. We saw a variant of `load` in the C API in the form of `luaL_loadfile`. Consider the following piece of code

```
local id = 0
function genid ()
  id = id + 1
  return id
end
```

When one loads it, the function `load` returns an anonymous function equivalent to the following code

```
function ()
  local id = 0
  function genid ()
    id = id + 1
    return id
  end
end
```

So, if a programmer loads Lua code stored in a string and then calls the resulting function, the programmer gets the equivalent of `eval`.

We use the term "chunk" to denote a piece of code fed to `load` (such as a source file). Chunks are the compilation units of Lua. When a programmer uses Lua in interactive mode, the Read-Eval-Print Loop (REPL) handles each input line as a separate chunk.

The function `load` simplifies the semantics of Lua in two ways: First, unlike `eval`, `load` is pure and total; it has no side effects and it always returns a value, either a function or an error message; second, it eliminates the distinction between "global" code and "function" code, as in the previous chunk of code. The variable `id`, which in the original code appears outside any function, is seen by Lua as a local variable in the enclosing anonymous function representing the script. Through lexical scoping, `id` is visible to the function `genid` and preserves its value between successive calls to that function. Thus, `id` works like a static variable in C or a class variable in Java.

Exploring Tables and Functions

Despite their apparent simplicity—or because of it—tables and functions form a basis for several other mechanisms in Lua, including modules, object-oriented programming, and exception handling. We now discuss some of them, emphasizing how they contribute to Lua's design goals.

Figure 2. A simple module in Lua.

```
local M = {}

function M.new (x, y)
  return {x = x, y = y}
end

function M.add (u, v)
  return M.new(u.x+v.x, u.y+v.y)
end

function M.norm (v)
  return math.sqrt(v.x^2 + v.y^2)
end

return M
```

Figure 3. A module in Lua using environments.

```
local sqrt = math.sqrt
local _ENV = {}

function new (x, y)
  return {x = x, y = y}
end

function add (u, v)
  return new(u.x+v.x, u.y+v.y)
end

function norm (v)
  return sqrt(v.x^2 + v.y^2)
end

return _ENV
```

Modules. The construction of modules in Lua is a nice example of the use of first-class functions and tables as a basis for other mechanisms. At runtime, a module in Lua is a regular table populated with functions, as well as possibly other values (such as constants). Consider this Lua fragment

```
print(math.sin(math.pi/6))
  --> 0.5
```

Abstractly, programmers read this code as calling the `sin` function from the standard math module, using the constant `pi` from that same module. Concretely, the language sees `math` as a variable (created when Lua loaded its standard libraries) containing a reference to a table. That table has an entry with the key `"sin"` containing the sine function and an entry `"pi"` with the value of π.

Statically, a module is simply the chunk that creates its corresponding table. Figure 2 shows a standard idiom for defining a simple module in Lua. The code creates a table in the local variable `M`, populates the table with some functions, and returns that table. Recall that Lua loads any chunk as the body of an enclosing anonymous function; this is how one should read that code. The variable `M` is local to that enclosing function and the final statement returns from that function.

Once defined in a file `mymodule.lua`, a programmer can use that module with code like this[a]

```
local vec = require "mymodule"
print(vec.norm(vec.new(10, 10)))
   --> 14.142135623731
```

In it, `require` is a regular function from the standard library; when the single argument to a function is a literal string, the code can omit the parentheses in the call. If the module is not already loaded, `require` searches for an appropriate source for the given name (such as by looking for files in a list of paths), then loads and runs that code, and finally returns what the code returns. In this example, `require` returns the table `M` created by the chunk.

Lua leverages tables, first-class functions, and `load` to support modules. The only addition to the language is the function `require`. This economy is particularly relevant for an embedded language like Lua. Because `require` is a regular function, it cannot create local variables in the caller's scope. Thus, in the example using `"mymodule"`, the programmer had to define explicitly the local variable `vec`. Yet this limitation gives programmers the ability to give a local name to the module.

On the one hand, the construction of modules in Lua is not as elegant as a dedicated language mechanism could be, with explicit import and export lists and other refinements, as in the "import machinery" in Python.[12] On the other hand, this construction has a clear semantics that requires no

[a] To test these pieces of code interactively, remove the `local` from the variable initializations. In interactive mode, Lua loads each line as an independent chunk. A local variable is thus visible only in the line where it was defined.

further explanation. It also has an inexpensive implementation. Finally, and also quite important, it has an easy integration with the C API: One can easily create modules in C; create mixed modules with some functions defined in Lua and others in C; and for C code call functions inside modules. The API needs no additional mechanisms to do these tasks; all it needs is the existing Lua mechanisms to manipulate tables and functions.

Environments. Local variables in Lua follow a strict lexical scoping discipline. A local variable can be accessed only by code that is lexically written inside its scope. Lexical scoping implies that local variables are one of the few constructions that do not cross the C API, as C code cannot be lexically inside Lua code.

A program in Lua can be composed of multiple chunks (such as multiple modules) loaded independently. Lexical scoping implies that a module cannot create local variables for other chunks. Variables like `math` and `require`, created by the standard libraries, should thus be created as global variables. However, using global variables in a large program can easily lead to overly complex code, entangling apparently unrelated parts of a program. To circumvent this conflict, Lua does not have global variables built into the language. Instead, it offers a mechanism of environments that, by default, gives the equivalent of global variables. Nevertheless, as we show later in this article, environments allow other possibilities.

Recall that any chunk of code in Lua is compiled as if inside an anonymous function. Environments add two simple rules to this translation: First, the enclosing anonymous function is compiled as if in the scope of a local variable named `_ENV`; and second, any free variable `id` in the chunk is translated to `_ENV.id`. For example, Lua loads the chunk `print(v)` as if it was written like this

```
local _ENV = <<some given value>>
return function ()
    _ENV.print(_ENV.v)
end
```

By default, `load` initializes `_ENV` with a fixed table, called the global environment. All chunks thus share this same environment by default, giving the illusion of global variables; in the chunk just mentioned, both `v` and `print` refer to fields in that table and thus behave as global variables. However, both `load` and the code being loaded can modify `_ENV` to any other value. The `_ENV` mechanism allows different scripts to have different environments, functions to be called with different environments, and other variations.

The translation of free variables needs semantic information to determine whether a variable is free. Nevertheless, the translation itself is purely syntactical. In particular, `_ENV` is a regular variable, needing no special treatment by the compiler. The programmer can assign new values to `_ENV` or declare other variables with that name. As an example, consider this fragment

```
do
  local _ENV = {}
  ...
end
```

Inside the `do` block, all free variables refer to fields in the new table `_ENV`. Outside the block, all free variables refer to the default environment.

A more typical use of `_ENV` is for writing modules. Figure 3 shows how to rewrite the simple module of Figure 2 using environments. In the first line, where the code "imports" a function from the `math` module, the environment is still the default one. In the second line, the code sets the environment to a new table that will represent the module. The code then defines the module components directly as free variables; instead of `M.norm`, it uses only `norm`, which Lua translates to `_ENV.norm`. The code ends the module with `return _ENV`.

This method for writing modules has two benefits: First, all external functions and modules must be explicitly imported right at the start; and second, a module cannot pollute the global space by mistake.

Object-oriented programming. Support for object-oriented programming in Lua follows the pattern we have been seeing in this article: It tries to build upon tables and functions, adding only the minimum necessary to the language.

Lua uses a two-tier approach to object-oriented programming. The first is implemented by Lua and the second by programmers on top of the first one. The first tier is class-based. Both objects and classes are tables, and the relation "instance of" is dynamic. Userdata, which represents C values in Lua, can also play the role of objects. Classes are called metatables. In this first tier, a class can define only methods for the standard operators (such as addition, subtraction, and concatenation). These methods are called metamethods.

Figure 4 illustrates how a programmer would use this basic mechanism to perform arithmetic on 2D vectors. The code starts with a table `mt` that would be the metatable for the vectors. The code then defines a function `newVector` to create 2D vectors. Vectors are tables with two fields, `x` and `y`. The standard function `setmetatable` establishes the "instance of" relation

Figure 4. An example of metatables.

```
local mt = {}

function newVector (x, y)
  local p = {x = x, y = y}
  setmetatable(p, mt)
  return p
end

function mt.__add (p1, p2)
  return newVector(p1.x + p2.x, p1.y + p2.y)
end

-- example of use
A = newVector(10, 20)
B = newVector(20, -40)
C = A + B
print(C.x, C.y)      --> 30    -20
```

between a new vector and `mt`. Next, the code defines the metamethod `mt.__add` to implement the addition operator for vectors. The code then creates two vectors, `A` and `B`, and adds them to create a new vector `C`. When Lua tries to evaluate `A+B`, it does not know how to add tables and so checks for an `__add` entry in `A`'s metatable. Given that it finds that entry, Lua calls the function stored there—the metamethod—passing the original operands `A` and `B` as arguments.

The metamethod for the indexing operator `[]` offers a form of delegation in Lua. Lua calls this metamethod, named `__index`, whenever it tries to retrieve the value of an absent key from a table. (For userdata, Lua calls that metamethod for all keys.) For the indexing operation, Lua allows the metamethod to be a function or a table. When `__index` is a table, Lua delegates to that table all access for an index that is absent in the original table, as illustrated by this code fragment

```
Proto = {x = 0, y = 0}
obj = {x = 10}
mt = { __index = Proto}
setmetatable(obj, mt)
print(obj.x) --> 10
print(obj.y) --> 0
```

In the second call to `print`, Lua cannot find the key `"y"` in `obj` and so delegates the access to `Proto`. In the first `print`, as `obj` has a field `"x"`, the access is not delegated.

With tables, functions, and delegation, we have almost all we need for the second tier, which is based on prototypes. In it, programmers represent objects also by tables or userdata. Each object can have a prototype, from which it inherits methods and fields. The prototype of an object `obj` is the object stored in the `__index` field of the metatable of `obj`. One can then write `obj.foo(x)`, and Lua will retrieve the method `foo` from the object's prototype, through delegation.

However, if we stopped here, there would be a flaw in the support for object-oriented programming in Lua. After finding and calling the method in the object's prototype, there would be no way for the method to access the original object, which is the intended receiver. Lua solves this problem through syntactic sugar. Lua translates a "method" definition like

```
function Proto:foo (x)
  ...
end
```

to a function definition:

```
function Proto.foo (self, x)
  ...
end
```

Likewise, Lua translates a "method" call `obj:foo(x)` to `obj.foo(obj,x)`. When the programmer defines a "method"—a function using the colon syntax—Lua adds a hidden parameter `self`. When the programmer calls a "method" using the colon syntax, Lua provides the receiver as the argument to the `self` parameter. There is no need to add classes, objects, or methods to the language, merely syntactic sugar.

Figure 5. A simple prototype-based design in Lua.

```
local Account = {bal = 0}
local mt = {__index = Account}

function Account:new ()
  local obj = {}
  setmetatable(obj, mt)
  return obj
end

function Account:deposit (amount)
  self.bal = self.bal + amount
end
function Account:balance ()
  return self.bal
end

return Account
```

Figure 5 illustrates these concepts. First the code creates a prototype, the table `Account`. The code then creates a table `mt` to be used as the metatable for instances of `Account`. It then adds three methods to the prototype: one for creating instances, one for making deposits, and one for retrieving the account's balance. Finally, it returns the prototype as the result of this module.

Assuming the module is in the file `Account.lua`, the following lines exercise the code

```
Account = require “Account”
acc = Account:new()
acc:deposit(1000)
print(acc:balance()) -->
1000
```

First, the code requires the module, then it creates an account; `acc` will be an empty table with `mt` as its metatable. De-sugared, the next line reads as `acc.deposit(acc,1000)`. The table `acc` does not have a `deposit` field, so Lua delegates that access to the table in the metatable's `__index` field. The result of the access is the function `Account.deposit`. Lua then calls that function, passing `acc` as the first argument (`self`) and `1000` as the second argument (`amount`). Inside the function, Lua will again delegate the access `self.bal` to the prototype because `acc` does not yet have a field `bal`. In subsequent calls to balance, Lua will find a field `bal` in the table `acc` and use that value. Distinct accounts thus have separate balances but share all methods.

The access to a prototype in the metatable's `__index` is a regular access, meaning prototypes can be chained. As an example, suppose the programmer adds the following lines to the previous example

```
Object = {name = “no name”}
setmetatable(Account,
   { _ _index = Object})
```

When Lua evaluates `acc.name`, the table `acc` does not have a `name` key, so Lua tries the access in its prototype, `Account`. That table also does not have that key, so Lua goes to `Account`'s prototype, the table `Object`, where it finally finds a `name` field.

Figure 6. Accounts with private fields.

```
local bal = {}
setmetatable(bal, {__mode = “k”})

local Account = {}
local mt = {__index = Account}

function Account:new ()
  local obj = {}
  setmetatable(obj, mt)
  bal[obj] = 0
  return obj
end

function Account:deposit (amount)
  bal[self] = bal[self] + amount
end

function Account:balance ()
  return bal[self]
end

return Account
```

The programmer can keep the balances private by storing them outside the object table, as shown in Figure 6. The key difference between this version and the one in Figure 5 is the use of bal[self] instead of self.bal to denote the balance of an account. The table bal is what we call a dual table. The call to setmetatable in the second line causes this table to have weak keys, thus allowing an account to be collected when there are no other references to it in the program. The fact that bal is local to the module ensures no code outside that module can see or tamper with an account's balance, a technique that is handy whenever one needs a private field in a structure.

An evaluation of Lua's support for object-oriented programming is not very different from the evaluation of the other mechanisms we have discussed so far. On the one hand, object-oriented features in Lua are not as easy to use as in other languages that offer specific constructs for the task. In particular, the colon syntax can be somewhat confusing, mainly for programmers who are new to Lua but have some experience with another object-oriented language. Lua needs that syntax because of its economy of concepts that avoids introducing the concept of method when the existing concept of function will suffice.

On the other hand, the semantics of objects in Lua is simple and clear. Also, the implementation of objects in Lua is flexible. Because method selection and the variable self are independent, Lua does not need additional mechanisms to call methods from other classes (such as “super”). Finally, this design is friendly to the C API. All it needs is basic manipulation of tables and functions, plus the standard function setmetatable. Lua programmers can implement prototypes in Lua and create userdata instances in C, create prototypes in C and instances in Lua, and define prototypes with some methods implemented in Lua and others in C. All these pieces work together seamlessly.

Exception handling. Exception handling in Lua is another mechanism that relies on the flexibility of functions. Several languages offer a try-catch construction for exception handling; any exception in the code inside a try clause jumps to a corresponding catch clause. Lua does not offer such a construction, mainly because of the C API.

More often than not, exceptions in a script are handled by the host application. A syntactic construction like try-catch is not easily mapped into an API with a foreign language. Instead, the C API packs exception-handling functionality into the higher-order function lua_pcall (“protected call”) we discussed when we visited the C API earlier in this article. The function pcall receives a function as an argument and calls that function. If the provided function terminates without errors, pcall returns true; otherwise, pcall catches the error and returns false plus an error object, which is any value given when the error was raised. Regardless of how pcall is implemented, it is exposed in the C API as a conventional function. The C API also offers a function to raise errors, called lua_error, whose only argument is the error object. The function error also appears in the C API as a regular function despite the fact that it never returns.

Both lua_pcall and lua_error are reflected into Lua via the standard library. In languages that support try-catch, typical exception-handling code looks like this

```
try {
  <<protected code>>
}
catch (errobj) {
  <<exception handling>>
}
```

The equivalent code in Lua is like this

```
local ok, errobj =
pcall(function ()
  <<protected code>>
end)

if not ok then
  <<exception handling>>
end
```

In this translation, anonymous functions with proper lexical scoping play a central role. Except for statements that invoke escape continuations (such as break and return), everything else can be written inside the protected code as if written in the regular code.

The use of pcall for exception handling has pros and cons similar to those for modules. On the one hand, the code may not look as elegant as in other languages that support the traditional try. On the other hand, it has a clear semantics. In particular, questions like “What happens with exceptions inside the catch clause?” have an obvious answer. Moreover, it has a clear and easy integration with the C API; it is exposed through conventional

Figure 7. A simple example of a coroutine in Lua.

```
co = coroutine.create(function (x)
    print(x)    --> 10
    x = coroutine.yield(20)
    print(x)    --> 30
    return 40
end)

print(coroutine.resume(co, 10))   --> 20
print(coroutine.resume(co, 30))   --> 40
```

functions; and Lua programs can raise errors in Lua and catch them in C and raise errors in C and catch them in Lua.

Coroutines

Like associative arrays and first-class functions, coroutines are a well-established concept in programming. However, unlike tables and first-class functions, there are significant variations in how different communities implement coroutines.[2] Several of these variations are not equivalent, in the sense that a programmer cannot implement one on top of the other.

Coroutines in Lua are like cooperative multithreading and have the following distinguishing properties:

First-class values. Lua programmers can create coroutines anywhere, store them in variables, pass them as parameters, and return them as results. More important, they can resume coroutines anywhere;

Suspend execution. They can suspend their execution from within nested functions. Each coroutine has its own call stack, with a semantics similar to collaborative multithreading. The entire stack is preserved when the coroutine yields;

Asymmetric. Symmetric coroutines offer a single control-transfer operation that transfers control from the running coroutine to another given coroutine. Asymmetric coroutines, on the other hand, offer two control-transfer operations, `resume` and `yield`, that work like a call–return pair; and

Equivalent to one-shot continuations.[2] Despite this equivalence, coroutines offer one-shot continuations in a format that is more natural for a procedural language due to its similarity to multithreading.

Figure 7 illustrates the life cycle of a coroutine in Lua. The program prints 10, 20, 30, and 40, in that order. It starts by creating a coroutine `co`, giving an anonymous function as its body. That operation returns only a handle to the new coroutine, without running it. The program then resumes the coroutine for the first time, starting the execution of its body. The parameter `x` receives the argument given to `resume`, and the program prints 10. The coroutine then yields, causing the call to `resume` to return the value 20, the argument given to `yield`. The program then resumes the coroutine again, making `yield` return 30, the value given to `resume`. The coroutine then prints 30 and finishes, causing the corresponding call to `resume` to return 40, the value returned by the coroutine.

In the case of modules, tables provide name spaces, lexical scoping provides encapsulation, and first-class functions allow exportation of functions.

Coroutines are not as widely used in Lua as tables and functions. Nevertheless, when required, coroutines play a pivotal role, due to their capacity for turning the control flow of a program inside out.

An important use of coroutines in Lua is for implementing cooperative multithreading. Games typically exploit this feature, because they need to be in control to remain responsive at interactive rates. Each character or object in a game has its own script running in a separate coroutine. Each script is typically a loop that, at each iteration, updates the character's state and then yields. A simple scheduler resumes all live coroutines at each game update.

Another use of coroutines is in tackling the "who-is-the-boss" problem. A typical issue with scripting languages is the decision whether to embed or to extend. When programmers embed a scripting language, the host is the boss, that is, the host program, written in the foreign language, has the main loop of the program and calls functions written in the scripting language for particular tasks. When programmers extend a scripting language, the script is the boss; programmers then write libraries for it in the foreign language, and the main loop of the program is in the script.

Embedding and extending both have advantages and disadvantages, and the Lua–C API supports them equally. However, external code can be less forgiving. Suppose a large, monolithic application contains some useful functionality for a particular script. The programmer wants to write the script as the boss, calling functions from that external application. However, the application itself assumes *it* is the boss. Moreover, it may be difficult to break the application into individual functions and offer them as a coherent library to the script.

Coroutines offer a simpler design. The programmer modifies the application to create a coroutine with the script when it starts; every time the application needs an input, it resumes that coroutine. That is the only change the programmer needs to make in the application. The script, for its part, also looks like a regular program, except it yields when it needs to send a command to the application. The control flow of the resulting program progresses as follows: The application starts, creates the coroutine, does its own initialization, and then waits for input by resuming the coroutine. The coroutine then starts running, does its own initialization, and performs its duties until it needs some service from the application. At this point, the script yields with a request, the call to `resume` made by the application returns, and the application services the given request. The application then waits for the next request by resuming the script again.

Presentation of coroutines in the C API is clearly more challenging than presentation of functions and tables. C code can create and resume coroutines without restrictions. In particular, resuming works like a regular function call: It (re) activates the given coroutine when called and returns when the coroutine yields or ends. However, yielding also poses a problem. Once a C function yields, there is no way to later return the control to that point in the function. The API offers two ways to circumvent this restriction: The first is to yield in a tail position: When the coroutine resumes, it goes straight to the calling Lua function. The second is to provide a continuation function when yielding. In this way, when the coroutine resumes, the control goes to the continuation function, which can finish the task of the original function.

We can see again in the API the advantages of asymmetric coroutines for a language like Lua. With symmetric coroutines, all transfers would have the problems that asymmetric coroutines have only when yielding. In our experience, resumes from C are much more common than yields.

Conclusion

Every design involves balancing conflicting goals. To address the conflicts, designers need to prioritize their goals. This is clearly true of the design of any programming language.

Lua has a unique set of design goals that prioritize simplicity, portability, and embedding. The Lua core is based on three well-known, proven concepts—associative arrays, first-class functions, and coroutines—all implemented with no artificial restrictions. On top of these components, Lua follows the motto "mechanisms instead of policies," meaning Lua's design aims to offer basic mechanisms to allow programmers to implement more complex features. For instance, in the case of modules, tables provide name spaces, lexical scoping provides encapsulation, and first-class functions allow exportation of functions. On top of that, Lua adds only the function `require` to search for and `load` modules.

Modularity in language design is nothing new.[11] For instance, it can be used to clarify the construction of a large application.[1] However, Lua uses modularity to keep its size small, breaking down complex constructions into existing mechanisms.

The motto "mechanisms instead of policies" also makes for a flexible language, sometimes too flexible. For instance, the do-it-yourself approach to classes and objects leads to proliferation of different, often incompatible, systems, but is handy when a programmer needs to adapt Lua to the class model of the host program.

Tables, functions, and coroutines as used in Lua have shown great flexibility over the years. Despite the language's continuing evolution, there has been little demand from programmers to change the basic mechanisms.

The lack of built-in complex constructions and minimalist standard libraries (for portability and small size) make Lua a language that is not as good as other scripting languages for writing "quick-and-dirty" programs. Many programs in Lua need an initial phase for programmers to set up the language, as a minimal infrastructure for object-oriented programming. More often than not, Lua is embedded in a host application. Embedding demands planning and the set-up of the language is typically integrated with its embedding. Lua's economy of concepts demands from programmers a deeper understanding of what they are doing, as most constructions are explicit in the code. This explicitness also allows such deeper understanding. We trust this is a blessing, not a curse.

References

1. Cazzola, W. and Olivares, D.M. Gradually learning programming supported by a growable programming language. *IEEE Transactions on Emerging Topics in Computing 4*, 3 (July 2016), 404–415.
2. de Moura, A.L and Ierusalimschy, R. Revisiting coroutines. *ACM Transactions on Programming Languages and Systems 31*, 2 (Feb. 2009), 6.1–6.31.
3. Gamasutra. *Game Developer* magazine's 2011 Front Line Award, Jan. 13, 2012; https://www.gamasutra.com/view/news/129084/
4. Hayes, B. Ephemerons: A new finalization mechanism. In *Proceedings of the 12th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications* (Atlanta, GA, Oct. 5–9). ACM, New York, 1997, 176–183.
5. Ierusalimschy, R. Programming with multiple paradigms in Lua. In *Proceedings of the 18th International Workshop on Functional and (Constraint) Logic Programming, LNCS, Volume 5979*. S. Escobar, Ed. (Brasilia, Brazil, June 28). Springer, Heidelberg, Germany, 2009, 5–13.
6. Ierusalimschy, R., de Figueiredo, L.H., and Celes, W. Lua—An extensible extension language. *Software: Practice and Experience 26*, 6 (June 1996), 635–652.
7. Ierusalimschy, R., de Figueiredo, L.H., and Celes, W. The evolution of Lua. In *Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages* (San Diego, CA, June 9–10). ACM Press, New York, 2007, 2.1–2.26.
8. Ierusalimschy, R., de Figueiredo, L.H., and Celes, W. Passing a language through the eye of a needle. *Commun. ACM 54*, 7 (July 2011), 38–43.
9. International Organization for Standardization. ISO 2000. *International Standard: Programming Languages, C*. ISO/IEC9899: 1999(E).
10. Jones, R., Hosking, A., and Moss, E. *The Garbage Collection Handbook*. CRC Press, Boca Raton, FL, 2011.
11. Kats, L. and Visser, E. The Spoofax Language Workbench: Rules for declarative specification of languages and IDEs. In *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications* (Reno/Tahoe, NV, Oct. 17–21). ACM Press, New York, 2010, 444–463.
12. The Python Software Foundation. *The Python Language Reference, 3.5 Edition*. The Python Software Foundation, 2015.
13. Sestoft, P. *Programming Language Concepts, Second Edition*. Springer, Cham, Switzerland, 2017.
14. Wikipedia. List of applications using Lua; https://en.wikipedia.org/w/index.php?title=List_of_applications_using_Lua&oldid=795421653

Roberto Ierusalimschy (roberto@inf.puc-rio.br) is an associate professor of computer science at PUC-Rio, the Pontifical Catholic University of Rio de Janeiro, Brazil.

Luiz Henrique de Figueiredo (lhf@impa.br) is a researcher at IMPA, the Institute for Pure and Applied Mathematics in Rio de Janeiro, Brazil.

Waldemar Celes (celes@inf.puc-rio.br) is an associate professor of computer science at PUC-Rio, the Pontifical Catholic University of Rio de Janeiro, Brazil.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/a-look-at-the-design-of-lua

DOI:10.1145/3186278

Systematic use of proven debugging approaches and tools lets programmers address even apparently intractable bugs.

BY DIOMIDIS SPINELLIS

Modern Debugging: The Art of Finding a Needle in a Haystack

THE COMPUTING PIONEER Maurice Wilkes famously described his 1949 encounter with debugging like this: "As soon as we started programming, [...] we found to our surprise that it wasn't as easy to get programs right as we had thought it would be. [...] Debugging had to be discovered. I can remember the exact instant [...] when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."[37]

Seven decades later, modern computers are approximately one million times faster and also have one million times more memory than Wilkes's Electronic Delay Storage Automatic Calculator, or EDSAC, an early stored-program computer using mercury delay lines. However, in terms of bugs and debugging not much has changed. As developers, we still regularly make mistakes in our programs and spend a large part of our development effort trying to fix them.

Moreover, nowadays, failures can occur nondeterministically in nanosecond time spans within computer systems consisting of thousands of processors spanning the entire planet running software code where size is measured in millions of lines. Failures can also be frighteningly expensive, costing human lives, bringing down entire industries, and destroying valuable property.[22] Thankfully, debugging technology has advanced over the years, allowing software developers to pinpoint and fix faults in ever more complex systems.

One may reasonably wonder how debugging is actually performed in practice. Three recent publications have shed light on a picture full of contrasts. A common theme is that the practice and problems of debugging have not markedly changed over the past 20 years. Michael Perscheid and colleagues at the SAP Innovation Center and the Hasso Plattner Institute in Potsdam, Germany, examined the debugging practices of professional software developers and complemented the results with an online study.[27] They found developers are not trained in debugging, spend 20% to 40% of their work time in it, structure their debugging process following a simplified scientific method (see Figure 1), are proficient in using symbolic debuggers, regularly debug by adding print statements, are unfamiliar with back-

» key insights

- Targeted software-development process improvements can aid debugging even in cases where their wholesale adoption is impractical.
- Debugging benefits from the widespread availability of code, data, and Q&A forums, and programmers can fix many tricky bugs through the generation and analysis of rich datasets.
- Modern debugging tools offer powerful and specialized facilities that can save hours of tedious unproductive debugging work.

IMAGERY BY BOBNEVV AND ART-SONIK

in-time debuggers and automatic fault localization, and consider design, concurrency, and memory faults as the most difficult to debug. The low level of knowledge and use associated with many advanced debugging techniques was also revealed in a mixed-methods study conducted by Moritz Beller and colleagues at the Delft University of Technology, the Netherlands, and myself.[3] In addition, a team led by Marcel Böhme of Monash University, Clayton, VIC, Australia, performed a controlled study by having software professionals fix faults in a carefully constructed benchmark suite of software faults,[6] finding that professionals typically agree on fault locations they identified using trace-based and interactive debugging. However, the study's subjects then went on to implement incorrect fixes, suggesting opportunities for automated regression testing.

Beginners sometimes view debugging as an opaque process of randomly trying things until locating a fault, a method closer to alchemy than to science. Yet debugging can be systematized into the process illustrated by the Unified Modeling Language activity diagram in Figure 1. The first step involves reliably reproducing the failure. It is up to the programmer to produce meaningful results when running experiments to find the failure's cause. Then comes the task of simplifying the failure's configuration into the smallest test case that would still cause the failure to occur.[40] The small test case simplifies and speeds up the programmer's subsequent fault-discovery work. The corresponding steps are outlined in the top part of Figure 1. The next steps, termed the "scientific method of debugging,"[40] are outlined in the bottom part of the figure. In them, the programmer develops a theory about a fault being witnessed, forms a hypothesis regarding the theory's effects, and gathers and tests data against the hypothesis.[24] The programmer repeatedly refines and tests the theory until the cause of the failure is found.

The programmer may sometimes short-circuit this process by guessing directly a minimal test case or the failure's cause. This is fine, especially if the programmer's intuition as an expert provides correct guidance to the cause. However, when the going gets tough, the programmer should humbly fall back on the systematic process instead of randomly poking the software trying to pinpoint the fault through sheer luck.

The goal of this article is to arm software developers with both knowledge-gathering and theory-testing methods, practices, tools, and techniques that give them a fighting chance when struggling to find the fault that caused a failure. Some techniques (such as examining a memory image, still often termed a magnetic memory "core dump") have been with programmers since the dawn of computing. Others (such as reverse debugging) are only now becoming routinely available. And yet others (such as automatic fault localization based on slicing or statistical analysis) do not seem to have caught on.[27] I hope that summarizing here the ones I find through my experience as most effective can improve any programmer's debugging performance.

On the Shoulders of Colleagues

The productivity boost I get as a developer by using the Web is such that I now rarely write code when I lack Internet access. In debugging, the most useful sources of help are Web search, specialized Q&A sites, and source-code repositories. Keep in mind that the terms of a programmer's work contract might prohibit some of these help options.

Web search. Looking for answers on the Web might sound like cheating. But when debugging, the programmer's goal is to solve a problem, not demonstrate academic knowledge and problem-solving skills. If the fastest way to pinpoint and fix a problem is to copy-and-paste an error message in a Web search engine and select the most promising answer, that is what the programmer should do. One can often obtain better results by polishing the query, removing context-dependent data (such as variable or file names) and enclosing the error message in quotes to search for the exact phrase, rather than just the words in it.

When the going gets tough, the programmer should humbly fall back on the systematic process instead of randomly poking the software trying to pinpoint the fault through sheer luck.

Web search typically works when a programmer encounters problems with widely used third-party software. Two possible reasons can yield an unproductive search: First, the programmer may be the first person ever to encounter the problem. This is unlikely with popular software but can happen when working with a cutting-edge release or with a niche or legacy product. There is always an unlucky soul who is first to post about a failure. Second, the error message the programmer is looking for may be a red herring, as with, say, a standard innocuous warning rather than the actual cause of the failure. One must judge search results accordingly.

Q&A sites. The Web can also help a programmer's debugging through Q&A forums (such as a specific product's issue tracker, a company's internal equivalent, or the various https://stackexchange.com/ sites). If the problem is general enough, it is quite likely an expert volunteer will quickly answer the question. Such forums should be used with courtesy and consideration: One should avoid asking an already-answered question, post to the correct forum, employ appropriate tags, ask using a working minimal example, identify a correct answer, and give back to the community by contributing answers to other questions. Writing a good question post sometimes requires significant research.[20] Then again, I often find this process leads me to solve the problem on my own.

Source-code availability. When the fault occurs within open source software, a programmer can use the Web to find, download, and inspect the corresponding code.[31] One should not be intimidated by the code's size or one's personal unfamiliarity. Chances are, the programmer will be looking at only a tiny part of the code around an error message or the location of a crash. The programmer can find the error message by searching through the code for the corresponding string. Crashes typically offer a stack trace that tells the programmer exactly the associated file and line number. The programmer can thus isolate the suspect code and look for clues that will help isolate the flaw. Is some part of the software misconfigured? Are wrong parameters being passed through an API? Is an object in an incorrect state for the method being called? Or is there perhaps an actual fault in the third-party software?

If the bug fix involves modifying open source software, the programmer should consider contributing the fix back to its developers. Apart from being a good citizen, sharing it will prevent the problem from resurfacing when the software is inevitably upgraded to a newer release.

Tuning the Software-Development Process

Some elements of a team's software-development process can be instrumental in preventing and pinpointing bugs. Those I find particularly effective include implementing unit tests, adopting static and dynamic analysis, and setting up continuous integration to tie all these aspects of software development together. Strictly speaking, these techniques aim for bug detection rather than debugging or preventing bugs before they occur, rather than the location of a failure's root cause. However, in many difficult cases (such as nondeterministic failures and memory corruption), a programmer can apply them as an aid for locating a specific bug. Even if an organization's software development process does not follow these guidelines, they can be adopted progressively as the programmer hunts bugs.

Unit tests. It is impossible to build a bug-free system using faulty software components and devilishly difficult to isolate a problem in a huge lump of code. Unit tests, which verify the functionality of (typically small) code elements in isolation, help in both directions,[28] increasing the reliability of routines (functions or methods) they test by guarding their correctness. In addition, when a problem does occur, the programmer can often try to guess what parts may be responsible for it and add unit tests that are likely to uncover it. This way of working gives the programmer a systematic approach for clearing suspect code until hitting the faulty one. The new unit tests the programmer adds also result in a better-tested system, making refactorings and other changes less risky.

When writing unit tests the programmer is forced to write code that is easy to test, modular, and relatively free of side effects. This can further simplify debugging, allowing the programmer to inspect through the debugger how each small unit behaves at runtime, either by adding suitable breakpoints or by directly invoking the code through the debugger's read-eval-print loop, or REPL, facility.

Debugging libraries and settings. Third-party libraries and systems can also aid a fault-finding mission through the debugging facilities they provide. Some runtime libraries and compilers (such as those of C and C++) provide settings that guard against pointer errors, memory buffer overflows, or memory leaks at the expense of lower runtime performance. Compilers typically offer options to build code for debugging by disabling optimizations (aggressive optimizations can confuse programmers when trying to follow the flow of control and data) and by including more information regarding the source code associated with the compiled code. By enabling these settings the programmer is better able to catch many errors.

Static analysis. One can catch some errors before the program begins to execute by reasoning about the program code through what is termed in the software engineering literature "static program analysis." For example, if a method can return a `null` value and this value is subsequently dereferenced without an appropriate check, a static-analysis tool can determine the program could crash due to a `null` pointer dereference. Tools (such as FindBugs[1] and Coverity Scan[5]) perform this feat through multiple approaches (such as heuristics, dataflow or constraint analysis, abstract interpretation, symbolic execution,[8] and type and effect systems).[23] The end result is a list of messages indicating the location of probable faults in a particular program. Depending on the tool, the approach being used, and the program's language, the list may be incomplete—false negative results—or include entries that do not signify errors—false positives. Nevertheless, finding and fixing such errors often prevents serious faults and can sometimes allow the programmer to find a failure's cause.

Dynamic analysis. An alternative approach for analyzing a program's dynamic behavior is to run it under a specialized tool. This is particularly useful when locating a fault involves sophisticated analysis of large and complex data structures that cannot be easily processed with general-purpose command line tools or a small script. Here are some examples of tools a programmer may find useful. In languages compiled with the LLVM Clang front-end the programmer can use AddressSanitizer,[30] while a program runs, to detect many memory-handling errors: out-of-bounds access, use after free, use after scope exit, and double or invalid frees. Another related tool is Valgrind[21] through which one can find potentially unsafe uses of uninitialized values and memory leaks. In addition, Valgrind's Helgrind and data race detector (DRD) tools can help find race conditions and lock order violations in code that uses the POSIX threads API. If the code is using a different thread API, the programmer should consider applying Intel Inspector technology,[29] which also supports Threading Building Blocks, OpenMP, and Windows threads.

Continuous integration. Running static program-analysis tools on code to pinpoint a fault can be like trying to turn the Titanic around after hitting an iceberg. At that point, catastrophic damage has already been done, and it is too late to change the course of events. In the case of a large software codebase, trying to evaluate and fix the scores of error messages spewed by an initial run of a static-analysis tool can be a thorny problem. The developers who wrote the code may be unavailable to judge the validity of the errors, and attempting to fix them might reduce the code's maintainability and introduce even more serious faults. Also, the noise of existing errors hides new ones appearing in fresh code, thus contributing to a software-quality death spiral. To avoid such a problem, the best approach is to integrate execution of static analysis into the software's continuous-integration process,[10] which entails regularly merging (typically several times a day) all developer work into a shared reference version. Running static analysis when each change is made can keep the software codebase squeaky clean from day one.

Making the Software Easier to Debug

Some simple software design and programming practices can make software easier to debug, by providing or configuring debugging functionality, logging and receiving debug data, and using high(er)-level languages. Again, a programmer can selectively adopt these practices during challenging bug-hunting expeditions.

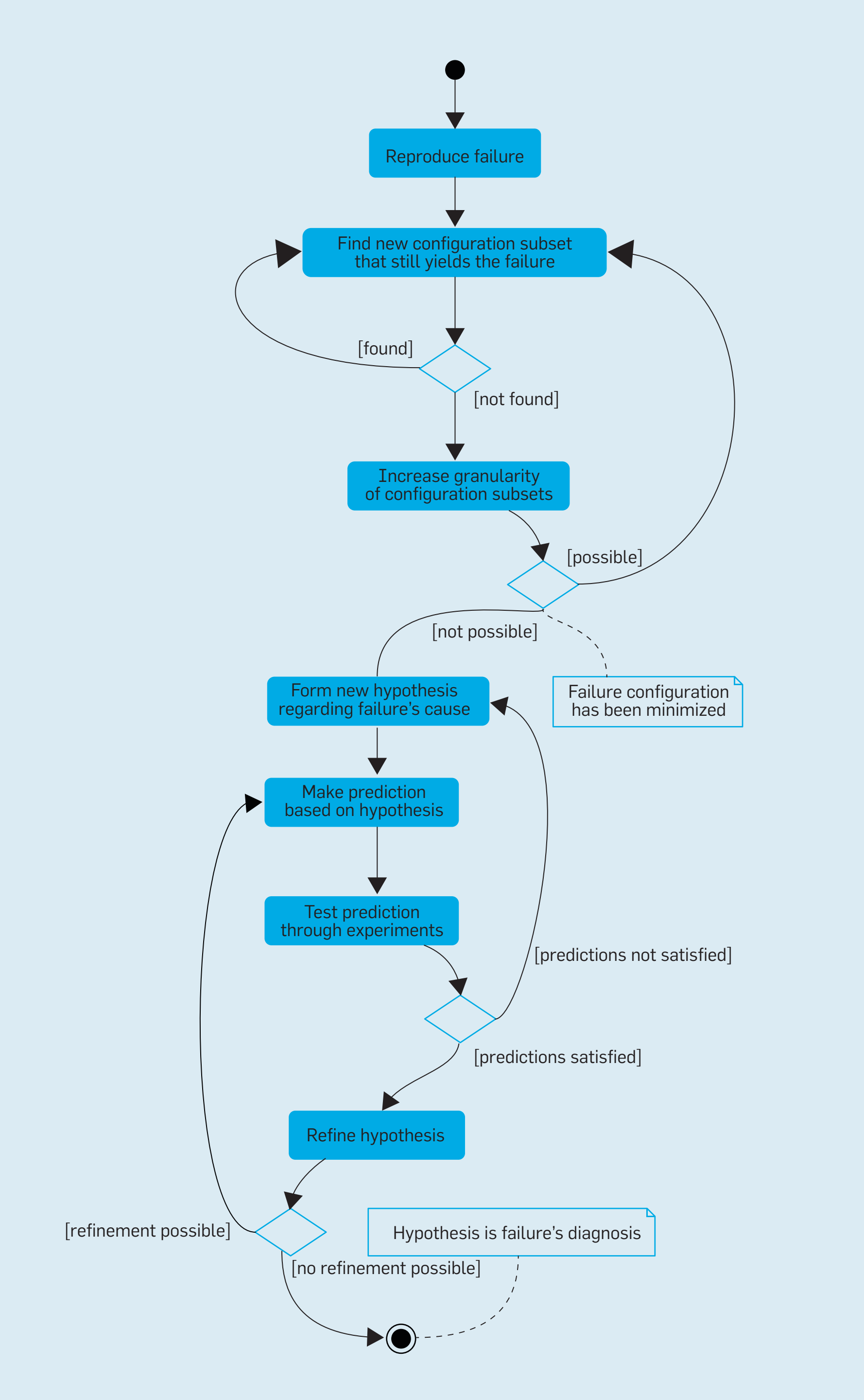
Software's debugging facilities. A helpful way to isolate failures is to build and use debugging facilities within the software. The aim here is to make the software's operation more predictable and transparent. For example, some programs that execute in the background (such as Unix daemons or Windows services) offer a debugging option that causes them to operate synchronously as a typical command-line program, outputting debugging messages on their standard output channel. This is the approach I always use to debug secure shell connection problems. Other debugging settings may make the software's operation more deterministic, which is always helpful when trying to isolate a fault through repeated executions. Such changes may include elimination of multiple threads, use of a fixed seed in random-number generators, and restriction of buffers and caches to a small size in order to increase the likelihood of triggering overflows and cache misses. Software developers should consider adding such facilities to the code they debug. They should, however, keep in mind that some debugging facilities can lead to security vulnerabilities. To avoid this risk, the programmer must ensure the facilities are automatically removed, disabled, or made obnoxiously conspicuous in production builds.

Logging. Programmers are often able to pinpoint faults by adding or using software-logging or -tracing statements.[33] In their simplest form they are plain print commands outputting details regarding the location and state of a program's execution. Unlike watchpoints added in a debugging session, the statements are maintained with the program and are easily tailored to display complex data structures in a readable format.

Modern software tracing is typically performed through a logging framework (such as Apache log4j and Apache log4net) that provides a unified API for capturing, formatting, and handling a program's logging output. Such a framework also allows programmers to tailor at runtime the program's output verbosity and corresponding performance and storage cost. Programmers typically minimize logging to that required for operational purposes when a program executes in a production environment but increase it to include detailed software tracing when they want to debug a failure.

Telemetry. An obvious extension of logging facilities is telemetry, or the

Figure 1. A process for systematic debugging.


Reproduce failure
Find new configuration subset that still yields the failure
[found]
[not found]
Increase granularity of configuration subsets
[possible]
[not possible]
Failure configuration has been minimized
Form new hypothesis regarding failure's cause
Make prediction based on hypothesis
Test prediction through experiments
[predictions not satisfied]
[predictions satisfied]
Refine hypothesis
[refinement possible]
Hypothesis is failure's diagnosis
[no refinement possible]

ability to obtain debugging data from remote program executions (such as those by the program's end users). Ideally, a programmer would want to be able to obtain the following types of data: First, data associated with the execution context (such as the version of the program, helper code, and operating system); values of environment variables; and contents of configuration files. Then comes data about the program's operational status (such as commands executed, settings, and data files). And finally, in cases of program crashes a programmer also needs details regarding the location of the program crash (such as method name, line number, and program counter), runtime context (such as call stack, values of parameters, local variables, and registers), and the reason behind the crash (such as uncaught exception or illegal memory access).

Setting up a telemetry facility before the software is distributed can be a lifesaver when a nasty bug surfaces. On some platforms, a third-party library can be used to collect the required data, outsource its collection, and access the results through a Web dashboard. Keep in mind that telemetry records often contain personal data. When raw memory is recorded, confidential data the reporting code did not gather explicitly (such as passwords and keys) can end up in the telemetry database. Software development teams should consider (carefully) what data to collect, how long to store it, and how they will protect it, and disclose these details to the user.

High-level languages. Another last-resort approach a programmer may have to turn to when debugging complex algorithms, data structures, or protocols may be the implementation of the code in a higher-level formalism. This approach is useful when the programming language developers are working with blurs their focus on the problem's essence. Verbose type declarations, framework boilerplate, unsafe pointers, obtuse data types, or spartan libraries may prevent one from expressing and fixing the parts that matter, burying the programmer instead in a tar-pit of tangential goo. Lifting the code's level of abstraction may simplify finding whether a knotty failure stems from errors in the logic or in the implementation.

My favorite source of intelligence regarding a system's operation is the calls it makes to the operating system.

Suitable technologies for this approach may be scripting languages (such as Python and R), domain-specific languages,[19] or model-driven software development.[35] Adopting high-level formalisms that allow for symbolic reasoning lets the programmer kill two birds with one stone: fix the bug the programmer is after and provide (qualified) guarantees that no other bugs exist in the part of the code the programmer has analytically reasoned about regarding its correctness. Once the alternative implementation is working, the programmer can decide whether architectural, operational, and performance considerations should allow keeping the code in its new formalism, whether to rewrite it (carefully), or automatically transform it to its original programming language.

Insights from Data Analytics

Data is the lifeblood of debugging. The more data that is associated with a failure, the easier it is to find the corresponding fault. Fortunately, nowadays, practically limitless secondary storage, ample main memory, fast processors, and broadband end-to-end network connections make it easy to collect and process large volumes of debugging data. The data can come from the development process (such as from revision-control systems and integrated development environments, or IDEs), as well as from program profiling. The data can be analyzed with specialized tools, an editor, command-line tools, or small scripts.

Some of the processes described here have been systematized and automated by Andreas Zeller of Saarland University, Germany, under the term "delta debugging"[38] and used to locate cause-effect chains in program states[39] and simplify failure-inducing inputs.[41] Although the corresponding tools were mostly research prototypes, the same ideas can be applied on an ad hoc basis to improve the effectiveness of debugging tasks. Consider these representative examples:

Revision-control data. Bugs often occur as the software evolves. By keeping software under version control, using configuration management tools (such as Git, Mercurial, and Subversion), the programmer can dig into a project's history to aid debugging work. Here are some examples: If a program crashes or misbehaves at a particular program line the programmer can analyze the source code to see the last change associated with that line (for example, with the `git blame` command). A review of the change can then reveal that, say, one's colleague who implemented it forgot to handle a specific case. Alternatively, by reading the version-control log of software changes, the programmer can find a recent change that may be related to the failure being witnessed and examine it in detail.

Another neat use of the version-control system for debugging is to automatically find the change that introduced a fault. Under Git the programmer constructs a test case that causes the fault and then specifies it to the `git bisect` command together with a window of software versions where the fault probably appeared. The command will then run a binary search among all the versions within the window in order to determine the exact change that triggered the failure.

Differential debugging. Differences between datasets can also reveal a fault when the programmer can lay hands on a working system and a failing one.[34] The goal is to find where and why the operation of the two systems diverges. The data that can be used for this purpose can come from their generated log files, their execution environment, or traces of their operation. In all cases the programmer must ensure the two system configurations are as similar as possible, apart from exhibiting the failure.

Examining log files for differences can be easily performed by configuring the most detailed logging possible, collecting the logs, removing nonessential differences (or keeping only the pertinent records), and comparing them.

Looking for differences in the environment in which the two systems operate involves examining a program's user input, command-line arguments, environment variables, accessed files (including configuration, executables, and libraries), and associated services.

Investigating differences in the operation of the two systems is more difficult, but thankfully many tools can help. My favorite source of intelligence regarding a system's operation is the calls it makes to the operating system. They (and their results) often determine to a very large extent a program's behavior. Consequently, any divergence between operating system calls is a valuable hint regarding the fault the programmer is trying to locate. For example, the programmer may see that one program tries to open a file that does not exist, times out on a network connection, or runs out of memory. To trace system calls, the programmer can use the strace, ktrace, or truss tools under Unix and similar systems and Procmon[18] under Windows.

The interactions the programmer wants to investigate may also occur at other levels of a system's stack. The programmer can trace calls to dynamically linked libraries with ltrace[7] (Unix) and Procmon (Windows). The programmer can also untangle interactions with services residing on other hosts by examining network packets through Wireshark's[25] nifty GUI. Keep in mind that most relational database systems provide a way to keep and examine a log of executed SQL statements. The programmer can sometimes better understand a program's behavior by obtaining a snapshot of its open files and network connections. Tools that provide this information include lsof (Unix), netstat (Unix and Windows), and tcpvview (Windows). Finally, two tools, DTrace[14] and SystemTap[11] allow the programmer to trace a system's operation across the entire software stack. They should be used if available.

A programmer can also investigate a failing program's trace log without using a working program's log as a reference. However, such an investigation typically requires a deeper understanding of the program's operation, the ability to pinpoint the pertinent log parts, and access to the source code in order to decipher the trace being read. Things to look for in such cases are failing system calls, library calls that return with an error, network timeouts, and empty query result sets.

The log files of the failing system and the working system often differ in subtle ways that hinder their automatic comparison; for example, they may include different timestamps, process identifiers, or host names. The solution is thus to remove the unessential differing fields. The programmer can do that with the editor, Unix filter tools, or a small script. And then apply one of several file-differencing tools to find where the two log files diverge.

Editor tricks. A powerful text editor or IDE can be a great aid when analyzing log data. Syntax coloring can help the programmer identify the relevant parts. With rectangular selections and regular expressions one can eliminate boilerplate or nonessential columns to focus on the essential elements or run a file-difference program on them. The programmer can also identify patterns associated with a bug using search expressions and matched-text highlighting. Finally, by displaying multiple buffers or windows the programmer can visually inspect the details of different runs.

Command-line tools. The programmer can also perform and, more important, automate many of these tasks and much more with Unix-derived command-line filter tools[15,32] available natively or as add-ons on most platforms (such as GNU/Linux, Windows using Cygwin, and macOS). The programmer can easily combine them to perform any imaginable debugging analysis task. This is important, as effective debugging often requires developing and running ad hoc processing tasks.

Here are several examples of how typical Unix command-line tools can be used in a debugging session: The programmer fetches data from the file system using the `find` command from webpages and services using `curl` and from compiled files (depending on the platform) by running `nm`, `javap`, or `dumpbin`. The programmer can then select lines that match a pattern with `grep`, extract fields with `cut`, massage the content of lines with `sed`, and perform sophisticated selections and summarize with `awk`. With normalized datasets at hand, the programmer can then employ `sort` and `uniq` to create ordered sets and count occurrences, `comm` and `join` to find set differences and join sets together, and `diff` to look at differences. Lastly, the number of results can be summarized with `wc`, the first or last records can be obtained with `head` or `tail`, and a hash for further processing derived through `md5sum`. For tasks that are performed often, it is a good practice to package the invocation of the corresponding commands into a Unix shell script and distribute the script as part of the code's software-developer tools.

As a concrete case, consider the task of locating a resource leak in the code. A simple heuristic could involve looking for mismatches between the number of calls to `obtainResource` and calls to `releaseResource`; see Figure 2 for a small Bash script that performs this task. The script uses the `grep` and `sort` commands to create two ordered sets: one with the number of calls to `obtainResource` in each file and another with the corresponding number of calls to `releaseResource`. It then provides the two sets to the `comm` command that will display the cases where records in the two sets do not match.

Scripting languages. If the editor cannot handle the required debugging analysis and the programmer is daunted by the Unix command-line interface, one can also analyze data with a scripting language (such as Python, Ruby, or Perl). Typical tasks that need to be mastered in order to analyze debugging data include sequential reading of text records from a file, splitting text lines on some delimiter, extracting data through regular expression matching, storing data in associative arrays, and iterating through arrays to summarize the results. An advantage of such scripts over other data-analysis approaches is the programmer can easier integrate them within composite project workflows that may involve sending email messages to developers or updating databases and dashboards.

Profiling. When debugging performance issues, the tools the programmer can use various profilers to debug individual processes. The simplest work through sampling, interrupting the program's behavior periodically and giving a rough indication regarding the routines where the program spends most of its time. The programmer thus identifies the routines on which to concentrate optimization efforts. One notch more advanced is graph-based profilers[13] that intercept each routine's entry and exit in order to provide precise details not only of a routine's contribution to the software's overall CPU use but also how the cost is distributed among the routine's callers. An added complication of performance debugging in modern systems is that machine instructions can vary their execution time by at least an order of magnitude based on context. To get to the bottom of such problems the programmer needs to obtain details of low-level hardware interactions (such as cache misses and incorrect jump predictions) through tools that use the CPU's performance counters. These counters tally CPU events associated with performance and expose them to third-party tools (such as the Concurrency Visualizer extension for Visual Studio, Intel's VTune Performance Analyzer, and the Linux `perf` command). For example, performance counters can allow the programmer to detect performance issues associated with false sharing among threads.

Getting More from a Debugger

Given the propensity of software to attract and generate bugs, it is hardly surprising that the capabilities of debuggers are constantly evolving:

Data breakpoints. One impressive facility in many modern debuggers is the ability to break a program's operation when a given value changes, the so-called "data breakpoints." Unlike code or control breakpoints that are easily implemented by patching the code location where a breakpoint is inserted, data breakpoints are difficult to implement efficiently because the corresponding value needs to be checked after each CPU instruction. A debugger could check the value in software by single-stepping through the program's instructions but would reduce its execution speed to intolerable levels.

Many current CPUs instead offer the ability to perform this check through their hardware using so-called write monitors. All the debugger has to do is to set special processor registers with the memory address and the length of the memory area the programmer wants to monitor. The processor will then signal the debugger every time the contents of these locations change. Based on this facility, a debugger can also implement a conditional data breakpoint that interrupts the program's execution when a value satisfies a given condition. The computational overhead in this case is a bit greater because the debugger

Figure 2. Ad hoc location of a probable resource leak.

```
# List non-common lines between the two sets
comm -3 <(
  # Counts per file of obtainResource
  grep -rc obtainResource . | sort) <(
  # Counts per file of releaseResource
  grep -rc releaseResource . | sort)
```

Figure 3. Example of a reverse-debugging session.

```
Breakpoint 1, main () at hairy_code.c:1219
1219              read_data();
(gdb) record
(gdb) next
1220              analyze_data();
(gdb) next
hairy_code: Panic!
1221              display_results();
(gdb) reverse-next
1220              analyze_data();
(gdb) step
analyze_data () at hairy_code.c:1209
1209              if (n == 0)
(gdb) step
1210                      warnx("Panic!");
```

has to check the condition on every change but still orders of magnitude lower than the alternative of checking after every instruction.

Data breakpoints are especially useful when the programmer is unfamiliar with the program's operation and wants to pinpoint what statements change a particular value. They also come in handy in languages that lack memory bounds checking (such as C and C++) in order to identify the cause of memory corruption. All a programmer has to do is set a data breakpoint associated with the corrupted element and wait for the data breakpoint to trigger.

Reverse debugging. Another cool feature available through recent advances in hardware capabilities is "reverse debugging,"[12] or the ability to run code in reverse, in effect traveling back in time. When forward-stepping through the code starting from a statement *A*, a programmer finds statements and variables that can be influenced by *A*, called by researchers a "forward slice."[40] When stepping through code in reverse from *A*, a programmer finds statements and values that could have influenced *A*; this backward slice can help the programmer understand how the program ended up in a specific state.

Reverse debugging is implemented through brute-force computation by having the debugger log the effect of each instruction and thereby obtain the data required to undo it. It has become feasible with fast CPUs and abundant main memory. When debugging a single application, not all actions can be undone; once an operating system call has been performed on a program, effects that cross the debugger's event horizon are there to stay. Nevertheless, the capability can be beneficial when debugging algorithmic code. It is most useful in cases where, while searching for the cause of a failure, a programmer might inadvertently step or glance over the culprit statements. At this point, the programmer can rewind the execution to the point before the culprit statements and move forward again more cautiously.

As an example, consider debugging a problem associated with the display of the cryptic message "Panic!," which appears in hundreds of places within the code. At some point the programmer may be going over code like this

```
read _ data();
analyze _ data();
display _ results();
```

Stepping into each routine to see if it prints the message may take ages. Figure 3 shows the log of a gdb debugger session, demonstrating how a programmer can find the location through reverse debugging; the source code line listed before each `gdb` prompt is the one to be executed next. Graphical interfaces to similar functionality are also available through commercial offerings (such as Microsoft's IntelliTrace for the .NET platform and the Chronon Time Travelling Debugger for the Java ecosystem).

The programmer first sets up gdb for reverse debugging by issuing the `record` command, then runs each function but steps over its innards with the `next` command. Once the programmer sees the "Panic!" message, which is emitted by the `analyze data` routine, the programmer issues the `reverse-next` command to undo the previous `next` and move the execution context again just before the call to the `analyze.data` routine. This time the programmer issues the `step` command to step *into* the routine and find why the message appeared.

Capture and replicate. With multicore processors found in even low-end smartphones today, multithreaded code and the bugs associated with it are a (frustrating) fact of life. Debugging such failures can be difficult because the operation of multithreaded programs is typically nondeterministic; each run of a program executes the threads in a slightly different order that may or may not trigger the bug. I recall the agony of debugging multithreaded rendering code that would occasionally miscalculate just a couple of pixels in a four-million-pixel image. To pinpoint a race condition that exhibits itself in a few nanoseconds within a multi-hour program run, programmers need all the help they can get from powerful software.

Tools helpful in such cases are often those able to capture and replicate in full detail a program's memory-access operations[16] (such as the PinPlay/DrDebug Program Record/Replay Toolkit,[26] which can be used with Eclipse or gdb, and the Cronon recorder and debugger, which works with Java applications). The way programmers work with these tools is to run the application under their control until the failure emerges. This run will generate a recording of the session that can then be replayed under a specialized debugger to locate the statement that causes the failure. With the statement in hand the programmer can look at the program state to see what caused the particular statement to execute or why its execution was not prevented through a suitable lock. These tools thus transform a fleeting nondeterministic failure into a stable one that can be targeted and debugged with ease.

Running and dead processes. Two time-honored but still very useful things programmers can do with a debugger is to debug processes that are already running and processes that have crashed. Debugging a running process is the way to go if it is misbehaving with a failure that is difficult to reproduce. In this case, a programmer would use the operating system's process-display command (such as `ps` under Unix and TaskManager under Windows) to find the numerical identifier of the offending process. The programmer can then fire the debugger, instructing it to debug the process's executable file but also to attach itself to the running process specified by its identifier. From this point onward, programmers can use the debugger as they normally would:

Interrupt stuck program. A programmer can interrupt a stuck program to see at what point the program entered into an endless loop or issued a nonreturning system call;

Add new breakpoints. A programmer can add new breakpoints to see when and how a program reaches a particular code position; and

Examine values. A programmer can examine the values of variables and display the call stack.

Debugging a crashed process allows programmers to perform a post-mortem examination of the facts related to its demise. Some systems allow programmers to launch a debugger at the moment a process crashes. A more flexible alternative involves obtaining an image of the memory associated with the process, the so-called "core dump" (Unix) or Minidump (Windows). This allows the programmer to obtain the dump from a production environment or a customer site and then dissect it on the development environment. There are various methods for obtaining a process's memory dump. On Unix systems, a programmer typically will configure the operating system core file size limit through the system's shell and then wait for the process to crash or send it a `SIGQUIT` signal. On Windows systems a programmer can use the Procdump[18] program to achieve the same results. In both cases, obtaining a memory dump from a still-running but hung process allows the programmer to debug infinite loops and concurrency deadlocks.[9]

Although a memory dump will not allow a programmer to resurrect and step through the execution of the corresponding process, though it is still useful, because the programmer can examine the sequence of calls that were in effect at the point of the crash, the local variables of each routine in that sequence, and the values of global and heap-allocated objects.

No bug can elude a programmer who perseveres.

Debugging Distributed Systems

Modern computing rarely involves an isolated process running on a system that matches a programmer's particular development environment. In many cases, the programmer is dealing with tens to thousands of processes, often distributed around the world and with diverse hardware ranging from resource-constrained Internet of Things (IoT) devices, to smartphones, to experimental and specialized platforms. While these systems are the fuel powering the modern economy, they also present programmers with special challenges. According to the insightful analysis by Ivan Beschastnikh and colleagues at the University of British Columbia, these are heterogeneity, concurrency, distributing state, and partial failures.[4] Moreover, following my own experience, add the likely occurrence of events that would be very rare on an isolated machine, the difficulty of correlating logs across several hosts,[2] and replicating failures in the programmer's development environment.

Remote debugging. The emergence of cloud computing and the IoT have brought with them the necessity of being able to debug systems remotely. A debugger with a graphical interface is not ideal in such situations because it might not be sufficiently responsive when debugging a cloud application across the planet or when a particular IoT platform may lack the power to run it. Consequently, it may make sense for programmers to acquaint themselves with a debugger's command-line interface, as well as the shell commands required to debug more complex systems. An alternative that may sometimes work is a GUI debugger's ability to communicate with a small remote debugger-monitor program the programmer installs and runs at the remote end.

Monitoring. When debugging distributed systems, monitoring and logging are the name of the game. Monitoring will flash a red light when something goes wrong, giving the team an opportunity to examine and understand why the system is misbehaving and thus help pinpoint the underlying cause. In such cases a programmer is often not debugging the code of individual processes but the architecture, configuration, and deployment of systems that may span an entire datacenter or the entire planet. A team can monitor individual failures and performance trends with systems like Nagios, NetData, Ganglia, and Cacti. An interesting approach for generating and thus being able to debug rare failures in complex distributed systems is to cause controlled component failures through specialized software, an approach pioneered by Netflix through its ChaosMonkey.[36]

Event logging. Given that it is not yet possible for a programmer to single-step concurrently through the multitude of processes that might comprise a modern system, when debugging such failures a programmer must rely on event logging, which involves processes logging operational events that target system administrators and reliability engineers. Unlike the software-tracing statements a programmer may use to pinpoint a failure in an individual process, event logging is always enabled in a production environment. By providing "observability," logging can help operations personnel ascertain an application's health status, view its interactions with other processes, and determine changes in a system's configuration. By listing metrics and error messages, logs can reveal a sickly application (such as one with unusually high latency or memory use) or expose one that fails due to insufficient privileges. Such things can help programmers pinpoint a specific application as a contributing factor in a more complex failure.

Virtualization and system simulators. One family of technologies that can help debug software running on hardware that does not match a given development environment includes virtual machines, emulators, and system simulators. With virtual machines and operating system virtualization systems (such as Docker), software development teams can create a single environment that can be used for development, debugging, and production deployment. Such containers are also useful when a programmer wants to find and eliminate configuration-related errors. Moreover, development environments for some commonly used embedded platforms (such as smartphones) come with an emulator, allowing programmers to experience the capabilities of the target hardware from the comfort of a desktop. Finally, when a team is developing software and hardware together, a full system simulator (such as Simics[17]) will provide a high-fidelity view of the complete platform stack.

Conclusion

The number of possible faults in a software system can easily challenge the limits of human ingenuity. Debugging the corresponding failures thus requires an arsenal of tools, techniques, methods, and strategies. Here I have outlined some I find particularly effective, but there are many others I consider useful, as well as many specialized ones that may work wonders in a particular environment.

Each debugging session represents a new venture into the unknown. Programmers should work systematically, starting with an approach that matches the failure's characteristics, but adapt it quickly as they uncover more things about the failure's probable cause. Programmers should not hesitate to switch from Web searching, to logging, to single-stepping, to constructing a unit test, or a specialized tool. No bug can elude a programmer who perseveres. And keep in mind that the joy of fixing a fault is proportional to the work the programmer puts into debugging the failure.

Acknowledgments

My thanks to Moritz Beller, Alexander Lattas, Dimitris Mitropoulos, Tushar Sharma, and the anonymous reviewers for insightful comments on earlier versions of this article.

References

1. Ayewah, N., Hovemeyer, D., Morgenthaler, J.D., Penix, J., and Pugh, W. Using static analysis to find bugs. *IEEE Software 25*, 5 (Sept. 2008), 22–29.
2. Bailis, P., Alvaro, P., and Gulwani, S. Research for practice: Tracing and debugging distributed systems; programming by examples. *Commun. ACM 60*, 7 (July 2017), 46–49.
3. Beller, M., Spruit, N., Spinellis, D., and Zaidman, A. On the dichotomy of debugging behavior among programmers. In *Proceedings of the 40th International Conference on Software Engineering* (Gothenburg, Sweden, May 27–June 3). ACM Press, New York, 2018, 572–583.
4. Beschastnikh, I., Wang, P., Brun, Y., and Ernst, M.D. Debugging distributed systems. *Commun. ACM 59*, 8 (Aug. 2016), 32–37.
5. Bessey, A., Block, K., Chelf, B., Chou, A., Fulton, B., Hallem, S., Henri-Gros, C., Kamsky, A., McPeak, S., and Engler, D. A few billion lines of code later: Using static analysis to find bugs in the real world. *Commun. ACM 53*, 2 (Feb. 2010), 66–75.
6. Böhme, M., Soremekun, E.O., Chattopadhyay, S., Ugherughe, E., and Zeller, A. Where is the bug and how is it fixed? An experiment with practitioners. In *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering* (Paderborn, Germany, Sept. 4–8). ACM Press, New York, 2017, 117–128.
7. Branco, R.R. Ltrace internals. In *Proceedings of the Linux Symposium*, A.J. Hutton and C.C. Ross, Eds. (Ottawa, ON, Canada, June 27–30, 2007), 41–52; https://www.kernel.org/doc/ols/2007/ols2007v1-pages-41-52.pdf
8. Cadar, C. and Sen, K. Symbolic execution for software testing: Three decades later. *Commun. ACM 56*, 2 (Feb. 2013), 82–90.
9. Cantrill, B. and Bonwick, J. Real-world concurrency. *Commun. ACM 51*, 11 (Nov. 2008), 34–39.
10. Duvall, P.M., Matyas, S., and Glover, A. *Continuous Integration: Improving Software Quality and Reducing Risk.* Pearson Education, Boston, MA, 2007.
11. Eigler, F.C. Problem solving with Systemtap. In *Proceedings of the Linux Symposium*, A. J. Hutton and C. C. Ross, Eds. (Ottawa, ON, Canada, July 19–22, 2006), 261–268; https://www.kernel.org/doc/ols/2006/ols2006v1-pages-261-268.pdf
12. Engblom, J. A review of reverse debugging. In *Proceedings of the 2012 System, Software, SoC and Silicon Debug Conference* (Vienna, Austria, Sept. 19–20). Electronic Chips & Systems Design Initiative, Gières, France, 2012, 28–33.
13. Graham, S.L., Kessler, P.B., and McKusick, M.K. An execution profiler for modular programs. *Software: Practice & Experience 13*, 8 (Aug.1983), 671–685.
14. Gregg, B. and Mauro, J. *DTrace: Dynamic Tracing in Oracle Solaris, Mac OS X, and FreeBSD.* Prentice Hall Professional, Upper Saddle River, NJ, 2011.
15. Kernighan, B.W. Sometimes the old ways are best. *IEEE Software 25*, 6 (Nov. 2008), 18–19.
16. LeBlanc, T.J. and Mellor-Crummey, J.M. Debugging parallel programs with Instant Replay. *IEEE Transactions on Computers C-36*, 4 (Apr. 1987), 471–482.
17. Magnusson, P.S., Christensson, M., Eskilson, J., Forsgren, D., Hallberg, G., Hogberg, J., Larsson, F., Moestedt, A., and Werner, B. Simics: A full system simulation platform. *Computer 35*, 2 (Feb. 2002), 50–58.
18. Margosis, A. and Russinovich, M.E. *Windows Sysinternals Administrator's Reference.* Microsoft Press, Redmond, WA, 2011.
19. Mernik, M., Heering, J., and Sloane, A.M. When and how to develop domain-specific languages. *ACM Computing Surveys 37*, 4 (Dec. 2005), 316–344.
20. Nasehi, S.M., Sillito, J., Maurer, F., and Burns, C. What makes a good code example?: A study of programming Q&A in StackOverflow. In *Proceedings of the 28th IEEE International Conference on Software Maintenance* (Riva del Garda, Trento, Italy, Sept. 23–30). IEEE Press, 2012, 25–34.
21. Nethercote, N. and Seward, J. Valgrind: A framework for heavyweight dynamic binary instrumentation. In *Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation* (San Diego, CA, June 10–13). ACM Press, New York, 2007, 89–100.
22. Neumann, P.G. *Computer Related Risks.* Addison-Wesley, Reading, MA, 1995.
23. Nielson, F., Nielson, H.R., and Hankin, C. *Principles of Program Analysis.* Springer, Berlin, Germany, 2015.
24. O'Dell, D.H. The debugging mind-set. *Commun. ACM 60*, 6 (June 2017), 40–45.
25. Orebaugh, A., Ramirez, G., and Beale, J. *Wireshark & Ethereal Network Protocol Analyzer Toolkit.* Syngress, Cambridge, MA, 2006.
26. Patil, H., Pereira, C., Stallcup, M., Lueck, G., and Cownie, J. Pinplay: A framework for deterministic replay and reproducible analysis of parallel programs. In *Proceedings of the Eighth Annual IEEE/ACM International Symposium on Code Generation and Optimization* (Toronto, ON, Canada, Apr. 24–28). ACM Press, New York, 2010, 2–11.
27. Perscheid, M., Siegmund, B., Taeumel, M., and Hirschfeld, R. Studying the advancement in debugging practice of professional software developers. *Software Quality Journal 25*, 1 (Mar. 2017), 83–110.
28. Runeson, P. A survey of unit-testing practices. *IEEE Software 23*, 4 (July 2006), 22–29.
29. Sack, P., Bliss, B.E., Ma, Z., Petersen, P., and Torrellas, J. Accurate and efficient filtering for the Intel Thread Checker race detector. In *Proceedings of the First Workshop on Architectural and System Support for Improving Software Dependability* (San Jose, CA, Oct. 21–25). ACM Press, New York, 2006, 34–41.
30. Serebryany, K., Bruening, D., Potapenko, A., and Vyukov, D. Address-Sanitizer: A fast address sanity checker. In *Proceedings of the 2012 USENIX Annual Technical Conference* (Boston, MA, June 13–15). USENIX Association, Berkeley, CA, 2012, 309–318.
31. Spinellis, D. *Code Reading: The Open Source Perspective.* Addison-Wesley, Boston, MA, 2003.
32. Spinellis, D. Working with Unix tools. *IEEE Software 22*, 6 (Nov./Dec. 2005), 9–11.
33. Spinellis, D. Debuggers and logging frameworks. *IEEE Software 23*, 3 (May/June 2006), 98–99.
34. Spinellis, D. Differential debugging. *IEEE Software 30*, 5 (Sept./Oct. 2013), 19–21.
35. Stahl, T. and Volter, M. *Model-Driven Software Development: Technology, Engineering, Management.* John Wiley & Sons, Inc., New York, 2006.
36. Tseitlin, A. The anti-fragile organization. *Commun. ACM 56*, 8 (Aug. 2013), 40–44.
37. Wilkes, M. *The Birth and Growth of the Digital Computer.* Lecture delivered at the Digital Computer Museum, available through the Computer History Museum, Catalog Number 102695269, Sept. 1979; https://youtu.be/MZGZfsr1KfY
38. Zeller, A. Automated debugging: Are we close? *Computer 34*, 1 (Nov. 2001), 26–31.
39. Zeller, A. Isolating cause-effect chains from computer programs. In *Proceedings of the 10th ACM SIGSOFT Symposium on Foundations of Software Engineering* (Charleston, SC, Nov. 18–22). ACM Press, New York, 2002, 1–10.
40. Zeller, A. *Why Programs Fail: A Guide to Systematic Debugging, Second Edition.* Morgan Kaufmann, Burlington, MA, 2009.
41. Zeller, A. and Hildebrandt, R. Simplifying and isolating failure-inducing input. *IEEE Transactions on Software Engineering 28*, 2 (Feb. 2002), 183–200.

Diomidis Spinellis (dds@aueb.gr) is a professor in and head of the Department of Management Science and Technology in the Athens University of Economics and Business, Athens, Greece, and author of *Effective Debugging: 66 Specific Ways to Debug Software and Systems*, Addison-Wesley, 2016.

IEEE

IEEE computer society

SERVICES

CALL FOR PAPERS

2019 IEEE World Congress on Services

BigData/CLOUD/EDGE/ICCC/ICIOT/ICWS/SCC

July 8-13, 2019 Milan, Italy

http://conferences.computer.org/services/2019/

2019 IEEE World Congress on Services (SERVICES) will be held on July 8-13, 2019 in Milan, Italy. The Congress is solely sponsored by the IEEE Computer Society under the auspice of the Technical Committee on Services Computing (TCSVC). The scope of the Congress will cover all aspects of services computing and applications, current or emerging. It covers various systems and networking research pertaining to cloud, edge and Internet-of-Things (IoT), as well as technologies for intelligent computing, learning, big data and blockchain applications, while addressing critical issues such as high performance, security, privacy, dependability, trustworthiness, and cost-effectiveness. Authors are invited to prepare early and submit original papers to any of these conferences at *www.easychair.org*. All submitted manuscripts will be peer-reviewed by at least 3 reviewers. Accepted and presented papers will appear in the conference proceedings published by the IEEE Computer Society Press. The Congress will be organized with the following seven affiliated conferences/congresses:

IEEE International Congress on Big Data (BigData Congress)
Big data acquisitions, analyses, storage, and mining for various services and applications

IEEE International Conference on Cloud Computing (CLOUD)
Innovative cloud computing for both high quality infrastructure and mobile services

IEEE International Conference on Edge Computing (EDGE)
High quality services computing between cloud systems and Iot devices

IEEE International Conference on Web Services (ICWS)
Innovative web services for various effective applications

IEEE International Conference on Cognitive Computing (ICCC)
Cognitive computing, learning algorithms for intelligent services and applications

IEEE International Congress on Internet of Things (ICIOT)
Innovative IoT technology for digital world services

IEEE International Conference on Services Computing (SCC)
Intelligent services computing, lifecycles, infrastructure and mobile environments

Key Dates
Early Paper submission due: December 1, 2018
Review comments to authors of early-submission papers: January 15, 2019
Normal Paper submission due: February 4, 2019
Final notification to authors: March 15, 2019.
Camera-ready manuscripts due: April 1, 2019
Congress Date: July 8 – 13, 2019

2019 Congress General Chair
Peter Chen, Carnegie Mellon University, USA

2019 Congress Program Chair-in-Chief
Elisa Bertino, Purdue University, USA

2019 Congress Vice Program Chair-in-Chief
Ernesto Damiani, University of Milan, Italy

Workshop Chairs
Shangguang Wang, BUPT
Stephan Reiff-Marganiec, University of Leicester

Steering Committee
Elisa Bertino, Purdue University, USA
Carl K. Chang, Chair, Iowa State University, USA,
Rong N. Chang, TCSVC Chair, IBM, USA
Peter Chen, Carnegie Mellon University USA
Ernesto Damiani, University of Milan, Italy
Ian Foster, University of Chicago
& Argonne National Lab, USA
Dennis Gannon, Indiana University, USA
Michael Goul, Arizona State University, USA
Frank Leymann, University of Stuttgart, Germany
Hong Mei, Beijing Institute of Technology, China
Stephen S. Yau, Arizona State University, USA

CALL FOR WORKSHOP PROPOSALS: See *http://conferences.computer.org/services/2019/* for more information.
Send inquiries to: ieeecs.services@gmail.com

DOI:10.1145/3186331

Conventional storage software stacks are unable to meet the needs of high-performance Storage-Class Memory technology. It is time to rethink 50-year-old architectures.

BY DANIEL WADDINGTON AND JIM HARRIS

Software Challenges for the Changing Storage Landscape

AS WE EMBARK on a new era of storage performance, the limitations of monolithic OS designs are beginning to show. New memory technologies (for example, 3D XPoint™ technology) are driving multi-GB/s throughput and access latencies at sub-microsecond scales. As the performance of these devices approaches the realms of DRAM, the overhead incurred by legacy IO stacks increasingly dominates.

To address this concern, momentum is gathering around new ecosystems that enable effective construction of tailored and domain-specific IO architectures. These ecosystems rely on bringing both device control and data planes into user space, so that they can be readily modified and intensely optimized without jeopardizing system stability.

This article begins by giving a quantitative exploration of the need to shift away from kernel-centric generalized storage IO architectures. We then discuss the fundamentals of user space (kernel-bypass) operation and the potential gains that result. Following this, we outline key considerations necessary for their adoption. Finally, we briefly discuss software support for NVDIMM-based hardware and how this is positioned to integrate with a user space philosophy.

Evolution of Storage IO

Since the release of the Intel 8237 in the IBM PC platform (circa 1972), network and storage device IO has centered around the use of Direct Memory Access (DMA). This enables the system to transfer data to and from a device to main memory with no involvement of the CPU. Because DMA transfers can be initiated to any part of main memory, coupled with the need to execute privileged machine instructions (for example, masking interrupts), device drivers of this era were well suited to the kernel. While executing device drivers in user space was in theory possible, it was unsafe because any misbehaving driver could easily jeopardize the integrity of the whole system.

As virtualization technologies evolved, the consequence of broad

» key insights

- **NVMe and memory-based storage technologies are experiencing an exponential growth in performance with aggressive parallelism and fast new media. Traditional IO software architectures are unable to sustain these new levels of performance.**
- **IOMMU hardware is a key enabler for realizing safe and maximal performing user space device drivers and storage IO stacks.**
- **Kernel-bypass strategies rely on "asynchronous polling" whereby threads actively check device completion queues. Naive designs can lead to excessive busy-waiting and inefficient CPU utilization.**

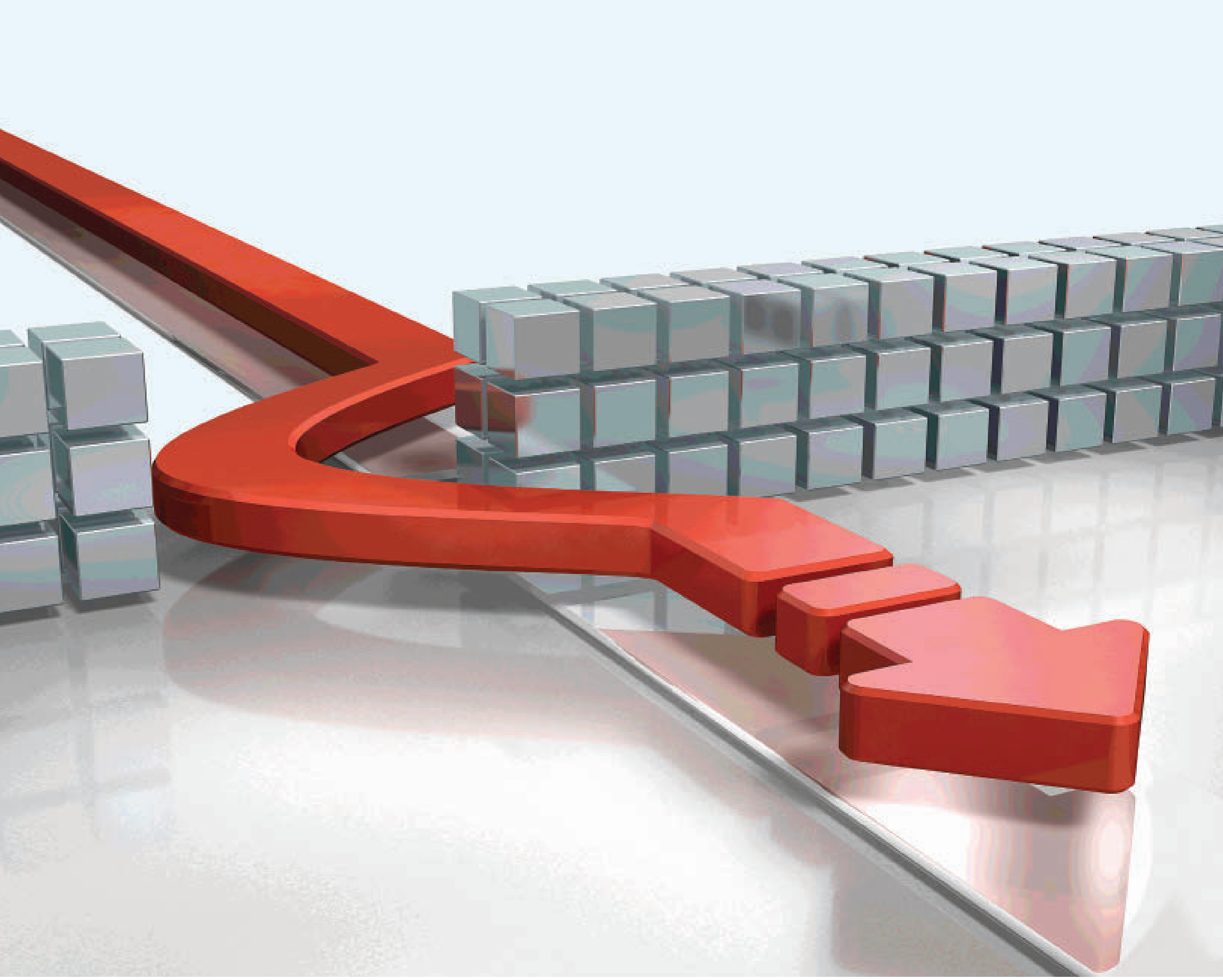


IMAGE BY CYCLONEPROJECT

access to memory for device drivers would become a prominent issue for system stability and protection between hosted virtual machines. One approach to this problem is the use of device emulation. However, emulation incurs a significant performance penalty because each access to the device requires a transition to the Virtual Machine Monitor (VMM) and back.

An alternative approach is to use para-virtualization that, by modifying the host OS device drivers so they interact directly with the hypervisor without entering the VMM, improves performance over prior emulation techniques. The downside is that the host OS code needs to be modified (including interrupt handling) and additional latency results from more IO layering.

Virtualization and direct device assignment. To minimize the impact of virtualization and indirection, certain use cases aim at providing a virtualized guest with direct ownership and access to specific hardware devices in the system. Such a scheme improves IO performance at the expense of the ability to transparently share devices across multiple guests using the hypervisor. The key enabling hardware technology for device virtualization is the IO Memory Management Unit (IOMMU). This provides the ability to remap device (DMA) addresses to physical addresses in the system, in much the same way that the MMU performs a translation from virtual to physical addresses (see Figure 1). Around 2006, IOMMU capabilities were made available in the Intel platform through its Virtualization Technology for Directed IO (VT-d)[1] followed by AMD-Vi on the AMD x86 platform.

Although IOMMU technology was driven primarily by the need to provide virtual machines with direct device access its availability has, more recently, become a pivotal enabler for rethinking device driver and IO architectures in non-virtualized environments.

Another significant advance in device virtualization was made by the PCI Express SR-IOV (Single Root IO Virtualization) extension.[11] SR-IOV enables multiple system images (SI) or virtual machines (VMs) in a virtualized environment to share PCI hardware resources. It reduces VMM overhead by giving VM guests direct access to the device.

New frontiers of IO performance. Over the last three decades, compute, network, and storage performance have grown exponentially according to Moore's Law (see Figure 2). However, over the last decade, growth of compute performance has slowed compared

Figure 1. MMU and IOMMU duality.

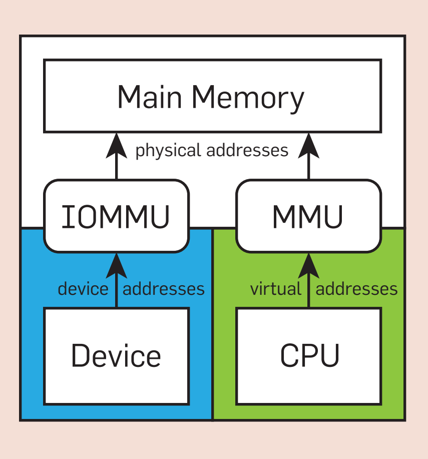
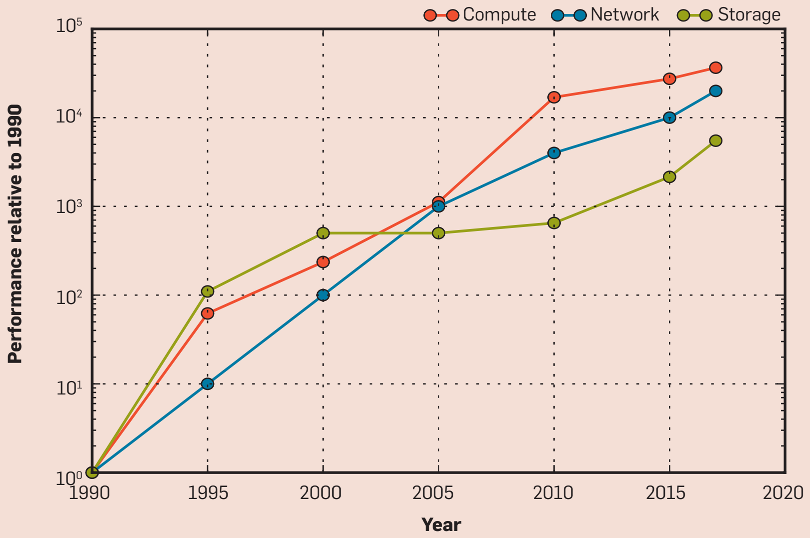


Figure 2. Relative IO performance growth. (Data collected by IBM Research, 2017)



to the growth of storage performance mainly due to the CPU frequency ceiling causing a shift in microprocessor growth strategy. We expect this trend to continue as new persistent memory technologies create an aggressive upswing in storage performance.

The accompanying table shows some performance characteristics of select state-of-the-art IO devices. Lower latency, increased throughput and density, and improved predictability continue to be key differentiators in the networking and storage markets. As latency and throughput improve, the CPU cycles available to service IO operations are reduced. Thus, it is evident that the latency overhead imposed by traditional kernel-based IO paths has begun to exceed the latency introduced by the hardware itself.

Reconsidering the IO Stack

The prominent operating systems of today, such as Microsoft Windows and Linux, were developed in the early 1990s with design roots stemming from two decades earlier. Their architecture is that of monolithic. This means that core OS functionality executes in kernel space and it cannot be readily modified or adapted. Threads within the kernel, are alone, given access to privileged processor instructions (for example, x86 Ring-0). Kernel functionality includes interrupt handling, file systems, scheduling, memory management, security, IPC and device drivers. This separation of user and kernel space came about as a solution to guard against "untrusted" applications from accessing resources that could interfere with other applications or interfere with OS functionality directly. For example, disallowing applications from terminating other applications or writing to memory outside of their protected memory space, is fundamental to system stability.

In the early stages of modern OS development, hardware parallelism was limited. It was not until more than a decade later (2006) that the multicore microprocessor would appear. Following the footsteps of multicore, network devices would also begin to support multiple hardware queues so that parallel cores could be used to service high-performance networking traffic. This multi-queue trend would also appear in storage devices, particularly with the advent of NVMe SSD (Solid State Drive). Today, hardware-level parallelism, in both CPU and IO is prominent. Intel's latest Xeon Platinum processors provide up to 28 cores and hyper-threading on each. AMD's latest, Naples server processor, based on Zen, provides 32 cores (64 threads) in a single socket. Many state-of-the-art NVMe drives and network interface cards support 64 or more hardware queues. The trend toward parallel hardware is clear and not expected to diminish anytime soon. The consequence of this shift from single-core and single-queue designs, to multi-core and multi-queue designs is that the IO subsystem has needed to evolve to support concurrency.

Strained legacy stacks. In terms of IO request rates, storage devices are an order of magnitude slower than network devices. For example, the fastest SSD devices operate at around 1M IOPS (IO operations per second) per device, whereas a state-of-the-art NIC device is capable of handling more than 70M packets per second. This slower rate

State-of-the-art IO device performance reference points.

Class	Technology	Performance
Compute	Server CPUs	24+ cores per die, 10nm, 4GHz, 100M transistors per mm2
Memory	DDR4-3200 DRAM	3200MT/s or 25.6GB/s (288-pin DIMM)
Backplane	PCI Express 4.0	16GT/s or 31GB/s for x16 channels
Networking	Mellanox ConnectX-6	200Gbps/200M PPS
SSD Storage	Intel Optane P4800X NVMe	2.3GB/s random read/write, < 10 usec latency
SSD Storage	Samsung PM1725a NVMe	6.4GB/s sequential read, 1.08MIOPS, 95 usec latency
Memory (future)	3D XPoint NV-DIMM Technology	< 1 usec latency (expected)

means that legacy OS improvement efforts in the storage space are still considered worthwhile.

With the advent of multicore, enhancing concurrency is a clear approach to improving performance. Many legacy OS storage subsystems realize concurrency and asynchrony through kernel-based queues serviced by worker threads. These are typically allocated for each processor core. Software queues can be used to manage the mapping between application threads running on specific cores, and the underlying hardware queues available on the IO device. This flexibility was introduced into the Linux 3.13 kernel in 2012[5] providing greatly improved IO scaling for multicore and multi-queue systems. The Linux kernel block IO architecture is aimed at providing good performance in the "general" case. As new IO devices (both network and storage) reach the realms of tens of millions of IOPS, the generalized architecture and layering of the software stack begin to strain. Even state-of-the-art work on improving kernel IO performance is limited in success.[15] Furthermore, even though the block IO layer may scale well, layering of protocol stacks and file systems typically increases serialization and locking, and thus impacts performance.

To help understand the relationship between storage IO throughput and CPU demand, Figure 3 shows IOPS scaling for the Linux Ext4 file system. This data is captured with the `fio` micro-benchmarking tool configured to perform random-writes of 4K blocks (random-read performance is similar). No filesharing is performed (the workload is independent). The experimental system is an Intel E5-2699 v4 two-socket server platform with 512GB main memory DRAM. Each processor has 22 cores (44 hardware threads) and the system contains x24 NVMe Samsung 172Xa SSD 1.5 TB PCIe devices. Total IO throughput capacity is ∼6.5M IOPS (25GB/s). Each device is PCI Gen 3 x8 (7.8GB/s) onto the PCI bus and a single QPI (memory bus) link is ∼19.2GB/s. Each processor has x40 PCI Gen 3.0 lanes (39.5GB/s).

The maximum throughput achieved is 3.2M IOPS (12.21GB/s). This is realized at a load of ∼26 threads (one per device) and 30% total CPU capacity. Adding threads from 17 to 26 gives negligible scaling. Beyond 26 worker threads, performance begins to degrade and become unpredictable although CPU utilization remains linear for some time.

File systems and kernel IO processing also add latency. Figure 4 shows latency data for direct device access (using Micron's kernel-bypass UNVMe framework) and the stock Ext4 file system. This data is from a single Intel Optane P4800X SSD. The filesystem and kernel latency (mean 13.92μsec) is approximately double that of the raw latency of the device (mean 6.25μsec). For applications where synchronous performance is paramount and latency is difficult to hide through pipelining, this performance gap can be significant.

Application-specific IO subsystems. An emerging paradigm is to enable customization and tailoring of the IO stack by "lifting" IO functions into user space. This approach improves system stability where custom IO processing is being introduced (that is, custom stacks can crash without jeopardizing system stability) and allows developers

Figure 3. Ext4 file system scaling on software RAID0.

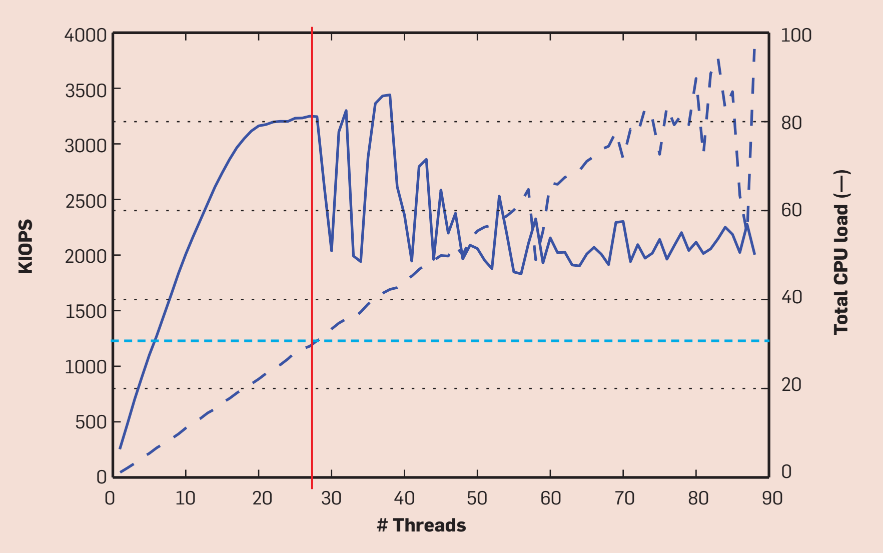
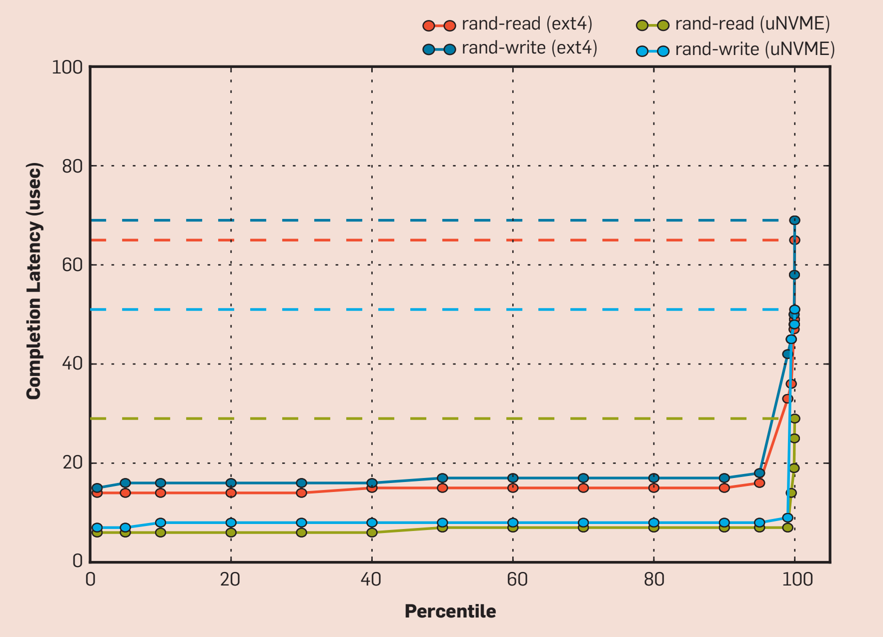


Figure 4. Ext4 vs. raw latency comparison.



to protect intellectual property where open source kernel licenses implies source release.

Although not originally designed for this purpose, a key enabler for user-level IO is the IOMMU. Specifically, the IOMMU provides the same capabilities to user-kernel processes as it does to guest-host virtualized OSes (see Figure 5). This effectively means that user-space device drivers (unprivileged processes) can be compartmentalized so that memory regions valid for device DMA operations can be limited by the IOMMU, and therefore device drivers can be prevented from accessing arbitrary memory regions (via a device's DMA engine).

Configuration of the IOMMU remains restricted to kernel functions operating at a higher privilege level (that is, ring 0). For example, in Linux, the Virtual Function IO (VFIO) kernel module can be used to configure the registered memory with the IOMMU and ensure memory is "pinned."

New architectures also allow interrupt handling to be localized to a subset of processor resources (that is, mapping MSI to specific local APICs) that are associated to a specific device driver execution. Coupled with device interrupt coalescing and atomic masking, this means that user-level interrupt handling is also viable. However, the interrupt vector must still reside in the kernel and be executed at a privileged level, at least for Intel and IBM Power architectures.

Foundational Kernel-bypass Ecosystems

In this section, we introduce the basic enablers for kernel-bypass in the Linux operating system. This is followed by a discussion of the Data Plane Development Kit (DPDK) and Storage Performance Development Kit (SPDK), two foundational open source projects started by Intel Corporation. DPDK has been widely adopted for building kernel-bypass applications, with over 30 companies and almost 400 individ-

Figure 5. MMU and IOMMU duality.

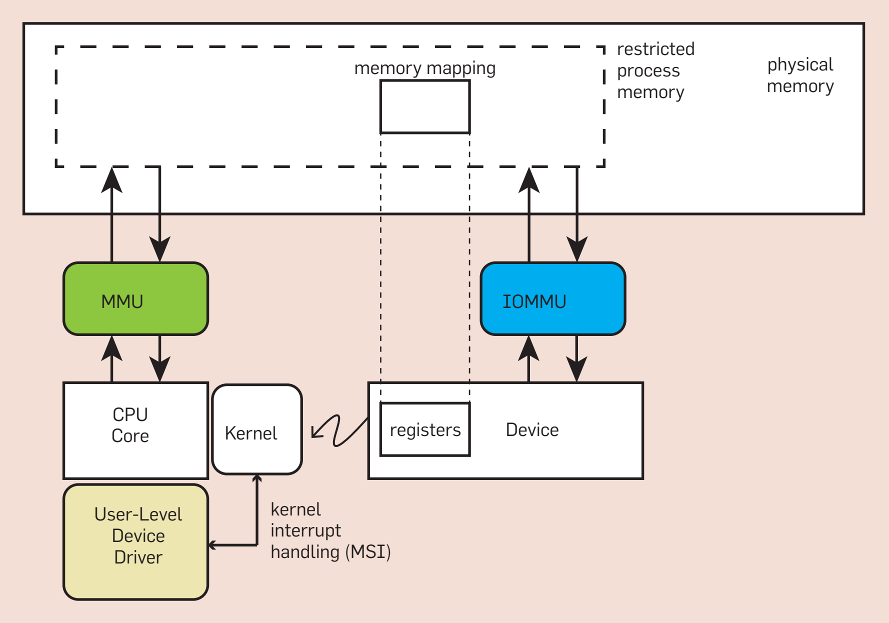
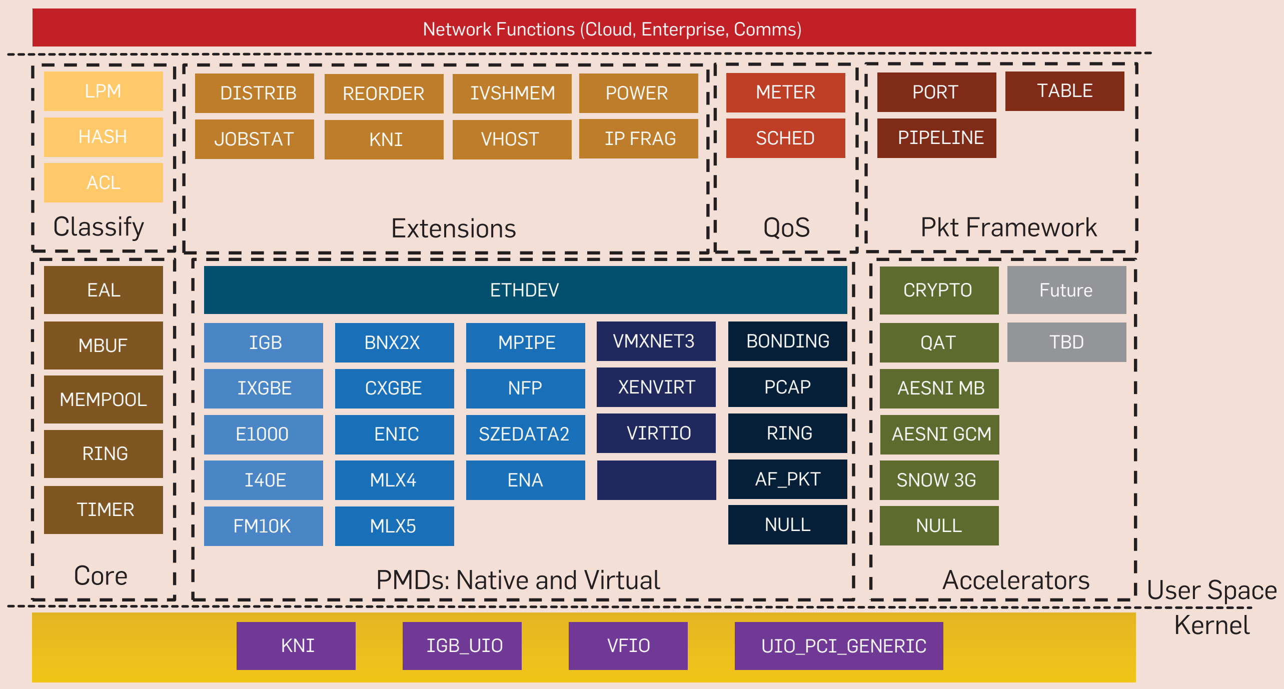


Figure 6. DPDK architecture.



uals contributing patches to the open source DPDK projects as of release 17.05. While DPDK is network-centric, it provides the basis for the SPDK storage-centric ecosystem. Other projects, such as FD.IO (http://fd.io) and Seastar (http://seastar-project.org) also use DPDK. These domain specifics are not discussed in this article.

Linux user space device enablers. Linux kernel version 2.6 introduced the User Space IO (UIO)[a] loadable module. UIO is the older of the two kernel-bypass mechanisms in Linux (VFIO being the other). It provides an API that enables user space handling of legacy INTx interrupts, but not message-signaled interrupts (MSI or MSI-X). UIO also does not support DMA isolation through IOMMU isolation. Even with these limitations, UIO is well suited for use in virtual machines, where direct IOMMU access is not available. In these situations, a guest VM user space process is not isolated from other processes in the same guest VM, but the hypervisor itself can isolate the guest VM from other VMs or host processes using the IOMMU.

For bare-metal environments, VFIO[b] is the preferred framework for Linux kernel-bypass. It operates with the Linux kernel's IOMMU subsystem to place devices into IOMMU groups. User space processes can open these IOMMU groups and register memory with the IOMMU for DMA access using VFIO ioctls. VFIO also provides the ability to allocate and manage message-signaled interrupt vectors.

Data plane development kit. DPDK (http://dpdk.org) was originally aimed at accelerating network packet processing applications. The project was initiated by Intel Corporation, but is now under the purview of the open source Linux Foundation. At the core of DPDK is a set of polled-mode Ethernet drivers (PMDs). These PMDs bypass the kernel, and by doing so, can process hundreds of millions of network-packets per second on standard server hardware.

DPDK also provides libraries to aid kernel-bypass application development. These libraries enable probing for PCI devices (attached via UIO or VFIO), allocation of huge-page memory, and data structures geared toward polled-mode message-passing applications such as lockless rings and memory buffer pools with per-core caches. Figure 6 shows key components of the DPDK framework.

Storage performance development kit. SPDK is based on the foundations of DPDK. It was introduced by Intel Corporation in 2015 with a focus on enabling kernel-bypass storage and storage-networking applications using NVMe SSDs. While SPDK is primarily driven by Intel, there are an increasing number of companies using and contributing to the effort. The project desires broader collaboration that may require adoption of a governance structure similar to DPDK. SPDK shows good promise for filling the same role for storage and storage networking as DPDK has for packet processing.

SPDK's NVMe polled-mode drivers provides an API to kernel-bypass applications for both direct-attached NVMe storage as well as remote storage using the NVMe over Fabrics protocol. Figure 7 shows the SPDK framework's core elements as of press time. Using SPDK, Walker[22] shows reduction in IO submission/completion overhead by a factor of 10 as measured with the SPDK software overhead measurement tool.

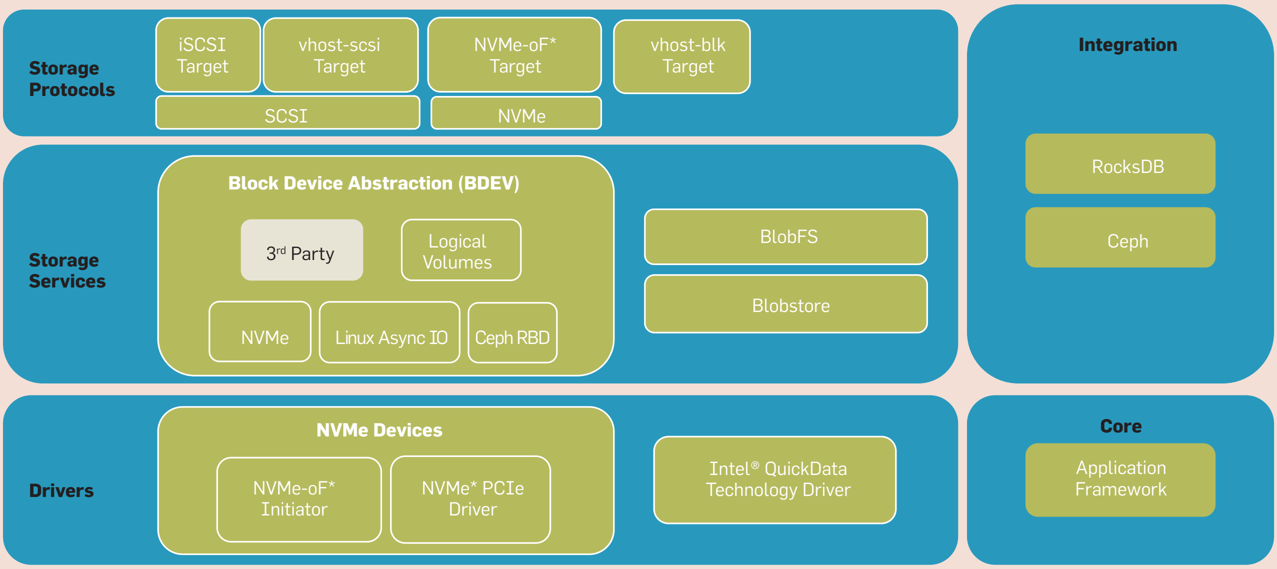
To provide the reader with a better understanding of the impact of legacy IO we present data from the 'fio' benchmarking tool (https://github.com/axboe/fio). Figure 8 shows performance data, for kernel-based IO (with Ext4 and raw block access) and SPDK. The data compares throughput with the number of client threads. Configuration is queue depth of 32, and IO size of 4KiB. Sequential read, sequential write, random read, random write, and 50:50 read-write workloads are examined.

The key takeaway is that SPDK requires only one thread to get over 90% of the device's maximum performance.

[a] https://lwn.net/Articles/232575/

[b] https://www.kernel.org/doc/Documentation/vfio.txt

Figure 7. SPDK architecture.



Note also that the SPDK data represents a 1:1 mapping of threads to hardware queues and therefore the number of threads is limited to the number of queues available (limited to 16 queues in this case). The kernel-based data represents the number of user-threads multiplexed (via two layers of software queues) to the underlying device queues.[5]

From the data, we can see that generality and the associated functionality impact performance. Reducing software overhead by tailoring and optimizing the stack (according to specific application requirements) improves storage applications in two ways. First, with fewer CPU cycles spent on processing IO, more CPU cycles are available for storage services such as compression, encryption, or storage networking. Second, with the advent of ultra-low latency media, such as Intel Optane, higher performance can be achieved for low queue depth workloads since the software overhead is much smaller compared to the media latency.

Klimovic et al.[14] have applied DPDK and SPDK in the context of distributed SSD access. Their results show performance improvements for the FlashX graph-processing framework of up to 40% versus iSCSI. They also make a comparison with RocksDB and show a delta of ~28% between iSCSI and their solution. This work is based on the IX Dataplane Operating system,[3] which is fundamentally based on kernel-bypass approaches.

Kernel-Bypass Design Considerations

Here, we present some design aspects and insights that adopters of kernel-bypass technology, such as DPDK and SPDK, should consider.

Cost of context switching. Raising IO operations into user space requires careful consideration of software architecture. Traditional OS designs rely on interrupts and context switching to multiplex access to the CPU. In a default Linux configuration for example, the NVMe device driver will use a per-core submission queue, serviced by the same core, and therefore context switching cannot be avoided.

Context switches are costly (more so than system calls) and should be avoided at high IO rates. They result in cache pollution that arises from both eviction of cache by the task contexts and the subsequent impact of working set memory of the newly scheduled task. The typical cost of a context switch is in the order of 2,000–5,000 clock cycles. Figure 9 presents data from lmbench (http://www.bitmover.com/lmbench/) running on a dual-socket Intel E5-2650 v 4 @ 2.2GHz, 32K L1, 256K L2, and 30MB L3 caches.

Polling-based designs minimize IO latency by eliminating the need to execute interrupt handlers for inbound IO, and removing system calls/context switches for outbound IO. However, polling threads must be kept busy performing useful work as opposed to spending time polling empty or full queues (busy-work).

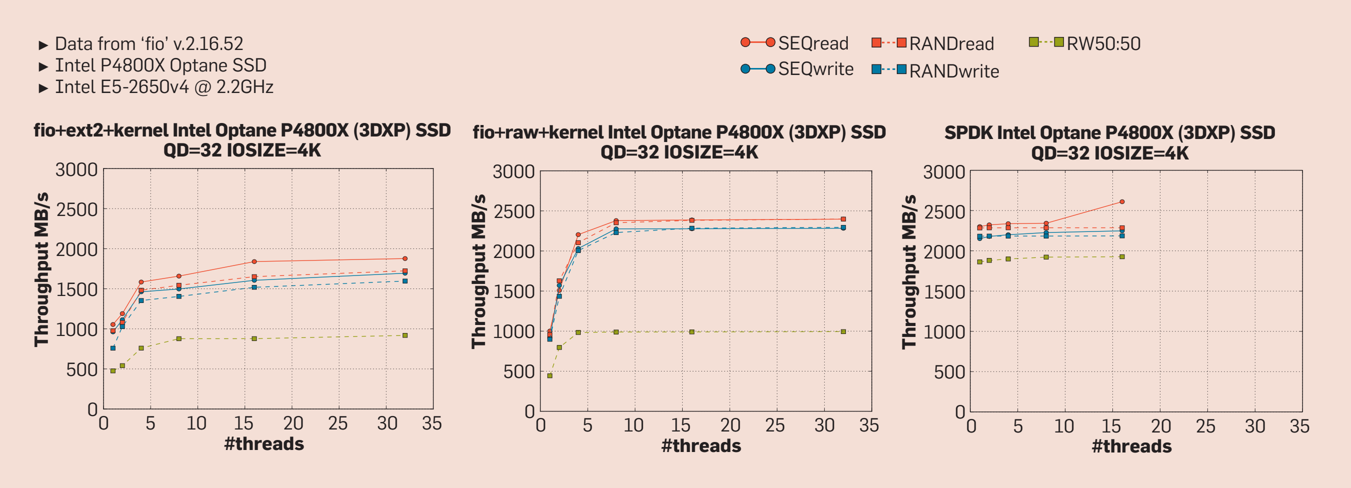
Asynchronous polling. A key design pattern that can be used to improve the utility of polling threads is asynchronous polling. Here, polling threads can round-robin (or some other scheduling policy) across multiple asynchronous tasks. For example, a single thread might service both hardware and software queues at the same time (see Figure 10). Hardware queues reside in memory on the device and are controlled by the device itself, while software queues reside in main memory and are controlled by the CPU. IO requests typically flow through both. Polling is asynchronous in that the thread does not synchronously wait for completions of a specific request, but retrieves the completion at a later point in time.

Asynchronous polling can be coupled with lightweight thread scheduling (co-routines) found, for example, in Intel Cilk.[16] Such technologies allow program-level logical concurrency to be applied without the cost of context switching. Each kernel thread services a task queue by applying stack swapping to redirect execution. Lightweight scheduling schemes typically execute tasks to completion, that is, they are non-preemptive. This is well suited to asynchronous IO tasks.

Lock-free inter-thread communications. Because polling threads cannot perform extensive work without risking device queue overflow (just as conventional interrupt service routines must be tightly bound and therefore typically defer work) they must off-load work to, or receive work from, other application threads.

This requires that threads must coordinate execution. A practical design pattern for this is message passing across lock-free FIFO queues. Differ-

Figure 8.Comparison of fio performance for Linux kernel vs. SPDK.


▸ Data from 'fio' v.2.16.52
▸ Intel P4800X Optane SSD
▸ Intel E5-2650v4 @ 2.2GHz

ent lock-free queue implementations can be used for different ratios of producer and consumer (for example, single-producer and single-consumer, single-producer and multi-consumer). Lock-free queues are well suited to high-performance user-level IO since they do not require kernel-level locking, but rely on machine-level atomic instructions. This means that exchanges can be performed without forcing a context switch (although one may undoubtedly occur if scheduled).

A basic implementation of lock-free queues will perform busy-waiting when the queue is empty or full. This means that the thread is continuously reading memory state and thus consuming 100% of the CPU it is running on. This results in high-energy utilization. Alternatively, it is possible to implement lock-free queues that support thread sleeping in empty or full conditions. This avoids busy-waiting by allowing the OS to schedule other threads in its place. Sleeping can be supported on either or both sides of the queue. To avoid race-conditions, implementations typically use an additional "waker" thread. This pattern is well-known in the field of user-level IPC (Inter-Process Call).[20]

Lock-free queues are established in shared memory and can be used for both inter-process and inter-thread message exchange. To optimize inter-thread message passing performance processor cores (on which the threads execute) and memory should belong to the same NUMA zone. Accessing memory across remote NUMA zones incurs approximately twice the access latency.

Combining polling and interrupt modes. Another strategy to avoid busy-waiting on queues is to combine polling and interrupt modes. To support this, VFIO provides a capability to attach a signal (based on a file-descriptor) to an interrupt so that a blocking user-level thread can be alerted when an interrupt event occurs. The following excerpt illustrates connecting an MSI interrupt to a file handle using the POSIX API:

```
efd = eventfd(0, 0);

ioctl(vfIO fd ,
VFIO _ EVENT _ FDMSI, &efd);
```

In this case, the IO threads wait on file descriptor events through read or poll system calls. Of course, this mechanism is costly in terms of performance since waking up and signaling a user-level thread from the kernel is expensive. However, because the interrupt is masked when generated, the user-thread controls unmasking and can thus arbitrarily decide when to revert to interrupt mode (for example, when an extended period of "quiet" time has passed).

Memory paging and swapping. An important role of the kernel is to handle page-faults and "swapping" memory to backing store when insufficient physical memory is available. Most operating systems use a lazy mapping strategy (demand-paged) so that virtual pages will not be mapped to physical pages until they are touched. Swapping provides an extended memory model; that is, the system presents to the application the appearance of more memory than is in the system. The mechanisms behind swapping are also used for mapped files, where a file copy shadows a region of memory. Traditionally, swapping is not heavily used because the cost of transferring pages to storage devices that are considerably slower than memory is significant.

For monolithic OS designs, page swapping is implemented in the kernel. When there is no physical page mapping for a virtual address (that is, there is no page table entry), the CPU generates a page-fault. In the Intel x86 architecture, this is realized as a machine exception. The exception handler is run at a high privilege level (CPL 0) and thus remains in the kernel. When a page-fault occurs, the kernel allocates a page of memory from a pool (typically known as the page cache) and maps the page to the virtual address by updating the page table. When the physical memory pool is exhausted, the kernel must evict an existing page by writing out the content to backing store and invalidating the page table entry (effectively un-mapping the page). In Linux, the eviction policy is based on a variation of the Least Recently Used (LRU) scheme.[10] This is a generalized policy aimed at working well for most workloads.

Because page-fault handling and page swapping rely on the use of privileged instructions and exception handling, implementing them in user space alone is inherently difficult. One approach is to use the POSIX mprotect and mmap/mumap system calls to explicitly control the page mapping process. In this case, page protection PROT_NONE can be used to force the kernel to raise a signal on the user-level process when the unmapped page is accessed. In our own work, we have been able to realize a paging overhead of around 20usec per 4K page (with SPDK-based IO), which is

Figure 9. Context switch latencies on Intel E5-2650 based server.

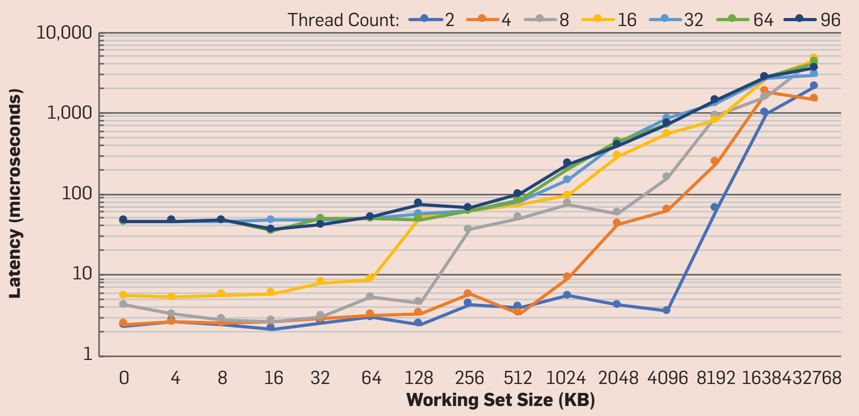
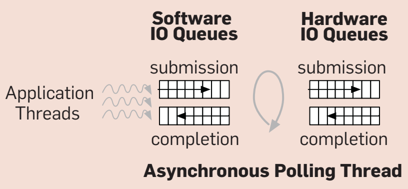


Figure 10. Asynchronous polling pattern.



comparable to that of the kernel (tested against memory mapped files).

Memory flushing Linux. To optimize write-through to storage, it is also necessary to track dirty pages, so that only those that have been modified are flushed out to storage. If a page has only been read during its active mapping, there is no need to write it back out to storage. From the kernel's perspective, this function can be easily achieved by checking the page's dirty bit in its corresponding page table entry. However, as noted earlier, accessing the page table from user space is problematic. In our own work, we have used two different approaches to address this problem.

The first is to use a CRC checksum over the memory to identify dirty pages. Both Intel x86 and IBM Power architectures have CRC32 accelerator instructions that can compute a 4K checksum in less than ~1000 cycles. Note that optimizations such as performing the CRC32 on 1024 byte blocks and performing a "short circuit" of the dirty page identification can reduce further the cost of CRC in this context.

An alternative approach is to use a kernel module to collect dirty page information on request from an application. This, of course, incurs an additional system call and page table walk. Consequently, this approach performs well with small page tables, but is less performant than CRC when traversal across many page table entries is needed.

Legacy integration. Designing around a kernel bypass architecture is a significant paradigm shift for application development. Consequently, there are some practical limitations to their adoption in legacy systems. These include:

- Integration with existing applications ased on a blocking threading model requires either considerable rewriting to adhere to an asynchronous/polling model, or shims to bridge the two together. The latter reduces the potential performance benefits.
- Sharing storage devices between multiple processes. Network devices handle this well via SR-IOV, but NVMe SR-IOV has only recently been added to NVMe specification. Hence, sharing NVMe devices across multiple devices must be done through software.
- Integration with the existing file system structures is difficult. While conceptually Filesystem in User Space (FUSE) technology could be used to integrate into the kernel-based file system hierarchy, the advantages of performance would be lost because of the need to still pass control into the kernel. Evolution of the POSIX API is needed to support hybrid kernel and user IO. "Pure" user-space file systems are still not broadly available.
- Legacy file systems and protocol stacks incorporate complex software that has taken years of development and debugging. In some cases, this software can be integrated through "wrappers." However, in general this is challenging and redeveloping the software from the ground up is more economic.

Polling-based designs minimize IO latency by eliminating the need to execute interrupt handlers for inbound IO, and removing system calls/context switches for outbound IO.

Integration of NVDIMMs

Non-Volatile Dual Inline Memory Modules (NVDIMMs) attach non-volatile memory directly to the memory bus, opening the possibility of application programs accessing persistent storage via load/store instructions. This requires additional libraries and/or programming language extensions[5,9] to support the coexistence of both volatile and non-volatile memory. The fundamental building blocks needed are persistent memory management (for example, pool and heap allocators), cache management, transactions, garbage collection, and data structures that can operate with persistent memory (for example, support recovery and reinstantiation).

Today, two prominent open source projects are pushing forward the development of software support for persistent memory. These are pmem.io (http://pmem.io/), driven primarily by Intel Corporation in conjunction with SNIA, and The Machine project (https://www.labs.hpe.com/the-machine) from HP Labs. These projects are working to build tools and libraries that support access and management of NVDIMM. Key challenges that are being explored by these projects and others,[3,8,17,20] include:

- *Cross-heap pollution:* Pointers to volatile data structures should not "leak" into the non-volatile heap. New programming language semantics are needed to explicitly avoid programming errors that lead to dangling and invalid references.

- *Transactions:* Support for ACID (atomicity, consistency, isolation, durability) transactions offering well-defined guarantees about modifications to data structures that reside in persistent memory and are accessible by multiple threads.
- *Memory leaks and permanent corruption:* Persistence makes memory leaks and errors that are normally recoverable through program restart or reset, more pernicious. Strong safety guarantees are needed to avoid permanent corruption.
- *Performance:* Providing tailored capabilities and leveraging the advantages of low latency and high throughput enabled by NVDIMM technology.
- *Scalability:* Scaling data structures to multi-terabytes also require scaling of metadata and region management structures.
- *Pointer swizzling:* Modifying embedded (virtual address) pointer references for object/data structure relocation.[21]

The real impact of NVDIMMs remains to be seen. However, work by Coburn et al.[6] on NV-Heaps has shown that for certain applications the move from a transactional database to persistent memory can bring significant performance gains.

NVDIMM-based persistent memory lends itself to integration with user space approaches because it inherently provides access directly to the user space application (although mapping and allocation may remain the kernel's control). This enables efficient, zero-copy DMA-centric movement of data through the memory hierarchy and into the storage device. A longer-term vision is for a converged memory-storage paradigm whereby traditional storage services (for example, durability, encryption) can be layered into the memory paradigm. However, to date, this topic remains largely unaddressed by the community.

Outlook

Mainstream operating systems are based on IO architectures with a 50-year heritage. New devices now challenging these traditional designs bring unprecedented levels of concurrency and performance. The result is that we are entering an era of CPU-IO performance inversion, where CPU resources are becoming the bottleneck. Careful consideration of execution paths is now paramount to effective system design.

User space, kernel-bypass strategies, provide a vehicle to explore and quickly develop new IO stacks. These can be used to exploit alignment of requirements and function, becoming readily tailored and optimized to meet the specific needs of an application. Flexibility of user space software implementation (as opposed to kernel space) enables easier development and debugging, and enables the leverage of existing application libraries (for example, machine learning).

For the next decade, microprocessor design trends are expected to continue to increase on die transistor count. As instruction-level parallelism and clock frequency increases have reached a plateau, increased core count and on-chip accelerators are the most likely differentiators for future processor generations. There is also the possibility of "big" and "little" cores whereby heterogeneous cores, with different capabilities (for example, pipelining, floating point units, and clock frequency), exist on the same processor package. This is already evident in ARM-based mobile processors. Such an approach could help drive a shift away from interrupt-based IO, toward polling IO whereby "special" cores are dedicated to IO processing (possibly at a lower clock frequency). This would both eliminate context switches and cache pollution, and would also enable improved energy management and determinism in the system.

Large capacity, NVDIMM-based persistent memory is on the horizon. The availability of potentially up to terabytes of persistent memory, with sub-microsecond access latencies and cache-line addressability, will accelerate the need to make changes in the IO software architecture. User space IO strategies are well positioned to meet the demands of high-performance storage devices and to provide an ecosystem that can effectively adopt load/store addressable persistence.

References

1. Abramson, D. et al. Intel virtualization technology for directed IO. *Intel Technology J. 10*, 3 (2006), 179–192.
2. Atkinson, M. and Morrison, R. Orthogonally Persistent Object Systems. *The VLDB J. 4*, 3 (July 1995), 319–402.
3. Belay, A., Prekas, G., Klimovic, A., Grossman, S., Kozyrakis, C. and Bugnion, E. IX: A protected dataplane operating system for high throughput and low latency. In *Proceedings of USENIX Operating Systems Design and Implementation*, Oct. 2014, 49–65.
4. Bhattacharya, S.P. A Measurement Study of the Linux TCP/IP Stack Performance and Scalability on SMP systems, Communication System Software and Middleware, 2006.
5. Bjørling, M., Axboe, J., Nellans, D. and Bonnet, P. Linux block IO: Introducing multi-queue SSD access on multi-core systems. In *Proceedings of the 6th International Systems and Storage Conf.*, 2013, 22:1–22:10. ACM, New York, NY, USA.
6. Coburn, J. et al. NV-Heaps: Making persistent objects fast and safe with next-generation, non-volatile memories. *SIGPLAN Notices 46*, 3 (Mar. 2011), 105–118.
7. Dearle, A., Kirby, G.N.C. and Morrison, R. Orthogonal persistence revisited. In *Proceedings of the 2nd International Conference on Object Databases*, 2010, Springer Berlin, Heidelberg.
8. Gorman, M. *Understanding the Linux Virtual Memory Manager.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.
9. Grundler, G. Porting drivers to HP ZX1. Ottawa Linux Symposium, 2002.
10. Intel Corporation. Intel 64 and IA-32 Architectures Optimization Reference Manual. No. 248966-033, June 2016.
11. Intel Corporation. PCI-SIG Single Root IO Virtualization Support in Intel® Virtualization Technology for Connectivity; https://www.intel.com/content/dam/doc/white-paper/pci-sig-single-root-io-virtualization-support-in-virtualization-technology-for-connectivity-paper.pdf
12. Kannan, S., Gavrilovska, A. and Schwan, K. PVM: Persistent virtual memory for efficient capacity scaling and object storage. In *Proceedings of the 11th European Conference on Computer Systems*, 2016, 13:1–13:16. ACM, New York, NY, USA.
13. Kemper, A. and Kossmann, D. Adaptable pointer swizzling strategies in object bases: Design, realization, and quantitative analysis. *International J. Very Large Data Bases 4*, 3 (July 1995), 519–567.
14. Klimovic, A., Litz, H. and Kozyrakis, C. ReFlex: Remote Flash Local Flash. In *Proceedings of the 22nd International Conference on Architectural Support for Programming Languages and Operating Systems*, 2017, 345–359. ACM, New York, NY.
15. Kumar, P. and Huang, H. Falcon: Scaling IO performance in multi-SSD volumes. In *Proceedings of USENIX Annual Technical Conference* (Santa Clara, CA, July 2017).
16. Lewin-Berlin, S. Exploiting multicore systems with Cilk. In *Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, 2010, 18–19. ACM, New York, NY, USA. ACM.
17. Lin, F.X. and Liu, X. Memif: Towards programming heterogeneous memory asynchronously. *SIGARCH Computing Architecture News 44*, 2 (Mar. 2016), 369– 383.
18. Siemon, D. Queueing in the Linux network stack. *Linux J. 231* (July 2013).
19. Tuning throughput performance for Intel Ethernet adapters (2017); http://www.intel.com/content/www/us/en/support/ network-and-i-o/ethernet-products/000005811.html
20. Unrau, R. and Krieger, O. Efficient sleep/wake-up protocols for user-level IPC. In *Proceedings of the 1998 International Conference on Parallel Processing*.
21. Volos, H., Tack, A.J. and Swift, M.M. Mnemosyne: Lightweight persistent memory. *SIGPLAN Notices 47*, 4 (Mar. 2011), 91–104.
22. Walker, B. SPDK: Building blocks for scalable high-performance storage applications. SNIA Storage Developer Conference, 2016, Santa Clara, CA, USA; https://www.snia.org/sites/default/files/SDC/2016/presentations/performance/BenjaminWalker_SPDK_Building_Blocks_SDC_2016.pdf,

Daniel Waddington (daniel.waddington@ibm.com) is a research staff member at IBM Almaden Research Center in San Jose, CA, USA.

Jim Harris (james.r.harris@intel.com) is a principal engineer in the Network Platforms Group at Intel Corporation, Chandler, AZ, USA.

research highlights

DOI:10.1145/3266289

To view the accompanying paper, visit doi.acm.org/10.1145/3266291

Technical Perspective
Backdoor Engineering

By Markus G. Kuhn

IMAGINE YOU ARE a cyber spy. Your day job is to tap cryptographically protected communications systems. But how? Straightforward cryptanalysis has long become impractical: the task of breaking modern algorithms, if implemented correctly, far exceeds all computational power available to humanity. That leaves *sabotage*.

You can target many Achilles heels of a crypto system: random-bit generators, side channels, binary builds, certification authorities, and weak default configurations. You infiltrate the teams that design, implement, and standardize commercial security systems and plant there hidden weaknesses, known as *backdoors,* that later allow you to bypass the cryptography.

Take random-bit generation. Security protocols distinguish intended peers from intruders only through their knowledge of secret bit sequences. Servers have to choose many key values at random to protect each communication session, and an adversary who can successfully guess these can impersonate legitimate users.

One trick to backdoor random bits can be understood with basic high-school algebra. A deterministic random-bit generator (DRBG) is initialized (seeded) with a start state s_0, and then iterated with some generator function: $s_{i+1} := G(s_i)$

$$\begin{tikzcd} s_0 \arrow[r, "G(s_0)"] & s_1 \arrow[r, "G(s_1)"] & s_2 \; \ldots \end{tikzcd}$$

In simple DRBGs (say, for simulations), the s_i may serve as both the state of the generator, as well as its output. So anyone who saw an output s_i and knows G can easily predict all future outputs. Crypto-grade DRBGs make four improvements: (a) hardware noise sources (slow) seed s_0, (b) the state s_i has hundreds or thousands of bits, (c) a second function H derives output values $r_i := H(s_i)$ from the internal state

$$\begin{tikzcd} s_0 \arrow[r, "G(s_0)"] \arrow[d, "H(s_0)"] & s_1 \arrow[r, "G(s_1)"] \arrow[d, "H(s_1)"] & s_2 \; \ldots \arrow[d, "H(s_2)"] \\ r_0 & r_1 & r_2 \end{tikzcd}$$

and (d) both G and H are *one-way functions*. These can be computed efficiently, but their inverses not. After H, an adversary who can see some of the outputs r_i cannot infer anything about the internal states s_i or other outputs r_j. We know many excellent choices for G and H: one-way functions or permutations carefully engineered to be fast and to have no other known exploitable properties. Most are constructed from secure hash functions or block ciphers.

As a saboteur, you do not want these used. Instead, you lure your victims toward a far more dangerous option: the class of algebraic one-way functions that enabled public-key crypto. These are orders of magnitude slower and require much bigger values for equal security. Modular exponentiation is a simple example. If you follow a few rules for choosing a big integer g and a big prime number p, then $G(x) := g^x \bmod p$ is such a one-way function. While g^x alone is monotonic, and thus easy to invert, the mod p operation (take the remainder after division by p) ensures the result remains uniformly spread over a fixed interval and appears to behave highly randomly. The inverse *discrete logarithm* problem, of calculating x when given $(g^x \bmod p, p, g)$, becomes computationally infeasible, and we have a one-way function. (In the following, we drop mention of the mod p operation, and just apply it automatically after each arithmetic operation.) The exponentiation operator g^x has an important additional property, not affected by the mod operation: $(g^x)^y = (g^y)^x$. While this commutativity is useless to honest designers of DRBGs, it can be invaluable to saboteurs.

Convince your victims that $G(s_i) := g^{s_i}$ and $H(s_i) := h^{s_i}$ are excellent choices for generating random numbers of the highest security:

$$\begin{tikzcd} s_0 \arrow[r, "g^{s_0}"] \arrow[d, "h^{s_0}"] & s_1 \arrow[r, "g^{s_1}"] \arrow[d, "h^{s_1}"] & s_2 \; \ldots \arrow[d, "h^{s_2}"] \\ r_0 & r_1 & r_2 \end{tikzcd}$$

You can claim "provable security based on number-theoretical assumptions," but this is, of course, just a smoke screen. The sole advantage of this construction is that it allows a backdoor. If you can choose g as $g := h^e$, then knowing your secret integer e immediately allows you to convert any output value r_i into the next internal state of the DRBG as $(r_i)^e = (h^{s_i})^e = (h^e)^{s_i} = g^{s_i} = s_{i+1}$:

$$\begin{tikzcd} s_0 \arrow[r, "g^{s_0}"] \arrow[d, "h^{s_0}"] & s_1 \arrow[r, "g^{s_1}"] \arrow[d, "h^{s_1}"] & s_2 \; \ldots \arrow[d, "h^{s_2}"] \\ r_0 \arrow[ur, dashed, "r_0^e"'] & r_1 \arrow[ur, dashed, "r_1^e"'] & r_2 \end{tikzcd}$$

So, if you contact a server and receive one r_i, you can now immediately predict all future r_j used to protect the communication with others, and decrypt or impersonate their messages. Job done. And nobody else can do this, because finding e from h and g is computationally infeasible (the aforementioned discrete logarithm problem). Unless, of course, they steal your backdoor by generating their own e' and replacing your g with their $g' := h^{e'}$.

The following article by Checkoway et al. reports on the amazing independent reconstruction of exactly such a backdoor, discovered in the firmware of a VPN router commonly used to secure access to corporate intranets. In 2004, the NSA planted the above DRBG in NIST standard SP 800-90, including a g and h of their choice. The details differ only slightly (elliptic curve operations rather than modular exponentiation, which uses slightly different notation; the top 16 bits of r_i discarded, can be guessed via trial and error). The basic idea is identical.

But planting a backdoor in a standard is not enough. You now also have to ensure industry implements it correctly, such that an r_i reaches you intact. And that nobody else replaces your g. And that is where this story begins. ■

Markus G. Kuhn (mgk25@cam.ac.uk) is a Senior Lecturer teaching computer security and cryptography at the University of Cambridge, England.

DOI:10.1145/3266291

Where Did I Leave My Keys?

Lessons from the Juniper Dual EC Incident

By Stephen Checkoway, Jacob Maskiewicz, Christina Garman, Joshua Fried, Shaanan Cohney, Matthew Green, Nadia Heninger, Ralf-Philipp Weinmann, Eric Rescorla, and Hovav Shacham

Abstract

In December 2015, Juniper Networks announced multiple security vulnerabilities stemming from unauthorized code in ScreenOS, the operating system for their NetScreen Virtual Private Network (VPN) routers. The more sophisticated of these vulnerabilities was a passive VPN decryption capability, enabled by a change to one of the parameters used by the Dual Elliptic Curve (EC) pseudorandom number generator.

In this paper, we described the results of a full independent analysis of the ScreenOS randomness and VPN key establishment protocol subsystems, which we carried out in response to this incident. While Dual EC is known to be insecure against an attacker who can choose the elliptic curve parameters, Juniper had claimed in 2013 that ScreenOS included countermeasures against this type of attack. We find that, contrary to Juniper's public statements, the ScreenOS VPN implementation has been vulnerable to passive exploitation by an attacker who selects the Dual EC curve point since 2008. This vulnerability arises due to flaws in Juniper's countermeasures as well as a cluster of changes that were all introduced concurrently with the inclusion of Dual EC in a single 2008 release. We demonstrate the vulnerability on a real NetScreen device by modifying the firmware to install our own parameters, and we show that it is possible to passively decrypt an individual VPN session in isolation without observing any other network traffic. This incident is an important example of how guidelines for random number generation, engineering, and validation can fail in practice. Additionally, it casts further doubt on the practicality of designing a safe "exceptional access" or "key escrow" scheme of the type contemplated by law enforcement agencies in the United States and elsewhere.

1. INTRODUCTION

In December 2015, Juniper announced that an "internal code review" revealed the presence of "unauthorized code in ScreenOS that could allow a knowledgeable attacker [...] to decrypt VPN connections." In response to this, Juniper released patched versions of ScreenOS, the operating system powering the affected NetScreen devices, but has declined to disclose any further information about the intrusion and vulnerability.

Immediately following Juniper's advisory, security researchers around the world—including our team—began examining the ScreenOS firmware to find the vulnerabilities Juniper had patched. They found that the change that rendered ScreenOS encryption breakable did nothing but replace a few embedded constants in Juniper's pseudorandom number generator. The reason why this results in an attacker being able to decrypt connections is Juniper's design decision to use the NSA-designed Dual EC Pseudorandom Number Generator (PRNG).[4, 12] Dual EC has the problematic property that an attacker who knows the discrete logarithm of one of the input parameters (Q) with respect to a generator point, and is able to observe a small number of consecutive bytes from the PRNG, can then compute the internal state of the generator and thus predict all future output. Thus, it is critical that the discrete logarithm of Q remain unknown. The changes to the ScreenOS code replaced Juniper's chosen Q with one selected by the attacker.

From one perspective, the Juniper incident is just a particularly intricate software vulnerability, which is interesting on its own terms. More importantly, however, it sheds light on the contentious topic of "exceptional access" technologies which would allow law enforcement officials to gain access to the plaintext for encrypted data. A key component of any exceptional access system is restricting access to authorized personnel, with the most commonly proposed approach being encrypting the target keying material under a key (or keys) known to law enforcement which are then kept under tight control. The use of Dual EC in ScreenOS creates what is in effect an exceptional access system with Q as the public key and the discrete log of Q as the private decryption key. Historically, analysis of exceptional access systems has focused on the difficulty of controlling the decryption keys. In the specific case of ScreenOS, we do not know whether anyone had access to the corresponding key, but the Juniper incident starkly illustrates another risk: that of an attacker modifying a system's exceptional access capability in order to replace the authorized public key with one under her control, thus turning an exceptional access system designed for use by law enforcement into one which works for the attacker.

In this paper, we attempt to tell the story of that incident, pieced together by forensic reverse engineering of dozens of ScreenOS firmware revisions stretching back nearly a decade, as well as experimental validation on NetScreen hardware. We first provide background on Dual EC itself, then examine the way that it is used in ScreenOS and why this leads to such a severe vulnerability, then

The original version of this paper is entitled "A Systematic Analysis of the Juniper Dual EC Incident" and was published in *Proceedings of the 23rd ACM Conference on Computer and Communications Security* (Vienna, 2016), 468–479.

move to examine the history of the incident itself, and finally consider what lessons we can draw from this story.

2. DUAL EC IN SCREENOS

Cryptographic systems typically include deterministic PRNGs that expand a small amount of secret internal state into a stream of values which are intended to be indistinguishable from true randomness. An attacker able to predict the output of a PRNG will often be able to break any protocol implementation dependent on it, for instance by being able to predict cryptographic keys (which should remain secret) or nonces (which should often remain unpredictable).

Dual EC is a cryptographic PRNG standardized by National Institute of Standards and Technology (NIST) which is based on operations on an elliptic curve. Dual EC has three public parameters: the elliptic curve and two points on the curve called P and Q. ScreenOS uses the elliptic curve P-256 and sets P to be P-256's standard generator as specified in NIST Special Publication 800-90.[4] That standard also specifies the Q to use, but ScreenOS uses Juniper's own elliptic curve point Q instead. The finite field over which P-256 is defined has roughly 2^{256} elements. Points on P-256 consist of pairs of 256-bit numbers (x, y) that satisfy the elliptic curve equation. The internal state of Dual EC is a single 256-bit number s.

Let $x(\cdot)$ be the function that returns the x-coordinate of an elliptic curve point; $||$ be concatenation; $\mathrm{lsb}_n(\cdot)$ be the function that returns the least-significant n bytes of its input in big-endian order; and $\mathrm{msb}_n(\cdot)$ be the function that returns the most-significant n bytes. Starting with an initial state s_0, one invocation of Dual EC implementation generates a 32 pseudorandom byte *output* and a new state s_2 as

$$s_1 = x(s_0 P) \qquad r_1 = x(s_1 Q)$$

$$s_2 = x(s_1 P) \qquad r_2 = x(s_2 Q)$$

$$output = \mathrm{lsb}_{30}(r_1) \;||\; \mathrm{msb}_2(\mathrm{lsb}_{30}(r_2)),$$

where sP and sQ denote scalar multiplication on P-256.

In 2007, Shumow and Ferguson showed[16] that Dual EC was subject to a state reconstruction attack by an adversary who knows the value d such that $P = dQ$ and who can observe a single output value. The key insight is that multiplying the point s_1Q by d yields the internal state $x(d \cdot s_1Q) = x(s_1P) = s_2$. Although s_1Q is itself not known, 30 of the 32B of its x-coordinate (namely r_1) constitute the first 30B of *output*, and the attacker can guess the remaining bytes; the x-coordinate of an elliptic curve point determines its y-coordinate up to sign.

Assuming that the attacker knows the discrete log of Q, the major difficulty is recovering a complete output value; an attacker who only knows part of the value must exhaustively search the rest. The number of candidates grows exponentially as fewer bytes of r_1 are revealed, and recovery is intractable with fewer than about 26B. In ScreenOS, Dual EC is always used to generate 32B of output at a time, and therefore the attack is straightforward. When 30B of r_1 are available, as in Juniper's implementation, the attacker must consider 2^{16} candidate points. From the attacker's perspective, this is the optimal situation.

Importantly, as far as is publicly known, Dual EC is secure against an attacker who knows P and Q but does not know d, as recovering d would require the ability to compute discrete logarithms, which would break elliptic curve cryptography in general.

3. THE SCREENOS PRNG SUBSYSTEM

Listing 1 shows the decompiled source code for the functions implementing the PRNG in ScreenOS version 6.2.0r1; the same function is present in other releases in the 6.2 and 6.3 series. It consists of two PRNGs, Dual EC and ANS X9.31 (Appendix A.2.4; Ref.[2]).

Note that identifiers such as function and variable names are not present in the binary; we assigned these names based on our analysis of the apparent function of each symbol. Similarly, specific control flow constructs are not preserved by the compilation/decompilation process. For instance, the `for` loop on line 21 may in fact be a `while` loop or some other construct in Juniper's source code. Decompilation does, however, preserve the functionality of the original code. For clarity, we have omitted Federal Information Processing Standards (FIPS) checks that ensure that the X9.31 generator has not generated duplicate output.

A superficial reading of the `prng_generate()` function suggests that Dual EC is used only to generate keys for the X9.31 PRNG, and that it is the output of X9.31 that is returned to callers (in the `output` global buffer). The Dual EC vulnerability described in Section 2 requires raw Dual EC output, so it cannot be applied. Indeed, a 2013 knowledge base article by Juniper[8] claims exactly this. (We discuss this knowledge base article further in Section 6.)

Listing 1: The core ScreenOS 6.2 PRNG subroutines.

```
 1 char block[8], seed[8], key[24]; // X9.31 vars
 2 char output[32]; // prng_generate output
 3 unsigned int index, calls_since_reseed;
 4
 5 void prng_reseed(void) {
 6   calls_since_reseed = 0;
 7   if (dualec_generate(output, 32) != 32)
 8     error("[...] unable to reseed\n", 11);
 9   memcpy(seed, output, 8);
10   index = 8;
11   memcpy(key, &output[index], 24);
12   index = 32;
13 }
14
15 void prng_generate(void) {
16   int time[2] = { 0, get_cycles() };
17   index = 0;
18   ++calls_since_reseed;
19   if (!one_stage_rng())
20     prng_reseed();
21   for (; index <= 31; index += 8) {
22     // FIPS checks removed for clarity
23     x9_31_generate_block(time, seed, key, block);
24     // FIPS checks removed for clarity
25     memcpy(&output[index], block, 8);
26   }
27 }
```

In this reading, the `prng_reseed()` function is occasionally invoked to reseed the X9.31 PRNG state. This function invokes the Dual EC generator, directing its output to the 32B buffer `output`. From this buffer, it extracts a seed and cipher key for the X9.31 generator. With X9.31 seeded, the `prng_generate()` function generates 8B of X9.31 output at a time (line 23) into `output`, looping until it has generated 32B of output (lines 21–26). Each invocation of `x9_31_generate_block` updates the X9.31 seed state in the `seed` buffer.

The straightforward reading given above is wrong.

First, and most importantly, `index`, the control variable for the loop that invokes the X9.31 PRNG in `prng_generate()` at line 21, is a global variable. The `prng_reseed()` function, if called, sets it to 32, with the consequence that, whenever the PRNG is reseeded, `index` is already greater than 31 at the start of the loop and therefore no calls to the X9.31 PRNG are executed.[a]

Second, in the default configuration, `one_stage_rng()` always returns false, so `prng_reseed()` is always called. In the default configuration, then, the X9.31 loop is *never* invoked. (There is an undocumented ScreenOS command, `set key one-stage-rng`, that makes `one_stage_rng()` always return true; running this command induces a different PRNG vulnerability, discussed in the full version of this paper.[5])

Third, the `prng_reseed()` happens to use the `output` global buffer as a staging area for Dual EC output before it copies parts of that output to the other global buffers that hold the X9.31 seed and key. This is the same global buffer that the `prng_generate()` function was supposed to fill with X9.31 output, but fails to. When callers look for PRNG output in `output`, what they find is 32B of raw Dual EC output.

For comparison, Listing 2 shows the decompiled source code for the PRNG function in ScreenOS 6.1, before Juniper's revamp. In ScreenOS 6.1, the loop counter, `index`, is a local variable rather than a global; the X9.31 PRNG is reseeded from system entropy every 10,000 calls, instead of every call and from Dual EC; and PRNG output is placed in a caller-supplied buffer instead of a global variable.

In addition, the ScreenOS 6.1 PRNG subsystem produces 20B at a time, not 32B as in ScreenOS 6.2 and 6.3. We discuss the significance of this difference in the next section.

4. INTERACTION WITH IKE

ScreenOS implements the Internet Protocol Security (IPsec) VPN protocol. To choose the keys that protect a VPN session, the client and the ScreenOS device perform an Internet Key Exchange (IKE)[7, 11] handshake.

Listing 2: The core ScreenOS 6.1 PRNG subroutine.

```
char block[8], seed[8], key[24]; // X9.31 vars
unsigned int calls_since_reseed;

void prng_generate(char *output) {
  unsigned int index = 0;
  // FIPS checks removed for clarity
  if (calls_since_reseed++ > 9999)
    prng_reseed();
  // FIPS checks removed for clarity
  int time[2] = { 0, get_cycles() };
  do {
    // FIPS checks removed for clarity
    x9_31_generate_block(time, seed, key, block);
    // FIPS checks removed for clarity
    memcpy(&output[index], block, min(20-index,8));
    index += min(20-index, 8);
  } while (index <= 19);
}
```

In the same version 6.2 release of ScreenOS that added Dual EC (Section 2) and modified the PRNG subsystem to expose raw Dual EC output (Section 3), Juniper made a cluster of IKE implementation changes that make it possible for an attacker who knows the Dual EC secret *d* to decrypt VPN connections. In the remainder of these sections, we provide a brief description of the relevant features of IKE and then explain the impact of these changes.

4.1. Overview of IKE

IKE and its successor IKEv2 are traditional Diffie–Hellman-based handshake protocols in which two endpoints (dubbed the *initiator* and the *responder*) establish a Security Association (SA) consisting of parameters and a set of keys used for encrypting traffic. Somewhat unusually, IKE consists of two phases:

Phase 1 establishes an "IKE SA" that is tied to the end-points but not to any particular class of non-IKE network traffic. In this phase, the two sides exchange Diffie–Hellman (DH) shares and nonces, which are combined to form the derived keys. The endpoints may be authenticated in a variety of ways including a signing key and a statically configured shared secret.

Phase 2 establishes SAs that protect non-IKE traffic (typically IPsec). The IKE messages for this phase are protected with keys established in the first phase. This phase may involve a DH exchange but may also just consist of an exchange of nonces, in which case the child SA keys are derived from the shared secret established in the first phase.

IKEv2 refers to these phases as "Initial Exchange" and "CREATE_CHILD_SA," respectively; for simplicity we will use the IKEv1 Phase 1/Phase 2 terminology in the rest of this article.

An attack on IKE where ScreenOS is the responder would proceed as follows: (1) using the responder nonce in the first phase, compute the Dual EC state; (2) predict the responder's DH private key and use that to compute the DH shared secret for the IKE SA, which is used to generate the first set of keys; (3) using these traffic keys decrypt the second phase traffic to recover both initiator and responder nonces and public keys; (4) recover the responder's private key, either by running Dual EC forward (the best case scenario) or by repeating the Dual EC attack using the new responder nonce; (5) use the responder's private key and the initiator's public key to compute the shared secret for the second phase SA and

[a] The global variable reuse was first publicly noted by Willem Pinckaers on Twitter. Online: https://twitter.com/_dvorak_/status/679109591708205056, retrieved February 18, 2016.

thereby the traffic keys; and (6) use the traffic keys to decrypt the VPN traffic.

However, while this is straightforward in principle, there are a number of practical complexities and potential implementation decisions which could make this attack easier or more difficult (or even impractical) as described below.

4.2. Nonce size

For Dual EC state reconstruction to be possible, the attacker needs more than just to see raw Dual EC output. She needs at least 26B of the *x*-coordinate of a single elliptic-curve point to recover the Dual EC state; fewer bytes would be insufficient (Section 2).

Luckily for the attacker, the first 30B of the 32B returned by ScreenOS's Dual EC implementation belong to the *x*-coordinate of a single point, as we saw in Section 2. Luckily again for the attacker, ScreenOS's PRNG subsystem also returns 32B when called, and these are the 32B returned by a Dual EC invocation, as we saw in Section 3. Finally, IKE nonces emitted by ScreenOS are 32B long and produced from a single PRNG invocation. To summarize: In ScreenOS 6.2 and 6.3, IKE nonces always consist of 30B of one point's *x*-coordinate and 2B of the next point's *x*-coordinate—the best-case scenario for Shumow–Ferguson reconstruction.

It is worth expanding on this point. The IKE standards allow any nonce length between 8 and 256B (Section 5; Ref.[7]). An Internet-wide scan of IKE responders by Adrian et al.[3] found that a majority use 20B nonces. We are not aware of any cryptographic advantage to nonces longer than 20B. ScreenOS 6.1 sent 20B nonces and, as we noted in Section 3, its PRNG subsystem generated 20B per invocation. In ScreenOS 6.2, Juniper introduced Dual EC, rewrote the PRNG subsystem to produce 32B at a time, and modified the IKE subsystem to send 32B nonces.

4.3. NONCES AND DH KEYS

An attacker who knows the *d* corresponding to Juniper's point *Q* and observes an IKE nonce generated by a ScreenOS device can recompute the device's Dual EC state at nonce generation time. She can roll that state forward to predict subsequent PRNG outputs, though not back to recover earlier outputs. ScreenOS uses its PRNG to generate IKE Diffie–Hellman shares, so the attacker will be able to predict DH private keys generated after the nonce she saw and compute the session keys for the VPN connections established using those IKE handshakes.

This scenario is clearly applicable when the attacker has a network tap close to the ScreenOS device, and can observe many IKE handshakes. But what if the attacker's network tap is close to the VPN *client* instead? She might observe only a single VPN connection. If the nonce for a connection is generated after the DH share, the attacker will not be able to recover that session's keys.

A superficial reading of the ScreenOS IKE code seems to rule out single-connection attacks: The KE payload containing the DH share is indeed encoded *before* the Nr payload containing the nonce.

Conveniently for the attacker, however, ScreenOS also contains a pre-generation feature that maintains a pool of nonces and DH keys that can be used in new IKE connections, reducing handshake latency. The pooling mechanism is quite intricate and appears to be designed to ensure that enough keys are always available while avoiding consuming too much run time on the device.

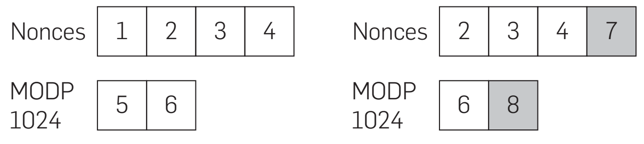
Independent First In, First Out (FIFO) queues are maintained for nonces, for each supported finite field DH group (MODP 768, MODP 1024, MODP 1536, and MODP 2048), and (in version 6.3) for each supported elliptic curve group (ECP 256 and ECP 384). The sizes of these queues depend on the number of VPN configurations that have been enabled for any given group. For instance, if a single configuration is enabled for a group then that group will have queue size of 2. The size of the nonce queue is set to be twice the aggregate size of all of the DH queues. At startup, the system fills all queues to capacity. A background task that runs once per second adds one entry to a queue that is not full. If a nonce or a DH share is ever needed when the corresponding queue is empty, a fresh value is generated on the fly.

The queues are filled in priority order. Crucially, the nonce queue is assigned the highest priority; it is followed by the groups in descending order of cryptographic strength (ECP 384 down to MODP 768). This means that in many (but not all) cases, the nonce for an IKE handshake will have been drawn from the Dual EC output stream *earlier* than the DH share for that handshake, making single-connection attacks feasible.

Figure 1 shows a (somewhat idealized) sequence of generated values, with the numbers denoting the order in which queue entries were generated, before and after an IKE Phase 1 exchange. Figure 1a shows the situation after startup: The first four values are used to fill the nonce queue and the next two values are used to generate the DH shares. Thus, when the exchange happens, it uses value 1 for the nonce and value 5 for the key, allowing the attacker to derive the Dual EC state from value 1 and then compute forward to find the DH share. After the Phase 1 exchange, which consumes a DH share and a nonce, and after execution of the periodic, queue-refill task, the state is as shown in Figure 1b, with the new values shaded.

Depending on configuration, the IKE Phase 2 exchange would consume either a nonce and a DH share or just a nonce. If the exchange uses both a nonce and a DH share, the dequeued nonce will again have been generated before

Figure 1. Nonce queue behavior during an IKE handshake. Numbers denote generation order, and values generated after the handshake are shaded. During a DH exchange, outputs 1 and 5 are used as the nonce and key, advancing the queue, and new outputs are generated to fill the end of the queue.



(a) At system startup.

(b) After a DH exchange.

the dequeued DH share. That property will continue to hold for subsequent IKE handshakes, provided that handshakes do not entirely exhaust the queues. Had the refill task not prioritized refilling the nonce queue before any DH group queue, single-connection attacks would not have been possible. Had the nonce queue been the same length as a DH share queue, single-connection attacks would not have been possible in configurations where IKE Phase 2 consumed a nonce but not a DH share.

ScreenOS 6.1 pregenerates DH shares but not nonces; the nonce queues we have described were added in ScreenOS 6.2, along with Dual EC. Had nonce queues not been added, no handshakes would have been vulnerable to single-connection decryption attacks.

In the presence of multiple nonce-only Phase-2 exchanges within a single Phase-1 exchange, multiple DH groups actively used in connections, queue exhaustion, or certain race conditions, the situation is more complicated, and it is possible for an IKE handshake phase to have its DH share generated before its nonce. Single-connection decryption attacks would fail for those handshakes. Refer to the full version of this paper for details.[5]

4.4. Recovering traffic keys

If the attacker can predict the Diffie–Hellman private key corresponding to the ScreenOS device's DH share for an IKE exchange, she can compute the DH shared secret for that exchange. With knowledge of the DH shared secret, computing the session keys used to encrypt and authenticate the VPN session being set up is straightforward, though the details depend on the IKE protocol version and the way in which the endpoints authenticate each other; for details, see the full version of this paper.[5]

For IKEv1 connections authenticated with digital signatures, the attacker knows everything she needs to compute the session keys. For IKEv1 connections authenticated with public key encryption, each peer's nonce is encrypted under the other's Rivest–Shamir–Adleman (RSA) public key, stopping the attack. IKEv1 connections authenticated with preshared keys fall somewhere in the middle: The attacker will need to know the preshared key in addition to the DH shared secret to compute the session keys. If the preshared key is strong, then the connection will still be secure. Fortunately for the attacker, many real-world VPN configurations use weak preshared keys (really passwords); in such cases having recorded an IKE handshake and recovered the DH shared secret, the attacker will be able to mount an offline dictionary attack on the preshared key. By contrast, the attacker will be able to compute session keys for IKEv2 connections in the same way, regardless of how they are authenticated.

Having computed the session keys, the attacker can decrypt and read the VPN traffic and, if she wishes, can tamper with it.

5. EXPERIMENTAL VALIDATION

To validate the attacks we describe above, we purchased a Juniper Secure Services Gateway 550M VPN device. We generated our own point Q and corresponding Dual EC secret d. We modified firmware version 6.3.0r12 to put in place our point Q, matching Dual EC Known Answer Test (KAT) values, and the (non-cryptographic) firmware checksum, and we installed the modified firmware on our device. (Our device did not have a code-signing certificate installed, so we did not need to create a valid cryptographic signature for our modified firmware.)

Using the new firmware, we configured the device with three separate VPN gateways, configured for IKEv1 with a preshared key, IKEv1 with a 1024-bit RSA signing certificate, and IKEv2 with a preshared key, respectively. We made connections to each gateway using the strongSwan VPN software as our initiator and recorded the traffic to our device. We successfully decrypted each connection by recovering the Dual EC state and traffic keys using just that connection's captured packets.

6. HISTORY OF THE JUNIPER INCIDENT

The history of the Juniper incident begins nearly a decade ago.[b] In October 2008, Juniper released ScreenOS 6.2. As described in detail above, this release (1) replaced an entropy-gathering procedure for (re)seeding the ANS X9.31 PRNG with Dual EC using a custom Q point; (2) modified the X9.31 reseed logic to reseed on every call rather than every ten thousand calls; (3) changed the loop counter in the `prng_generate` procedure as well as the procedure's output to be global variables, shared with the reseed procedure, thus ensuring that pseudorandom values are generated by Dual EC, and not X9.31; (4) changed the IKE nonce length from 20B to 32B; and (5) added a nonce pregeneration queue.

The result of the first four changes is that whoever knew the integer d corresponding to Juniper's Q could passively decrypt (some) VPN traffic. Each of the first four changes is critical to the attack described in this article. The fifth change enables single-connection attacks in many cases, but is not necessary for multi-connection attacks.

This state of affairs continued for four years. At some point prior to the release of ScreenOS 6.2.0r15 (September 2012) and ScreenOS 6.3.0r12 (August 2012), someone modified Juniper's source code. Based on the patched firmware revisions Juniper would later release, the modifications were quite small: The x-coordinate of Juniper's Dual EC's Q was changed as was the expected response to Dual EC's Known Answer Test. As a result, the set of people who could passively decrypt ScreenOS's VPN traffic changed from those who know Juniper's d (if any) to those who know the new d corresponding to the changed Q (presumably the attacker who made the change).

Apparently unrelated to the 2012 changes, a second source code modification was made. A hard-coded SSH and Telnet password was inserted into Juniper's code at some point before the release of ScreenOS 6.3.0r17 (April 2013). Logging in with this password yields administrator access.

[b] The dates in this section come from file dates, ScreenOS release notes, and Juniper's website, none of which agree precisely on *any* dates.

In early September 2013, the New York Times published an article based on documents from Snowden strongly implying that the National Security Agency (NSA) had engineered Dual EC to be susceptible to attack.[15] The article does not name Dual EC; it instead refers to a 2006 NIST standard with a "fatal weakness, discovered by two Microsoft cryptographers in 2007," presumably referring to Dan Shumow and Niels Ferguson's presentation at CRYPTO 2007.[16] This reporting led NIST to withdraw its recommendation for Dual EC.[14]

After NIST withdrew its recommendation, Juniper subsequently published a knowledge base article explaining their use of Dual EC in ScreenOS.

> ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 [*sic*] PRNG, which is the random number generator used in ScreenOS cryptographic operations.[8]

The first mitigation—using self-generated basis points—only defends against the attacks described in this paper if Q is generated so that nobody knows d; Juniper has provided no evidence that this is the case. As we describe in Section 3, Juniper's claim that the output of Dual EC is only used as an input to X9.31 is incorrect.

This was the situation on December 17, 2015 when Juniper issued an out-of-cycle security bulletin[9] for two security issues in ScreenOS: CVE-2015-7755[c] ("Administrative Access") and CVE-2015-7756[d] ("VPN Decryption").

This announcement was particularly interesting because it was not the usual report of developer error, but rather of malicious code which had been inserted into ScreenOS by an unknown attacker:

> During a recent internal code review, Juniper discovered unauthorized code in ScreenOS that could allow a knowledgeable attacker to gain administrative access to NetScreen® devices and to decrypt VPN connections. Once we identified these vulnerabilities, we launched an investigation into the matter, and worked to develop and issue patched releases for the latest versions of ScreenOS.[10]

The bulletin prompted a flurry of reverse-engineering activity around the world, including by our team. The "Administrative Access" issue was quickly identified as the 2013 source code modification. This issue has been extensively discussed by Moore.[13] Our analysis of the "VPN Decryption" issue, described in this article, shows that the 2012 code modification is responsible.

Our analysis implies several items of note. First, the 2012 code modification indicates that Juniper's 2013 knowledge base article[8] is incorrect when it states that ScreenOS uses Juniper's own Q point since, at that time, ScreenOS was shipping with the attacker's Q. Second, by the end of 2015, Juniper knew that Dual EC could be exploited in ScreenOS. Despite this, Juniper's initial fix was to revert the Q point to their initial value in each affected ScreenOS revision. Eventually, after press coverage of our results, Juniper committed to removing Dual EC from their PRNG subsystem.

7. EXCEPTIONAL ACCESS AND NOBUS

Law enforcement officials have been warning since 2014 that they are "going dark": that ubiquitous end-to-end encryption threatens investigations by rendering intercepted communications unreadable. They have called on technology companies to rearchitect their products so intercepted communications could be decrypted given a court order. Computer scientists have resisted such "exceptional access" mandates, arguing that whatever mechanism implements it would constitute a vulnerability that might be exploited by third parties.[1]

Attempts to design exceptional access mechanisms which do not introduce vulnerabilities go back at least as far as 1993, when the NSA introduced "Clipper," an encryption algorithm embedded in a hardware platform with a built-in "key escrow" capability, in which cryptographic keys were separately encrypted under a key known to the US government. Such a mechanism would be "NOBUS," in the jargon of the NSA, for "nobody but us" (p. 281; Ref.[6]): data would be cryptographically secure against anyone who did not have the keys but transparent to those who did.

While the key escrow mechanism designed for Clipper involved encrypting the traffic keys under the escrow key, it is also possible to build an exceptional access mechanism around a system like Dual EC, with the escrow key being the discrete log of Q. The common thread here is that the *key* is intended to be known only to authorized personnel.

Whatever the intent of Juniper's selection of Dual EC, its use created what was in effect an exceptional access system: one where the key was the d value corresponding to Juniper's choice of Q. We have no way of knowing whether anyone knew that d value or not, and Juniper has not described how they generated Q. However, around 2012, some organization gained the ability to make changes to Juniper's source code repository. They used that access to change the Dual EC point Q to one of their choosing, in essence swapping out the escrow key. Between September 2012 and December 2015, official releases of ScreenOS distributed by Juniper included the intruders' point Q instead of Juniper's. VPN connections to NetScreen devices running affected releases were subject to decryption by the intruders, assuming they know the d corresponding to their point Q.

8. LESSONS

The ScreenOS vulnerabilities we have studied provide important broader lessons for the design of cryptographic systems, which we summarize here.

[c] https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-7755

[d] https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-7756

[e] Of course, reducing nonce size cannot prevent all data exfiltration strategies. However, it may increase the difficulty of hiding the necessary code, and the complexity of executing an attack.

8.1. For protocol designers

Allowing nonces to vary in length, and in particular to be larger than necessary for uniquely identifying sessions, may be a bad idea. The authors are unaware of any cryptographic rationale for 256B nonces, as permitted by IPsec; it is simply an invitation for implementations to disclose sensitive state, intentionally or not.[e]

Adding even low-entropy shared secrets as key derivation inputs helps protect against entropy failures. We observe a difference in exploitability of the ScreenOS bugs between IKEv1 and IKEv2 that is entirely due to the different use of the preshared key between the two protocols. It is unfortunate that IKEv2 is easier to exploit.

8.2. For implementers and code reviewers

Cryptographic code must be locally auditable: It must be written in such a way that examining a function or a module in isolation allows the reader to understand its behavior.

ScreenOS's implementation failed to live up to this guideline. A loop counter in the core `prng_generate` routine was defined as a global variable and changed in a subroutine. This is a surprising-enough pattern that several experienced researchers who knew that the routine likely had a bug failed to spot it before Willem Pinckaers' contribution. The `prng_generate` and `prng_reseed` routines reuse the same 32B buffer, `output`, for two entirely different purposes: Dual EC output with which to seed X9.31, and output from the PRNG subsystem. ScreenOS's use of pregeneration queues makes it difficult to determine whether nonces or Diffie–Hellman shares are generated first. Someone reading the code for the top-level functions implementing IKE in isolation will conclude that Diffie–Hellman shares are generated first, whereas in practice the opposite is usually the case.

The state recovery attacks suffered by Juniper suggest that implementations may wish to avoid revealing the raw output of a random number generator entirely, perhaps by hashing any PRNG output before using it as a nonce. One could also design implementations so that separate PRNGs are used for different protocol components, to separate nonce security from key security.

Several of the above mistakes represent poor software engineering practices. Cryptographic code reviews, whether internal or external (e.g., for FIPS validation), should take code quality into account.

8.3. For NIST

Juniper followed then-current best practices in designing and verifying their random number generators. They used a NIST-certified algorithm, followed the FIPS-recommended procedure to verify the output using test vectors, and followed a commonly recommended engineering guideline to use a PRNG as a whitener for a potentially insecure random number generator, removing—at least in theory—the structured output that makes Dual EC vulnerable.

In this case, all three approaches failed. In particular, a crippling defect in the whitening countermeasure managed to go undetected in FIPS certification. This suggests potential future work for research in the verification of cryptographic systems. One step would be to track the origin and use of any buffers—especially shared buffers—and enforce a rule that all random number generator output can be traced back to an appropriate cryptographic function, such as a block cipher or hash. Some form of coverage analysis might also have revealed that the whitening is never performed.

To the extent that FIPS guidelines mandate the use of global state, they run counter to our suggestion, above, that cryptographic code be locally auditable.

Products are evaluated against FIPS standards by accredited laboratories. ScreenOS was FIPS certified with the X9.31 PRNG, yet the lab evaluating ScreenOS failed to spot that X9.31 was never invoked, as well as failing to detect the defect in the Dual EC implementation described in Section 3. NIST should revisit its laboratory accreditation program to ensure more thorough audits, especially of randomness subsystem code.

8.4. For attackers

The choice by the attacker to target the random number generation subsystem is instructive. Random number generators have long been discussed in theory as a target for kleptographic substitution attacks,[18] but this incident tells us that the threat is more real than has been known in the academic literature.

From the perspective of an attacker, by far the most attractive feature of the ScreenOS PRNG attack is the ability to significantly undermine the security of ScreenOS *without* producing any externally detectable indication that would mark the ScreenOS devices as vulnerable. This is in contrast to previous well-known PRNG failures, which were externally observable, and, in the case of the Debian PRNG flaw,[17] actually detected through observational testing. Indeed, the versions of ScreenOS containing an attacker-supplied parameter appear to have produced output that was cryptographically indistinguishable from the output of previous versions, thus preventing any testing or measurement from discovering the issue.

8.5. For journalists

Much of the coverage of the Juniper disclosure has focused on the unauthorized changes made in 2012 to the randomness subsystem and in 2013 to the login code. By contrast, our forensic investigation of ScreenOS releases highlights the changes made in the 6.2 series, in 2008, as the most consequential.

These changes, which introduced Dual EC and changed other subsystems in such a way that an attacker who knew the discrete log of Q could exploit it, were, as far as we know, added by Juniper engineers, not by attackers. This raises a number of questions:

How was the new randomness subsystem for the ScreenOS 6.2 series developed? What requirements did it fulfill? How did Juniper settle on Dual EC? What organizations did it consult? How was Juniper's point Q generated?

[f] Online: https://oversight.house.gov/hearing/federal-cybersecurity-detection-response-and-mitigation/.

We are not able to answer these questions with access to firmware alone. Juniper's source code version-control system, their bug-tracking system, their internal e-mail archives, and the recollections of Juniper engineers may help answer them.

Despite numerous opportunities, including public questions put to their Chief Security Officer and a congressional hearing on this incident,[f] Juniper has either failed or explicitly refused to provide any further details.

8.6. For policymakers

Much of the debate about exceptional access has focused on whether it is possible to construct secure exceptional access mechanisms, where "secure" is defined as only allowing authorized access—presumably by law enforcement. It is readily apparent that one of the major difficulties in building such a system is the risk of compromise of whatever keying material is needed to decrypt the targeted data.

The unauthorized change to ScreenOS's Dual EC constants made in 2012 illustrates a new threat: the ability for another party to modify the target software to subvert an exceptional access mechanism for its own purposes, with only minimally detectable changes. Importantly, because the output of the PRNG appears random to any entity that does not know the discrete log of Q, such a change is invisible both to users and to any testing which the vendor might do. By contrast, an attacker who wants to introduce an exceptional access mechanism into a program which does not already has one must generally make a series of extremely invasive changes, thus increasing the risk of detection.

In the case of ScreenOS, an attacker was able to subvert a major product—one which is used by the federal government—and remain undiscovered for years. This represents a serious challenge to the proposition that it is possible to build an exceptional access system that is available only to the proper authorities; any new proposal for such a system should bear the burden of proof of showing that it cannot be subverted in the way that ScreenOS was.

Acknowledgments

This material is based in part upon work supported by the U.S. National Science Foundation under awards EFMA-1441209, CNS-1505799, CNS-1010928, CNS-1408734, and CNS-1410031; The Mozilla Foundation; a gift from Cisco; and the Office of Naval Research under contract N00014-14-1-0333.

References

1. Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Green, M., Landau, S., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B., Specter, M., Weitzner, D.J. Keys under doormats: Mandating insecurity by requiring government access to all data and communications. *Commun. ACM 58*, 10 (Oct. 2015), 24–26.
2. Accredited Standards Committee (ASC) X9, Financial Services. ANS X9.31-1998: Digital signatures using reversible algorithms for the financial services industry (rDSA), 1998. Withdrawn.
3. Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J.A., Heninger, N., Springall, D., Thomé, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Béguelin, S., Zimmermann, P. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proceedings of CCS 2015*. C. Kruegel and N. Li, eds. ACM Press, New York, NY, Oct. 2015, 5–17.
4. Barker, E., Kelsey, J. NIST Special Publication 800-90: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Technical report, National Institute of Standards and Technology, June 2006.
5. Checkoway, S., Maskiewicz, J., Garman, C., Fried, J., Cohney, S., Green, M., Heninger, N., Weinmann, R.-P., Rescorla, E., Shacham, H. A systematic analysis of the Juniper Dual EC incident. In *Proceedings of CCS 2016*. S. Halevi, C. Kruegel, and A. Myers, eds. ACM Press, New York, NY, Oct. 2016, 468–479.
6 Granick, J.S. *American Spies: Modern Surveillance, Why You Should Care, and What To Do About It*. Cambridge University Press, Cambridge, 2017.
7. Harkins, D., Carrel, D. The Internet Key Exchange (IKE). RFC 2409 (Proposed Standard), Nov. 1998. Obsoleted by RFC 4306, updated by RFC 4109. Online: https://tools.ietf.org/html/rfc2409.
8. Juniper Networks. Juniper Networks product information about Dual_EC_DRBG. Knowledge Base Article KB28205, Oct. 2013. Online: https://web.archive.org/web/20151219210530/ https://kb.juniper.net/InfoCenter/index?page= content&id=KB28205&pmv=print&actp=LIST.
9. Juniper Networks. 2015-12 Out of Cycle Security Bulletin: ScreenOS: Multiple Security issues with ScreenOS (CVE-2015-7755, CVE-2015-7756), Dec. 2015.
10. Juniper Networks. Important announcement about ScreenOS®. Online: https://forums.juniper.net/t5/Security-Incident-Response/Important-Announcement-about-ScreenOS/ba-p/285554, Dec. 2015.
11. Kaufman, C. Internet Key Exchange (IKEv2) Protocol. RFC 4306 (Proposed Standard), Dec. 2005. Obsoleted by RFC 5996, updated by RFC 5282. Online: https://tools.ietf.org/html/rfc4306.
12. Kelsey, J. Dual EC in X9.82 and SP 800-90A. Presentation to NIST VCAT committee, May 2014. Slides online http://csrc.nist.gov/groups/ST/crypto-review/documents/dualec_in_X982_and_sp800-90.pdf.
13. Moore, H.D. CVE-2015-7755: Juniper ScreenOS Authentication Backdoor. https://community.rapid7.com/community/infosec/blog/2015/12/20/cve-2015-7755-juniper-screenos-authentication-backdoor, Dec. 2015.
14. National Institute of Standards and Technology. NIST opens draft Special Publication 800-90A, recommendation for random number generation using deterministic random bit generators for review and comment. http://csrc.nist.gov/publications/nistbul/itlbul2013_09_supplemental.pdf, Sept. 2013.
15. Perlroth, N., Larson, J., Shane, S. N.S.A. able to foil basic safeguards of privacy on Web. *The New York Times*, Sep. 5 2013. Online: http://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html.
16. Shumow, D., Ferguson, N. On the possibility of a back door in the NIST SP800-90 Dual Ec Prng. Presented at the Crypto 2007 rump session, Aug. 2007. Slides online: http://rump2007.cr.yp.to/15-shumow.pdf.
17. Yilek, S., Rescorla, E., Shacham, H., Enright, B., Savage, S. When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In *Proceedings of IMC 2009*. A. Feldmann and L. Mathy, eds. ACM Press, New York, NY, Nov. 2009, 15–27.
18. Young, A., Yung, M. Kleptography: Using cryptography against cryptography. In *Proceedings of Eurocrypt 1997*. W. Fumy, ed. volume 1233 of *LNCS*, Springer-Verlag, May 1997, 62–74.

Stephen Checkoway, University of Illinois at Chicago, IL, USA.

Jacob Maskiewicz, Eric Rescorla, Hovav Shacham, University of California, San Diego, CA, USA.

Christina Garman, Matthew Green, Johns Hopkins University, Baltimore, MD, USA.

Joshua Fried, Shaanan Cohney, Nadia Heninger, University of Pennsylvania, Philadelphia, PA, USA.

Ralf-Philipp Weinmann, Comsecuris, Duisberg, Germany.

DOI:10/1145/3266285

To view the accompanying paper, visit doi.acm.org/10.1145/3266287

Technical Perspective
Making Sleep Tracking More User Friendly

By Tanzeem Choudhury

AN EXCITING AREA of research in mobile and ubiquitous computing is the recent development of novel sensing systems capable of continuously tracking behavioral and physiological signals from individuals in their natural environment. Often referred to as digital biomarkers, these signals capture people's everyday routines, actions, and physiological changes that can explain outcomes related to health, cognitive abilities, and more.

A key behavioral biomarker is sleep, which is essential for human health, learning, cognitive abilities, and brain development. About one-third of adults suffer from some form of sleep disorder. However, current sleep-tracking options are mostly restricted to special sleep clinics or hospitals, where individuals are removed from their natural sleep environment and undergo polysomnography (PSG) that monitors brain activity via electroencephalography (EEG), eye movement via electrooculography (EOG), and muscle activity using electromyography (EMG).

These bio-signals are combined to infer sleep duration, sleep quality, and stages of sleep (light, deep, and REM sleep). Beyond sleep, the ability to track these signals can lead to new types of brain-computer interfaces and the detection of alertness and interests, eating moments, autism onset, and more.

In the recent years, researchers in ubiquitous and mobile computing have pushed the boundaries of sensing digital biomarkers, and have created novel systems that can track important markers in the real world. Crucially, these systems are unobtrusive, meaning they do not require removing individuals from their natural environment or to be burdened with a bulky sensing setup in order to get reliable measurements. One of the main challenges in this domain is to balance the fidelity and accuracy of signals in the presence of natural usage variations with the burden that is placed on the users. Existing sleep and bio-signal sensing solutions that measure EEG, EOG, or EMG include wearable headbands, eyemasks, smart shirts, and wristbands that are not as cumbersome as PSG and are much less expensive, but still are not comfortable enough for individuals to use continuously over extended periods of time.

There has also been much work related to developing sleep trackers that leverage sensor signals and usage patterns from smartphones so that they require minimal effort from users. Such systems have been used to estimate sleep duration, interruptions, and even chronotype. More recently, contactless sensing approaches that use WiFi or Doppler Radar measures have been used to detect specific sleeping problems, such as sleep apnea or more general sleep quality, and sleep stages based on changes in breathing rate, heart rate, and movement. These contactless and smartphone-based methods place very low or no burden on the user. Moreover, they are generally reliable for coarser features of sleep such as duration and interruptions. However, they are less reliable for getting finer-grained information about sleep stages.

The Lightweight In-ear BioSensing (LIBS) system work detailed in the following paper provides a nice balance in terms of minimizing the burden on users and the granularity at which we can automatically track various measures of sleep. The custom flexible electrodes wrapped around off-the-shelf foam earplugs are able to pick up multiple signals of interest (EEG, EOG, EMG) from the ear canal, and are designed for user comfort, allowing individuals to sleep more naturally. More importantly, they can be used at home in a user's normal bedroom setting.

The LIBS approach is a great example of how to trade off signal quality and user burden.

Using signal separation and classification algorithms, the LIBS system can pull out the different bio-signals from the single mixed in-ear channel, derive relevant spectral and temporal features from these signals, and classify stages of sleep. The LIBS approach is a great example of how to trade off signal quality and user burden. Systems that focus on designing low-burden sensing and that push the boundary in terms of sensing granularity and resolution can significantly increase the adoption of mobile and ubiquitous solutions in the real world, especially in the realm of healthcare where fidelity and accuracy of the measures are important. This is even more important if diagnosis and treatment decisions are to be made based on the measurements.

LIBS is an elegant engineering solution to overcome real-world usability barriers, it is accurate, and it provides a reliable alternative to highly intrusive PSG-based measures. Of course, in order to assess how broadly applicable the system is will require more longitudinal testing across variable sleep environments and across people with different sleeping habits and in diverse age groups. Nonetheless, LIBS takes a significant step in the right direction, and is sure to inspire more solutions that can effectively balance accuracy and usability.

Tanzeem Choudhury (tanzeem.choudhury@cornell.edu) is an associate professor of information science and director of the People-Aware Computing group at Cornell University, Ithaca, NY, USA.

DOI:10.1145/3266287

LIBS: A Bioelectrical Sensing System from Human Ears for Staging Whole-Night Sleep Study

By Anh Nguyen, Raghda Alqurashi, Zohreh Raghebi, Farnoush Banaei-Kashani, Ann C. Halbower, and Tam Vu

Abstract

Sensing physiological signals from the human head has long been used for medical diagnosis, human-computer interaction, meditation quality monitoring, among others. However, existing sensing techniques are cumbersome and not desirable for long-term studies and impractical for daily use. Due to these limitations, we explore a new form of wearable systems, called *LIBS*, that can continuously record biosignals such as brain wave, eye movements, and facial muscle contractions, with high sensitivity and reliability. Specifically, instead of placing numerous electrodes around the head, *LIBS* uses a minimal number of custom-built electrodes to record the biosignals from human ear canals. This recording is a combination of three signals of interest and unwanted noise. Therefore, we design an algorithm using a supervised Nonnegative Matrix Factorization (NMF) model to split the single-channel mixed signal into three individual signals representing electrical brain activities (EEG), eye movements (EOG), and muscle contractions (EMG). Through prototyping and implementation over a 30 day sleep experiment conducted on eight participants, our results prove the feasibility of concurrently extracting separated brain, eye, and muscle signals for fine-grained sleep staging with more than 95% accuracy. With this ability to separate the three biosignals without loss of their physiological information, *LIBS* has a potential to become a fundamental in-ear biosensing technology solving problems ranging from self-caring health to non-health and enabling a new form of human communication interfaces.

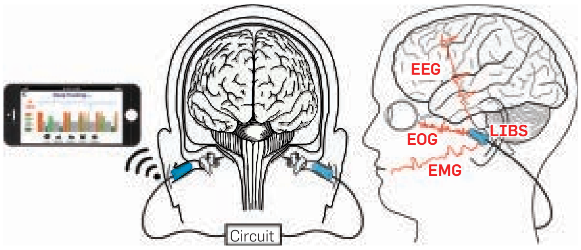
1. INTRODUCTION

Physiological signals generated from human brain, eye, and facial muscle activities can reveal enormous insight into an individual's mental state and bodily functions. For example, acquiring these biosignals is critical to diagnose sleep quality for clinical reasons, among other auxiliary signals. Even though providing highly reliable brain signal Electroencephalography (EEG), eye signal Electrooculography (EOG), and muscle signal Electromyography (EMG), the gold-standard methodology, referred to as Polysomnography (PSG),[9] has many limitations. Specifically, PSG attaches a large number of wired electrodes around human head, requires an expert sensor hookup at a laboratory, and provides a risk of losing sensor contact caused by body movements during sleep. Consequently, this gold-standard approach is uncomfortable, cumbersome to use, and expensive and time-consuming to set up.

As an effort to overcome the inherent limitations of PSG, there exist various wearable solutions developed to acquire the biosignals with high resolution and easy self-applicability. They involve electrode caps, commercial head-worn devices (e.g., EMOTIV, NeuroSky MindWave, MUSE, Kokoon, Neuroon Open, Aware, Naptime, Sleep Shepherd, etc.), and hearing aid-like research devices.[6, 10] However, these solutions are stiff, unstable, and only suitable for either short-term applications or in-hospital use. In other words, they are still inconvenient and less socially acceptable for outdoor, long-term, and daily activities.

To fill in this gap, we propose a Light-weight In-ear BioSensing (LIBS) system that can continuously record the electrical activities of human brain, eyes, and muscles concurrently using a minimum number of passive electrodes placed invisibly in the ear canals. In this work, particularly, the idea of sensing inside human ears has been motivated from the fact that the ear canals are reasonably close to all sources of the three biosignals of interest (i.e., EEG, EOG, and EMG signals) as shown in Figure 1. Furthermore, physical features of the ear canal allow a tight and fixed sensor placement, which is desirable for electrode stability and long-term wearability. Hence, we carefully develop *LIBS* using very flexible, conductive electrodes to maximize the quality of its contact area with the skin in the wearer's ear canals for good signal acquisition while maintaining a high level of comfort.

Figure 1. Conceptual illustration of *LIBS* and its relative position to the sources of EEG, EOG, and EMG signals.



(a) Conceptual LIBS

(b) LIBS towards signal sources

The original version of this paper is entitled "A Lightweight And Inexpensive In-ear Sensing System For Automatic Whole-night Sleep Stage Monitoring" and was published in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2016, ACM, New York, NY, USA.

However, as minimizing the number of used electrodes, we can achieve only the single-channel signal, which is a mixture of EEG, EOG, EMG signals, and unwanted noise. We then develop a signal separation model for *LIBS* to extract the three signals of interest from the in-ear mixed signal. To validate the lossless of essential physiological information in the separated signals acquired by *LIBS*, we finally develop a sleep stage classification algorithm to score every 30sec epoch of the separated signals into an appropriate stage using a set of discriminative features obtained from them. Through the hardware prototype and a one-month long user study, we demonstrated that the proposed *LIBS* was comparable to the existing dedicated sleep assessment system (i.e., PSG) in terms of accuracy.

Due to the structural variation across ear canals and overlapped characteristics of the EEG, EOG, and EMG signals, building *LIBS* is difficult because of three following key reasons. (1) The brain signal is quite small in order of micro-Volts (μV). Additionally, the human head anatomy shown in Figure 1(b) indicates that their sources are not too close to the location of *LIBS* placed in the ear canals to be sensed, especially in case of the weak brain source, (2) The characteristics of those three biosignals are overlapped in both time and frequency domains. Moreover, their activation is random and possibly simultaneous during the monitoring period, and (3) The signal quality is easily varied by the displacement of electrodes across device hookups and the variation of physiological body conditions across people. Consequently, our first challenge is to build sensors capable of providing a high level of sensitivity while recording the biosignals from afar and comfort while wearing the device. Our second challenge is then to provide a robust separation mechanism in the presence of multiple variances, which becomes a significant hurdle.

While addressing the above challenges to realizing *LIBS*, we make the following contributions through this work:

1. Developing a light-weight and low-cost earplug-like sensor with highly sensitive and soft electrodes, the whole of which is comfortably and safely placed inside human ears to continuously measure the voltage potential of the biosignals in long term with high fidelity.
2. Deriving and implementing a single-channel signal separation model, which integrates a process of learning source-specific prior knowledge for adapting the extraction of EEG, EOG, and EMG from the mixed in-ear signal to suit the variability of the signals across people and recordings.
3. Developing an end-to-end sleep staging system, which takes the input of three separated biosignals and automatically determines the appropriate sleep stages, as a proof-of-concept of *LIBS*'s potential in reality.
4. Conducting an over 30 day long user studies with eight subjects to confirm the feasibility and learn the usability of LIBS.

2. LIBS'S SYSTEM OVERVIEW

In this section, we present an overall design of *LIBS* in order to achieve the EEG, EOG, and EMG signals individually from the in-ear mixed biosignal. Additionally, we provide a module that automatically determines appropriate sleep stages from *LIBS*'s outputs acquired in sleep studies as its application. Generally, the whole-night sleep staging system, as illustrated in Figure 2, consists of three following primary modules.

2.1. Signal acquisition

Overall, this module focuses on tackling our first challenge that requires (1) an ability to adapt to the small uneven area inside human ear and its easy deformability under the jaw movements (e.g., teeth grinding, chewing, and speaking), (2) a potential to acquire the naturally weak biosignals, which have micro-Volt amplitude, and (3) a provision of comfortable and harmless wearing to the users. We fulfill these obstacles by firstly custom-making a deformable earplug-like sensors using a viscoelastic material with atop sensitive electrodes using several layers of thin, soft, and highly conductive materials. To possibly capture the weak biosignals from inside human ears, we then increase the distance between the main electrodes and the reference point to further enhance signal fidelity. Finally, we preprocess the collected signal to eliminate signal interference (e.g., body movement artifact and electrical noise).

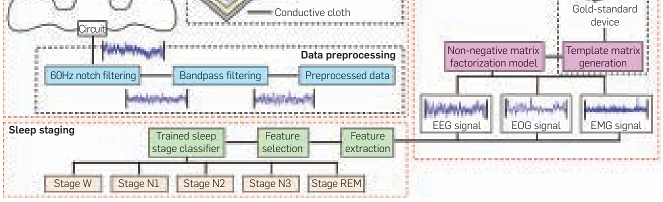
2.2. In-ear mixed signal separation

In this module, we form a supervised algorithm to overcome our second challenge for signal separation. This challenge, in detail, is related to (1) overlapping characteristics of three signals in both time and frequency domains, (2) a random activation of the sources generating them, and (3) their variation from person to person and in different recordings. We solve these problems by developing a supervised Nonnegative Matrix Factorization (NMF)-based model that can separate the preprocessed in-ear mixed signal into EEG, EOG, and EMG with high similarity to the ground truth given by the gold-standard device. Specifically, our separation algorithm initially learns prior knowledge of the biosignals of interest through their individual spectral templates. It then adapts the templates to the variation between people through a deformation step. Hence, the model we build can alter itself slightly to return the best fit between the expected biosignals and the given templates.

2.3. Automatic sleep staging

This last module provides a set of machine learning algorithms to continuously score sleep into appropriate sleep stages using EEG, EOG, and EMG separated from the in-ear mixed signal. Because those signals can have similar

Figure 2. *LIBS* architecture and its sleep staging application.



characteristics shared in some of stages, this module is challenging by an ability to (1) find the most informative and discriminative features describing all three biosignals when they are used together and then (2) construct an efficient classifier to perform sleep staging. We introduce a classification model that can automatic score the sleep after well trained. Firstly, we deploy an off-line training stage composing of three steps: feature extraction, feature selection, and model training. Specifically, a set of possible features corresponding to each of three separate signals are extracted. Next, a selection process is applied to choose features with a more discriminative process. Using a set of dominant features selected, the sleep stage classifier is trained with a measurement of similarity. Finally, the trained model is used in its second stage for on-line sleep stage classification.

3. IN-EAR MIXED SIGNAL ACQUISITION

In this section, we discuss the anatomical structure of human ears that leads to the custom design of *LIBS* sensor as well as its actual prototype using off-the-shelf electrical components.

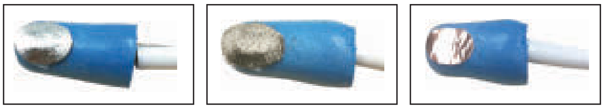
3.1. Sensor materials

Extensive anatomical study of human ears shows that the form of ear canal is easy to be affected when the jaw moves.[15] More remarkably, a person can have asymmetry between his left and right ears.[14] Beyond those special characteristics, to capture the good signals, it is important to eliminate a gap between the electrodes and human skin due to the nature of the ion current generated by the biosignals. Hence, *LIBS* sensor needs to flexibly reshape itself, well contact to the skin, well fit different ear structures and types of muscle contractions, and comfortably be worn in long term. One possible approach is to personalize a mold. However, this approach entails high cost and time consume. Therefore, a commercial earplug with noise-cancelled and flexible wires are offered to form the sensor prototype. Specifically, we have augmented an over-the-counter sound block foam earplug for its base. The soft elastic material (or memory foam) of the earplug enables the sensor to reshape to its original form shortly after being squeezed or twisted under the strain to insert into the ear. This fundamental property of the foam earplug provides a comfortable and good fit as it allows the sensor to follow the shape of the inner surface in the ear canal. In addition, it not only supplies a stable contact between the electrodes and the in-ear skin but reduces the motion artifact caused by jaw motion as well. Moreover, using the earplug completely eliminate the personalization of the base regarding the canal size. As an additional bonus, the soft surface and the lightweight property of the earplug make itself more convenient to be worn without much interference and to block out noise during sleep for our case study.

3.2. Electrode construction and placement

On the other hand, *LIBS* needs to possibly measure low-amplitude biosignals from a distance with high fidelity. Our method integrates several solutions into the hardware design to address this demand. We firstly tried different conductive materials as shown in Figure 3. However, our experiment resulted that copper is a hard material to be inserted into and placed inside the ear without harm. Oppositely, conductive fabric is a good choice that neither harms the in-ear skin nor is broken while being squeezed. However, because of the weave pattern of its fibers, which cases a non-identical resistivity (19Ω/sq) on the surface, we further coat their surface with many layers of thin pure silver leaves, which gives low and consistent surface resistance for providing reliable signals. Also, a very small amount of health-grade conductive gel is added. In Figure 2, the construction module shows the comprehensive structure of *LIBS* electrodes. Ultimately, we place the active and reference electrodes in two separate ear canals, hence intensify the potential of the signals by a distance increase. Finally, the recorded signal is transferred to an amplifier through shielded wires to prevent any external noise.

Figure 3. Prototypes with different conductive materials.



(a) Silver coated fabric electrode **(b)** Fabric electrode **(c)** Copper electrode

3.3. Microcontroller

In this prototype, we use a general brain-computer interface board manufactured by OpenBCI[16] group to sample and digitize the signal. The board is supplied by a battery source of 6V for safety and configured at a 2kHz sampling rate and a 24dB gain. The signal is stored in an on-board mini-SD card while recording and then processed offline in a PC.

4. NMF-BASED SIGNAL SEPARATION

Due to the limited cavity of the ear canal, the biosignal recorded by *LIBS* is inherently a single-channel mixture of at least four components including EEG, EOG, EMG signals, and unwanted noise. We assume that the mixed signal is a linear combination of aforementioned signals generated from a number of individual sources in the spectral domain,[7] which we mathematically express in Equation (1).

$$X = \sum_{i=1}^{3} w_i s_i + \epsilon \tag{1}$$

where s_i is the power spectrum of the three biosignals with their corresponding weight w_i and ϵ represents noises.

Generally, the problem of separating original signals from their combinations generated by concurrent multi-source activation has long been addressed for different systems. The classical example of this problem is the auditory source separation problem, also called a cocktail party problem, where various algorithms have been developed to extract individual voices of a number of people talking simultaneously in a room. Additionally, the problem of decoding a set of received signals to retrieve the orginal signals transmitted by multiple antennas via Multi-Input and Multi-Output (MIMO)[22] in wireless communication can also be another example. Although there exist mainstream techniques[25] such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), Empirical Mode Decomposition (EMD), Maximum Likelihood Estimation (MLE), and Nonnegative Matrix Factorization (NMF) built to solve the blind source separation problem, most of them require that (1) the number of collected channels is equal to or larger than the number of source signals (*except NMF*) and (2) the factorized components describing the source signal are known or selected manually. As a result, it is impossible to directly apply them in our work since their first constraint conflicts with the fact that *LIBS* has only one channel, which is fewer than the number of signals of interest (three signals).

To successfully address this challenge, we propose a novel source separation technique that takes advantage of NMF. However, there have existed two potential issues with a NMF-based model that might degrade the quality of the decomposed signals. They include (1) the inherent non-unique estimation of the original source signals (ill-posed problem) caused by the non-convex solution space of NMF and (2) the variance of the biosignals on different recordings. To solve them, our proposed NMF-based model is combined with source-specific prior knowledge learnt in advance for each user through a training process. Figure 4 demonstrates the high-level overview of this process, which leverages two different NMF techniques to learn source-specific information and to separate the mixing in-ear signal based on priory training.

Particularly, when a new user starts using *LIBS*, his groundtruth EEG, EOG, and EMG are shortly acquired using the gold-standard device (i.e., PSG) and fed into a single-class Support Vector Machine (SVM)-based NMF technique (SVM-NMF)[3] to build a personal spectral template matrix, called W, representing their basis patterns. Then, for any in-ear signal $\tilde{X}$ recorded by *LIBS*, our trained model approximately decomposes its power spectrum X into two lower rank nonnegative matrices

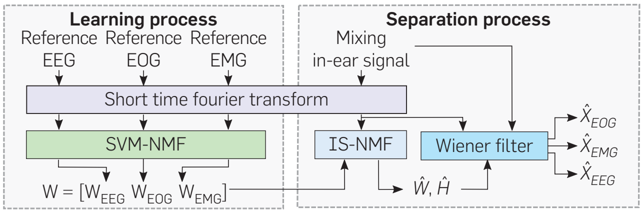
$$X \simeq WH \tag{2}$$

in which $X \in \mathfrak{R}^{m \times n}$ comprises m frequency bins and n temporal frames; W is calculated in advance and given; and H is the activation matrix expressing time points (positions) when the signal patterns in W are activated. Finding the best representative of both W and H is equivalent to minimizing a cost function defined by the distance between X and WH in Equation (3).

$$\left\{\hat{W}, \hat{H}\right\} = \underset{W,\, H \geq 0}{\arg\min}\; d(X \| W H) \tag{3}$$

We solve Equation (3) using multiplicative update rules to achieve a good compromise between the speed and the ease of implementation. While solving this equation, the template matrix taken from the learning process is used to initialize W. Hence, W is deformed to fit the in-ear signal acquired from that user at different nights.

Figure 4. Overview of signal separation process in *LIBS*.



Algorithm 1 Signal Separation Algorithm

```
Input:
    $IS$ - In-ear Signal
    $W_{ini}$ - Spectral Template Matrix
    $ST$ - Segment Time
Output:
    $\hat{X}_{EEG}, \hat{X}_{EOG}, \hat{X}_{EMG}$ - Separated Signals

$\tilde{X} \leftarrow$ PreprocessSignal($IS$);
$X \leftarrow$ ComputePowerSpectrum($\tilde{X}$);
$Seg \leftarrow$ SegmentSignal($IS$, $ST$);
for $i = 1 \rightarrow$ sizeof($Seg$) do
    $H_{ini} \leftarrow$ InitializeMatrixRandomly();
    $[\hat{W}, \hat{H}] \leftarrow$ IS_NMF($Seg_i$);
    $V_{EEG}(Seg_i) \leftarrow \hat{W}_{EEG}(Seg_i) * \hat{H}_{EEG}(Seg_i)$;
    $V_{EOG}(Seg_i) \leftarrow \hat{W}_{EOG}(Seg_i) * \hat{H}_{EOG}(Seg_i)$;
    $V_{EMG}(Seg_i) \leftarrow \hat{W}_{EMG}(Seg_i) * \hat{H}_{EMG}(Seg_i)$;
$\hat{X}_{EEG} \leftarrow$ rescontructSignal($X$, $V_{EEG}$);
$\hat{X}_{EOG} \leftarrow$ rescontructSignal($X$, $V_{EOG}$);
$\hat{X}_{EMG} \leftarrow$ rescontructSignal($X$, $V_{EMG}$);
```

In this work, adapting the technique from Ref. Damon et al.,[2] we specifically select the Itakura-Saito (IS) divergence d_{IS} as a measure to minimize the error between the power spectrum of the original signal and its reconstruction from W and H. The IS divergence, in detail, is a limit case of the β-divergence introduced in Ref. Févotte and Idier,[4] which is defined here

$$d_\beta(x \mid y) = \begin{cases} \dfrac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}) & \beta \in \mathbb{R} \setminus \{0, 1\} \\ x(\log x - \log y) + (y - x) & \beta = 1 \\ \dfrac{x}{y} - \log\dfrac{x}{y} - 1 & \beta = 0 \end{cases} \tag{4}$$

The reason is that a noteworthy property of the β-divergence (in which the IS divergence corresponds to the case $\beta = 0$) is its behavior w.r.t scale. Alternatively, IS divergence holds a scale-invariant property $d_{IS}(\lambda x \mid \lambda y) = d_{IS}(x \mid y)$ that helps minimize the variation of the signals acquired from one person in different recordings. The IS divergence is given by,

$$d_{IS}\left(X \mid W H\right) = \frac{X}{W H} - \log\frac{X}{W H} - 1 \tag{5}$$

Hence, Algorithm 1 provides the whole process of separating EEG, EOG, and EMG signals from the single-channel in-ear mixture using a per-user trained template matrix.

5. SLEEP STAGES CLASSIFICATION

Human sleep naturally proceeds in a repeated cycle of four distinct sleep stages: N1, N2, N3, and REM sleep. To study the sleep quantity and quality, the sleep stages are mainly identified by simultaneously evaluating three fundamental measurement modalities including brain activities, eye movements, and muscle contractions. In hospital, an expert can visually inspect EEG, EOG, and EMG signals collected from subjects during sleep and label each segment (i.e., a 30sec period) with the corresponding sleep stage based on known visual cues associated with each stage. Below we elaborate on each of aforementioned steps of our data analysis pipeline.

5.1. Feature extraction

The features selected for extraction are from a variety of categories as follows:

Temporal features. This category includes typical features used in the literature such as mean, variance, median, skewness, kurtosis, and 75th percentile, which can be derived from the time series. In sleep stage classification, both EOG and EMG signals are often analyzed in the time domain due to their large variation in amplitude and a lack of distinctive frequency patterns. Accordingly, based on our observations about these signals, we include more features that can distinguish N1 from REM, which are often misclassified. In particular, we consider average amplitude that is significantly low for EMG while relatively higher for EOG during the REM stage. Also to capture the variation in EOG during different sleep stages, we consider the variance and entropy for EOG in order to magnify distinctions between Wakefulness, REM, and N1 stages.

Spectral features. These features are often extracted to analyze the characteristics of EEG signal because brain waves are normally available in discrete frequency ranges in different stages. By transforming the time series signal into the frequency domain in different frequency bands and computing its power spectrum density, various spectral features can be studied. Here based on our domain knowledge about the EEG patterns in each sleep stage, we identify and leverage spectral edge frequencies to distinguish those stages.

Non-linear features. Bioelectrical signals show various complex behaviors with nonlinear properties. In details, since the chaotic parameters of EEG are dependent on the sleep stages,[11] they can be used for sleep stage classification. The discriminant ability of such features is demonstrated through the measures of complexity such as correlation dimension, Lyapunov exponent, entropy, fractal dimension, etc.[23]

For this study, relied on the literature of feature-based EOG, EMG, and EEG classification,[11] we consider the features listed in Table 1 from each of the aforementioned categories.

Table 1. List of features extracted from the biosignals.

Features	
Temporal features	average amplitude, variance, 75th percentile, skewness, kurtosis
Spectral features	absolute spectral powers relative spectral powers relative spectral ratio spectral edge frequency
Non-linear features	fractal dimension, entropy

5.2. Feature selection

Although each extracted feature has the ability to partially classify biosignals, the performance of a classification algorithm can degrade when all extracted features are used to determine the sleep stages. Therefore, in order to select a set of relevant features among the extracted ones, we compute the discriminating power of each of them[19] when they are used in combination. However, it is computationally impractical to test all of the possible feature combinations. Therefore, we adopt a procedure called Sequential Forward Selection (SFS)[26] to identify the most effective combination of features extracted from our in-ear signal. With SFS, features are selected sequentially until the addition of a new feature results in no performance improvement in prediction. To further improve the efficiency of our selection method, we have considered additional criteria for selecting features. In particular, we assigned a weight to each feature based on its classification capability and relevance to other features. Subsequently, these weight factors are adjusted based on the classification error. Furthermore, a feature is added to the set of selected features if it not only improves the misclassification error but also is less redundant given the features already selected. With this approach, we can efficiently rank discriminant features based on the intrinsic behavior of the EEG, EMG, and EOG signals.

5.3. Sleep stage classification

Various classification methods are proposed in the literature for similar applications and each has advantages and disadvantages. Some scholars[11] have chosen the Artificial Neural Network (ANN) classification approach for sleep scoring. In spite of the ANN ability to classify untrained patterns, long training time and complexity for selection of parameters such as network topology. Moreover, since decision tree is easier to implement and interpret as compared to other algorithms, it is widely used for sleep stage classification.

Another classification method used for sleep stage identification is SVM. SVM is a machine learning method based on statistical learning theory. Since SVM can be used for large data sets with high accuracy rates, it has also been widely used by various studies[18] to classify sleep stages. However, this approach suffers from long training time and difficulty to understand the learned function. Based on the existing comparative studies,[19] the decision tree (and more generally random forest) classification methods have achieved the highest performance since the tree structure can separate the sleep stages with large variation. As an example, decision tree classifiers are flexible and work well with categorical data. However, overfitting and high dimensionality are the main challenges in decision trees. Therefore, we use an ensemble learning method for classification of in-ear signal. Particularly, we deploy random forest with twenty five decision trees as a suitable classifier for our system. This classifier is able to efficiently handle high dimensional attributes and it also reduces computational cost on large training data sets. The set of features selected through SFS are used to construct a multitude of decision trees at training stage to identify the corresponding sleep stage for every 30sec segment of the biosignals in the classification stage.

6. EVALUATION

In this section, we first present the key results in proving the feasibility of *LIBS* to capture the usable and reliable biosignals, in which all EEG, EOG, and EMG is present. From the success of our proof-of-concept, we then show the performance of our proposed separation algorithm for splitting those three signals without loss of information. Finally, we evaluate the usability of *LIBS*'s outputs through the performance of the automatic sleep stage classification.

6.1. Experiment methodology

Beyond our *LIBS* prototype shown in Figure 5, we used a portable PSG device named Trackit Mark III supported by LifeLines Neurodiagnostic Systems Inc. company[21] with 14 EEG electrodes placed at the channel Fp1, Fp2, C3, C4, O1, and O2 (in accordance to the International 10–20 system) on the scalp, in proximity to the right and left outer cantus, and over the chin, which were all referenced to two mastoids, to collect the ground truth. This device individually acquires EEG, EOG, and EMG signals at 256Hz sampling rate and prefiltered them in the range of 0.1–70Hz.

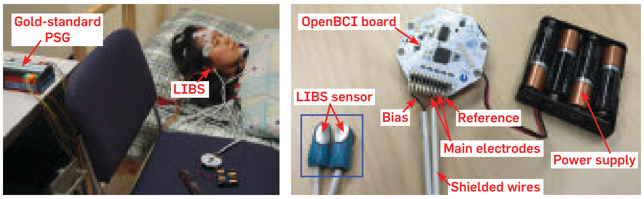
6.2. Validation of signal presence

In this evaluation, we assess the presence of the signals of interest in the in-ear mixed signal measured by *LIBS* by comparing the recording with the groundtruth signals acquired from the gold-standard PSG channels. While the user wears both devices at the same time, we illustrate the feasibility of *LIBS* to produce the usable and reliable signals through different experiments.

We first examined if *LIBS* can capture the EMG signal by asking a subject to do two different activities for contracting his facial muscles. Specifically, the subject kept his teeth remaining still and then grinding for 5sec and chewing for 20sec continuously. This combination was done for four times. From Figure 6a, we noticed that our *LIBS* device could clearly capture those events reflecting the occurrence of the EMG signal.

Similarly, we asked the subject to look forward for 20sec and then move his eyes to points pre-specified in four directions (i.e., left, right, up, and down) for 5sec. As a result, although the amplitude of the in-ear mixed signal is smaller than the gold-standard one, it still clearly exhibits the left and right movements of the eyes similar to the EOG signal channeled in the gold-standard device. As shown in Figure 6b, *LIBS* also has the ability to capture the horizontal and vertical eye movements as the reflection of EOG occurrence.

Figure 5. Demonstration of a sleep study and the first prototype of *LIBS*.

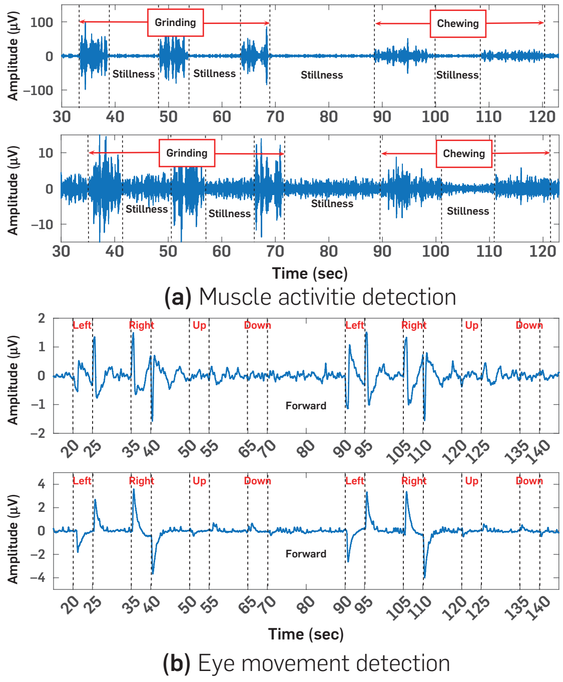


On the other hand, we conducted the following standard Brain-Computer Interaction (BCI) experiments to verify the occurrence of the EEG signal in *LIBS*'s recordings:

Auditory Steady-State Response (ASSR). This EEG paradigm measures the response of human brain while modulating auditory stimuli with specific frequency ranges.[24] In this experiment, we applied auditory stimuli in the frequencies of 40Hz in which each stimuli lasted for 30sec and was repeated three times with 20sec rest between them. Then, by looking at Figure 7, it is easy to recognize a sharp and dominant peak at 40Hz produced during the 40Hz ASSR experiment. Clearly, this result demonstrates the ability of *LIBS* to capture such the specific frequency in the in-ear mixed signal although the peak extracted from the gold standard electrodes was larger than that of *LIBS* electrode.

Steady-State Visually Evoked Potential (SSVEP). Similar to ASSR, SSVEP measures the brain wave responding to a visual

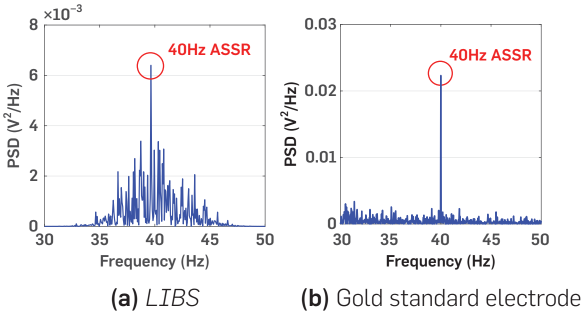
Figure 6. The detection of (a) muscle activities and (b) eye movements from *LIBS* (top) and the gold standard EMG and EOG channels (bottom), respectively.



(**a**) Muscle activitie detection

(**b**) Eye movement detection

Figure 7. The ASSR for 40Hz recorded from (a) *LIBS* and (b) the gold standard device at Channel C3 on scalp.



(**a**) *LIBS*

(**b**) Gold standard electrode

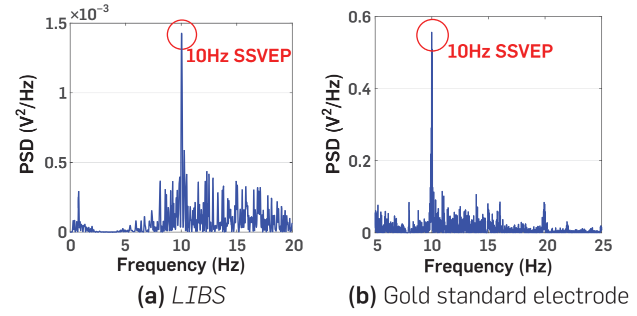
stimuli at specific frequencies.[12] Particularly, we created a blinking stimuli at 10Hz and played it for 20sec with three time repetition. Accordingly, the brain response in this SSVEP experiment comprehensibly presented as a dominant peak for *LIBS* and the gold standard on-scalp electrodes in Figure 8.

Alpha Attenuation Response (AAR). Alpha wave is a type of brain waves specified in the range of 8–13Hz. This brain wave is a sign of relaxation and peacefulness.[1] In this experiment, we asked the subject to completely relax his body while closing his eyes for 20sec and then open them for 10sec in five consecutive times. As analyzing the recorded in-ear mixed signal, Figure 9 shows that *LIBS* is able to capture the alpha rhythm from inside the ear. However, the detection of alpha rhythm in case of *LIBS* was not very clear. This can be due to the fact that the alpha waves were produced in frontal lobe that is in a distance from the ear location.

6.3. Signal separation validation

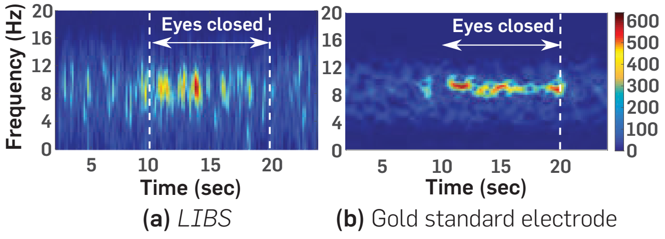
From the previous experiments, we proved that all of the EEG, EOG, and EMG signals appeared in the recordings of *LIBS* and were mixed in the original in-ear signal. We now show the result of our proposed NMF-based separation algorithm, which learns the underlying characteristics of gold standard EEG, EOG, and EMG signals individually and adapts its learned knowledge to provide the best decomposition from the mixed signal. In this evaluation, because the gold standard device (e.g., PSG device) cannot be hooked up in the ear canal to capture the same signal as our in-ear device does, similarity measures such as mutual information, cross-correlation, etc. cannot be used to provide a numeric comparison between the separated and gold standard signals. We then demonstrate the performance of our proposed model by analyzing the occurrence of special frequencies (i.e., the delta brain wave) in the separated EEG biosignal during the sleep study.

Figure 8. The SSVEP responses recorded from (a) *LIBS* and (b) the gold standard device at Channel O1 on scalp.



(a) *LIBS* (b) Gold standard electrode

Figure 9. The detection of alpha rhythms from (a) *LIBS* and (b) the gold standard device at Channel C4 on scalp.



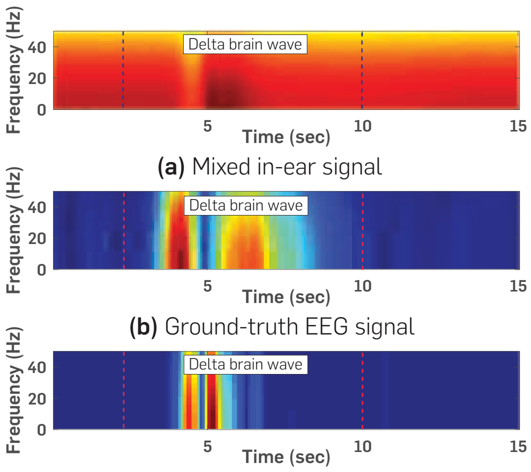
(a) *LIBS* (b) Gold standard electrode

Specifically, Figure 10a provides the spectrogram of a 30sec original in-ear mixed signal captured by *LIBS* during a sleep study and labeled as stage Slow-Wave Sleep (SWS) by the gold-standard device. In Figure 10b, the spectrogram of a corresponding 30sec ground-truth EEG signal is presented. By observing the second spectrogram, a delta brain wave in a frequency range lower than 4Hz is correctly found. However, the spectrogram in Figure 10a cannot show the detection of such the brain wave clearly. Its reason is that not only the delta brain wave exists but also other biosignals are added in this original signal. Finally, Figure 10c exhibits the spectrogram of the EEG signal separated from the original mixed signal by applying our proposed signal separation algorithm. Analyzing this figure proves that the separation model we propose has a capability of not only splitting the signals from the mixed one but keeping only the specific characteristics of the separated signal as well. Otherwise, the short appearance of the delta brain wave in the decomposed signal can be explained by the fact that the location where *LIBS* is placed is far from the source of the signal. By that, the amplitude of the signal is highly reduced.

6.4. Sleep stage classification evaluation

To evaluate the performance of our proposed sleep staging method, we conducted a 38hrs of sleep experiments over eight graduate students (three females and five males) with an average age of 25 to evaluate the performance of the proposed sleep stage classification system inputting the biosignals returned by *LIBS*. An full board Institutional Review Board (IRB) review was conducted and an approval was granted for this study. The participants were asked to sleep in a sleep lab while plugging *LIBS* into their ear canals and have a conventional PSG hook-up around their head simultaneously. After that, the Polysmith program[17] was run to score the ground-truth signals into different sleep stages at

Figure 10. Signal separation performance obtained by *LIBS* through a 30sec mixed in-ear signal (a) and compared with the ground-truth EEG signal (b), and its corresponding separated EEG signal (c).



(a) Mixed in-ear signal

(b) Ground-truth EEG signal

(c) EEG signal separated from in-ear signal every 30sec segment. For all studies, the sleeping environment was set up to be quiet, dark, and cool.

Statistically, we extracted the features from 4313 30sec segments using the original mixed signal as well as three separated signals. Training and test data sets are randomly selected from the same subject pool. Figure 11 displays the results of the sleep stage classification in comparison to the hypnogram of the test data scores out of the gold standard PSG. From this, we observe that the dynamics of the hypnogram is almost completely maintained in the predicted scores. Moreover, our result show that the end-to-end sleep staging system can achieve 95% accuracy on average.

We refer the readers to Ref. Nguyen et al.[13] for more detailed validations of signal acquisition and separation, their comparison with the signals recorded by the gold-standard device, and our user study.

7. POTENTIALS OF LIBS

We envision *LIBS* to be an enabling platform for not only healthcare applications but also those from other domains. Figure 12 illustrates the eight potential applications including in-home sleep monitoring, autism onset detection, meditation training, eating habit monitoring, autonomous audio steering, distraction and drowsiness detection, child's interest assessment, and human-computer interaction. We discuss these exemplary applications below.

7.1. Healthcare applications

We propose three applications that *LIBS* can be extended to serve in healthcare: autism act-out onset detection, meditation coaching, and eating habit monitoring.

Autism onset detection. Thanks to its ability to capture muscle tension, eye movements, and brain activities, *LIBS* has a potential to be an autism on-set detection and prediction wearable. Particularly, people with autism can have very sensitive sensory (e.g., visual, auditory, and tactile) functions. When any of their sensory functions leads to an overload, their brain signal, facial muscle, and eye movement are expected to change significantly.[5] We hope to explore this phenomenon to detect the relationship between these three signals and the on-set event from which a prediction model can be developed.

Meditation training. Meditation has a potential for improving physical and mental well-being when it is done in a right way. Hence, it is necessary to understand people's mindfulness level during the meditation to be able to provide more efficient instructions. Existing commercial off-the-shelf devices (e.g., MUSE) only capture the brain signal to tell how well users are meditating. Different from them, *LIBS* further looks at the eye and muscle signals to analyze the level of relaxation they have more accurately. As a result, *LIBS* promisingly helps improve the users' meditation performance.

Eating habit monitoring. Eating habits can provide critical evidences for various diseases.[8] As *LIBS* can capture the muscle signal very clearly, such information can be useful to infer how often the users chew, how fast they chew, how much they chew, and what the intensity of their chewing is. From all of that, *LIBS* can then predict what foods they are eating as well as how much they are eating. As a result, *LIBS* can provide users guidance to avoid their bad habits by themselves or to visit a doctor if necessary.

7.2. Non-health applications

LIBS can benefit applications and systems on other domain such as improving hearing aid devices, improving driver's safety, helping parents with orienting their child early on.

Autonomous audio steering. This application helps solve a classical problem in hearing aid, which is called cocktail party problem. As known, state-of-the-art hearing aid devices try to amplify the sounds coming from the area that has large amplitude, which is assumed as human voice, in a party. Consequently, the hearing aid will fail to support the wearers if any group of people behind them is talking very loudly, which is not the right person they want to talk to. Using our technology, using the eye signal *LIBS* can capture, it will possibly detect the area that the users are paying their attention to. Furthermore, combining with their brain signal, *LIBS* can further predict how please the wearers are with the output sound that their hearing aid is producing. With that in mind, *LIBS* can steer the hearing aid and improve its quality of amplification so that the hearing aid can provide the high-quality sounds coming from the right source to the users.

Distraction and drowsiness detection. Distraction and drowsiness are very serious factors in driving. Specifically, if people feel drowsy, their brain signal will be in alpha state, their eyes will be closed, and their chin muscle tone will become relax.[20] Also, it is easy to detect if people are distracted based on the localization of eye positions when we analyze the changes of the eye signal. Hence, *LIBS* with three separated brain, eye, and muscle signals should be able to determine the driver's drowsiness level or distraction to further send an alert for avoiding road accidents.

Figure 11. A hypnogram of 30min data resulted by our classification algorithm.

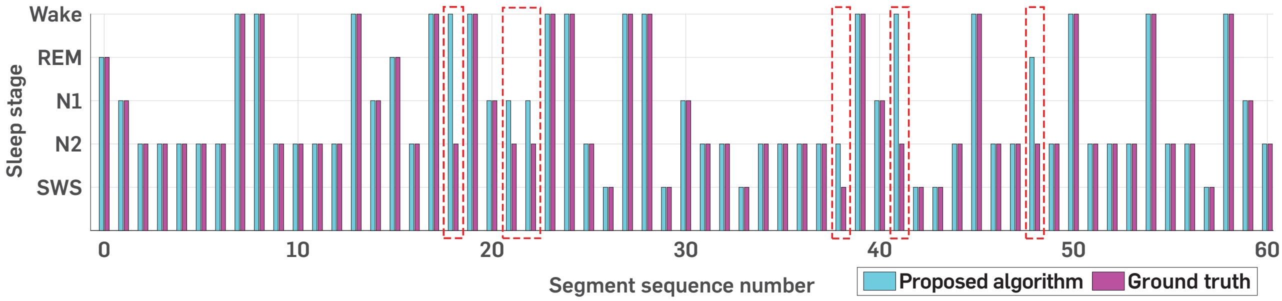
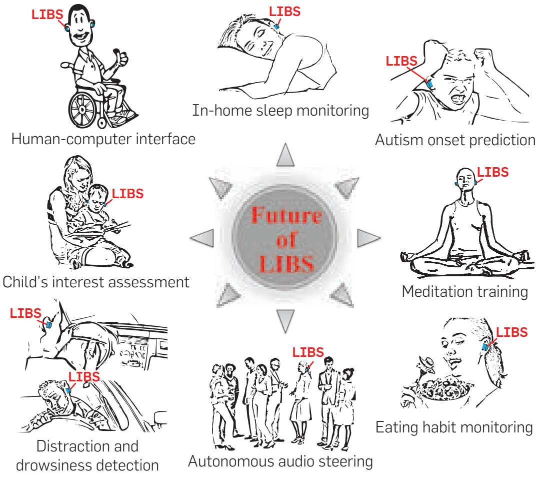


Figure 12. Potential applications of *LIBS*.



Child's interest assessment. With *LIBS*, child's interest assessment can be done less obtrusively and yield more accurate outcomes. Moreover, from that, the parents will be able to orient them accordingly so that they can learn what they like the most. Clinically, kids from the age of 0–2yrs don't have the ability to express their interest. More precisely, the only way to express their interest is their crying. As a result, the conventional gold-standard device (i.e., PSG) is usually used to read their biosignals, which relatively reflect their interest in what they are allowed to do. However, it is not comfortable for them to wear and do activities during the assessment. Hence, by leveraging *LIBS* to read the signal from their ears and at the same time letting them play different sports or learn different subjects, *LIBS* should be able to infer what the level of their interest is with high comfort.

Human-computer interaction. In a broader context, *LIBS* can be used as a form of Human Computer Interaction (HCI), which can especially benefit users with disability. In stead of using only the brain signal as found in many HCI and brain-to-computer systems today, *LIBS* can combine the information extracted from the three separated signals to enrich commands the user can build to interact with the computer in a more reliable way. This gives users more choices for integration with computing systems in a potentially more precise and convenient manner.

8. CONCLUSION

In this paper, we enabled *LIBS*, a sensing system worn inside human ear canals, that can unobtrusively, comfortably, and continuously monitor the electrical activities of human brain, eyes, and facial muscles. Different from existing hi-tech systems of measuring only one specific type of the signals, *LIBS* deploys a NMF-based signal separation algorithm to feasibly and reliably achieve three individual signals of interest. Through one-month long user study of collecting the in-ear signals during sleep and scoring them into appropriate sleep stages using a prototype, *LIBS* itself demonstrated a promising comparison to the existing dedicated sleep assessment systems in term of accuracy and usability. Further than an in-ear bio-sensing wearable, we view *LIBS* as a key enabling technology for concealed head-worn devices for healthcare and communication applications, especially for personalized health monitoring, digital assistance, and the introduction of socially-aware human-computer interfaces.

Acknowledgments

We thank LifeLines Neurodiagnostic Systems Inc. for their support in providing the gold-standard PSG device and thank Yiming Deng and Titsa Papantoni for their valuable feedback at the early stages of this work. This material is based in part upon work supported by the National Science Foundation under Grant SCH-1602428.

References

1. Alloway, C., et al. The alpha attenuation test: assessing excessive daytime sleepiness in narcolepsy-cataplexy. *Sleep 20*, 4 (1997), 258–266.
2. Damon, C., et al. Non-negative matrix factorization for single-channel EEG artifact rejection. In *ICASSP* (2013), 1177–1181.
3. Essid, S. A single-class SVM based algorithm for computing an identifiable NMF. In *Proceedings of 2012 IEEE ICASSP* (2012), 2053–2056.
4. Févotte, C., Idier, J. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural. Comput. 23*, 9 (2011), 2421–2456.
5. Gillingham, G. *Autism, Handle with Care!: Understanding and Managing Behavior of Children and Adults with Autism*. Future Education, Texas (1995).
6. Goverdovsky, V., et al. In-ear EEG from viscoelastic generic earpieces: Robust and unobtrusive 24/7 monitoring. *IEEE Sens. J.* (2015).
7. Hallez, H., et al. Review on solving the forward problem in EEG source analysis. *J. Neuroeng. Rehabil. 4*, 1 (2007), 1–29.
8. Kalantarian, H., et al. Monitoring eating habits using a piezoelectric sensor-based necklace. *Comput. Biol. Med. 58*, Supplement C (2015), 46–55.
9. Kushida, C., et al. Practice parameters for the indications for polysomnography and related procedures: an update for 2005. *Sleep 28*, 4 (2005), 499–521.
10. Lee, J.H., et al. CNT/PDMS-based canal-typed ear electrodes for inconspicuous EEG recording. *J. Neural Eng. 11*, 4 (2014).
11. Motamedi-Fakhr, S., et al. Signal processing techniques applied to human sleep EEG signals—A review. *Biomed. Signal Process. Control 10* (2014), 21–33.
12. Nawrocka, A., Holewa, K. Brain—Computer interface based on Steady—State Visual Evoked Potentials (SSVEP). In *Proceedings of 14th ICCC* (2013), 251–254.
13. Nguyen, A., Alqurashi, R., Raghebi, Z., Banaei-kashani, F., Halbower, A.C., Vu, T. A lightweight and inexpensive in-ear sensing system for automatic whole-night sleep stage monitoring. SenSys '16 (2016), 230–244.
14. Oliveira, R. The dynamic ear canal. *Ballachandra B, ed. The Human Ear Canal. San Diego: Singular Publishing Group* (1995), 83–111.
15. Oliveira, R., et al. A look at ear canal changes with jaw motion. *Ear Hear 13*, 6 (1992), 464–466.
16. OpenBCI. http://openbci.com/.
17. Polysmith–NIHON KOHDEN. http://www.nihonkohden.de/.
18. Ronzhina, M., et al. Sleep scoring using artificial neural networks. *Sleep Med. Rev. 16*, 3 (2012), 251–263.
19. Sen, B., et al. A Comparative Study on Classification of Sleep Stage Based on EEG Signals Using Feature Selection and Classification Algorithms. *J. Med. Syst. 38*, 3 (2014), 1–21.
20. Silber, M.H., et al. The visual scoring of sleep in adults. *J. Clin. Sleep Med. 3*, 02 (2007), 121–131.
21. Trackit Mark III—LifeLines Neurodiagnostic Systems. https://www.lifelinesneuro.com/.
22. Tse, D., Viswanath, P. *Fundamentals of Wireless Communication*. Cambridge University Press, New York (2005).
23. R.A.U., et al. Non-linear analysis of EEG signals at various sleep stages. *Comput. Methods and Programs Biomed. 80*, 1 (2005), 37–45.
24. van der Reijden, C., et al. Signal-to-noise ratios of the auditory steady-state response from fifty-five EEG derivations in adults. *J. Am. Acad. Audiol. 15*, 10 (2004), 692–701.
25. Virtanen, T., et al. Compositional Models for Audio Processing: Uncovering the structure of sound mixtures. *IEEE Signal Processing Mag. 32*, 2 (2015), 125–144.
26. Zoubek, L., et al. Feature selection for sleep/wake stages classification using data driven methods. *Biomed. Signal Process. Control 2*, 3 (2007), 171–179.

Anh Nguyen and Tam Vu, University of Colorado, Boulder, CO, USA.

Raghda Alqurashi, Zohreh Raghebi, and Farnoush Banaei-Kashani, University of Colorado, Denver, CO, USA.

Ann C. Halbower, University of Colorado, School of Medicine, Aurora, CO, USA.

CAREERS

Augusta University

Tenure Track and Tenured Positions at the Assistant, Associate, and Full Professor Levels

The School of Computer and Cyber Sciences at Augusta University was founded in 2017 with the mission to provide high-engagement, state-of-the-art education, and research across its Computer Science, Information Technology, and Cybersecurity disciplines, and with the vision of becoming a national leader in Cybersecurity. The School is embarking on a path of unprecedented growth to become a comprehensive research and education college, with substantial increases in faculty, and graduate and undergraduate enrollment.

Augusta, Georgia, is becoming a primary hub for cybersecurity in the United States, and the area is poised for explosive development. It is located at the center of a number of academic, governmental and corporate partnerships critical to the nation's cyber security, including the U.S. Army Cyber Center of Excellence, the National Security Agency Georgia, the future home of the United States Army Cyber Command, and the nearby Savannah River National Laboratory in South Carolina. The State of Georgia invested $100M in Georgia Cyber Center at Augusta University, a 167,000-square-foot research and education facility which opened on July 10, 2018 and is home to the School of Computer and Cyber Sciences. The second, 165,000-square building of the Center is under construction to be completed in December of 2018.

Augusta University has embarked on an ambitious, multi-year effort to significantly expand its computing, cybersecurity, and data science activities. Applications are being invited for 12 tenure-track and tenured positions at the Assistant, Associate, and Full Professor levels, with responsibilities to advance education and research in all mainstream areas of computer science and possibly drawing from closely related or emerging fields.

Information about the school and a description of open positions are available on the school website at http://www.augusta.edu/ccs.

Requirements

Applicants must hold a PhD in Computer Science or a related discipline at the time of appointment, have demonstrated excellence in research, and a strong commitment to teaching. Outstanding candidates in all areas of computer science will be considered with a target appointment date of Fall 2019. Review of applications and candidate interviews will begin December 1 and continue until the positions are filled.

To be considered as an applicant, the following materials are required:

- Cover letter
- Curriculum vitae including a list of publications
- Statement describing research accomplishments and future research plans
- Description of teaching philosophy and experience
- Names of at least three references

The above items should be either emailed to ccs@augusta.edu **or mailed to Chair Search Committee, School of Computer and Cyber Sciences, Augusta University, 1120 15th Street, UH-127, Augusta, GA 30912.**

Boston College

Assistant Professor of the Practice or Lecturer in Computer Science

The Computer Science Department of Boston College seeks to fill one or more non-tenure-track teaching positions, as well as shorter-term visiting teaching positions. All applicants should be committed to excellence in undergraduate education, and be able to teach a broad variety of undergraduate computer science courses. Faculty in longer-term positions will participate in the development of new courses that reflect the evolving landscape of the discipline.

Minimum requirements for the title of Assistant Professor of the Practice, and for the title of Visiting Assistant Professor, include a Ph.D. in Computer Science or closely related discipline. Candidates who have only attained a Master's degree would be eligible for the title of Lecturer, or Visiting Lecturer. See https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html for more information.

To apply go to
http://apply.interfolio.com/54268.
Application process begins October 1, 2018.

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at http://www.bc.edu/offices/diversity.

Boston College

Associate or Full Professor of Computer Science

Description:

The Computer Science Department of Boston College is poised for significant growth over the next several years and seeks to fill faculty positions at all levels beginning in the 2019-2020 academic year. Outstanding candidates in all areas will be considered, with a preference for those who demonstrate a potential to contribute to cross-disciplinary teaching and research in conjunction with the planned Schiller Institute for Integrated Science and Society at Boston College. See https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html and https://www.bc.edu/bc-web/schools/mcas/sites/schiller-institute.html for more information.

Qualifications:

A Ph.D. in Computer Science or a closely related discipline is required, together with a distinguished track record of research and external funding, and evidence of the potential to play a leading role in the future direction of the department, both in the recruitment of faculty and the development of new academic programs.

To apply go to
http://apply.interfolio.com/54226.
Application process begins October 1, 2018.

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at http://www.bc.edu/offices/diversity.

Boston College

Tenure Track, Assistant Professor of Computer Science

The Computer Science Department of Boston College is poised for significant growth over the next several years and seeks to fill faculty positions at all levels beginning in the 2019-2020 academic year. Outstanding candidates in all areas will be considered, with a preference for those who demonstrate a potential to contribute to cross-disciplinary teaching and research in conjunction with the planned Schiller Institute for Integrated Science and Society at Boston College. A Ph.D. in Computer Science or a closely related discipline is required for all positions. See https://www.bc.edu/bc-web/schools/mcas/departments/computer-science.html and https://www.bc.edu/bc-web/schools/mcas/sites/schiller-institute.html for more information.

Successful candidates for the position of Assistant Professor will be expected to develop strong research programs that can attract external research funding in an environment that also values high-quality undergraduate teaching.

Minimum requirements for all positions include a Ph.D. in Computer Science or closely related discipline, an energetic research program that promises to attract external funding, and a commitment to quality in undergraduate and graduate education.

To apply go to

https://apply.interfolio.com/54208.

Application review begins October 1, 2018.

Boston College is a Jesuit, Catholic university that strives to integrate research excellence with a foundational commitment to formative liberal arts education. We encourage applications from candidates who are committed to fostering a diverse and inclusive academic community. Boston College is an Affirmative Action/Equal Opportunity Employer and does not discriminate on the basis of any legally protected category including disability and protected veteran status. To learn more about how BC supports diversity and inclusion throughout the university, please visit the Office for Institutional Diversity at http://www.bc.edu/offices/diversity.

California Institute of Technology

Tenure-Track Faculty Position

The Computing and Mathematical Sciences (CMS) Department at the California Institute of Technology (Caltech) invites applications for a tenure-track faculty position in the fundamental mathematics and theory that underpins application domains within the CMS Department, within the Engineering and Applied Sciences (EAS) Division, or within the Institute as a whole. Areas of interest include (but are not limited to) algorithms, data assimilation and inverse problems, dynamical systems and control, geometry, machine learning, mathematics of data science, networks and graphs, numerical linear algebra, optimization, partial differential equations, probability, scientific computing, statistics, stochastic modeling, and uncertainty quantification.

CMS is a unique environment where research in applied and computational mathematics, computer science, and control and dynamical systems is conducted in a collegial atmosphere; application focii include distributed systems, economics, graphics, neuroscience, quantum computing, and robotics and autonomous systems. The CMS Department is part of the broader EAS Division comprising researchers working in, and at intersections between, the fields of aerospace, civil, electrical, mechanical, and medical engineering, as well as in environmental science and engineering, and in materials science and applied physics. The Institute as a whole represents the full range of research in biology, chemistry, engineering, physics, and the social sciences.

A commitment to world-class research, as well as high-quality teaching and mentoring, is expected. The initial appointment at the assistant professor level is for four years, and is contingent upon the completion of a Ph.D. degree in applied mathematics, computer science, statistics or in a related field in engineering or the sciences.

Applications will be reviewed beginning November 7, 2018, and applicants are encouraged to have all their application materials, including letters of recommendation, on file by this date. For a list of documents required, and full instructions on how to apply online, please visit https://applications.caltech.edu/jobs/cms.

Questions about the application process may be directed to search@cms.caltech.edu.

We are an equal opportunity employer and all qualified applicants will receive consideration for employment without regard to race, color, religion, sex, sexual orientation, gender identity, national origin, disability status, protected veteran status, or any other characteristic protected by law.

California State University, Sacramento

Tenure-Track Assistant Professor

California State University, Sacramento, Department of Computer Science. One tenure-track assistant professor position to begin with the Fall 2019 semester. Applicants specializing in any area of computer science will be considered. Those with expertise in areas related to software engineering, computer architecture, artificial intelligence, or deep learning are especially encouraged to apply. Ph.D. in Computer Science, Computer Engineering, or closely related field required by the time of the appointment. For detailed position information, including application procedure, please see http://www.csus.edu/about/employment/. Screening will begin November 19, 2018, and remain open until filled. AA/EEO employer. Clery Act statistics available. Mandated reporter requirements. Criminal background check will be required.

EPFL

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Faculty Positions in Computer and Communication Science

at the Ecole polytechnique fédérale de Lausanne (EPFL)

The School of Computer and Communication Sciences (IC) at EPFL invites applications for faculty positions in computer and communication sciences. We are seeking candidates for tenure-track assistant professor as well as for senior positions.

Successful candidates will develop an independent and creative research program, participate in both undergraduate and graduate teaching, and supervise PhD students.

The school is seeking candidates in the fields of: 1) Machine Learning and Data Science – including applications in bioinformatics, natural language processing, and speech recognition; 2) Verification and Formal Methods; 3) Systems; 4) CS Education and Learning Analytics. Candidates in other areas are also encouraged to apply and will be considered.

EPFL offers internationally competitive salaries, generous research support, significant start-up resources, and outstanding research infrastructure.

Academics in Switzerland enjoy many research funding opportunities, as well as an exceptionally high living standard.

To apply, please follow the application procedure at

https://facultyrecruiting.epfl.ch/position/10977288

The following documents are requested in PDF format: cover letter, curriculum vitae including publication list, brief statements of research and teaching interests, and contact information (name, postal address, and email) of 3 references for junior positions or 5 for senior positions. Screening will start on **December 1, 2018**. Further questions can be addressed to:

Profs. Rüdiger Urbanke and George Candea
Co-Chairs of the Recruiting Committee
CH-1015 Lausanne
recruiting.ic@epfl.ch

For additional information on EPFL and IC:
http://www.epfl.ch, http://ic.epfl.ch

EPFL is an equal opportunity employer and family friendly university. It is committed to increasing the diversity of its faculty. It strongly encourages women to apply.

California State University, San Bernardino

Assistant Professor (Tenure-Track)

The School of Computer Science and Engineering at California State University, San Bernardino invites applications for a tenure-track position at the Assistant Professor level, beginning September 2019. All areas of Computer Science will be considered.

The School of CSE offers the programs of B.S. in Computer Science (ABET accredited), B.S. in Computer Engineering (ABET accredited), B.S. in Bioinformatics, B.A. in Computer Systems, and M.S. in Computer Science.

Candidates must have a Ph.D. in Computer Science or a closely related field by the time of appointment. The position is primarily to support the B.S. in Computer Science (ABET accredited), B.A. in Computer Systems and M.S. in Computer Science programs. The candidate must display potential for excellence in teaching and scholarly work. The candidate is expected to supervise student research at both the undergraduate and graduate levels, and to actively participate in other types of academic student advising. The candidate will actively contribute to the School's curriculum development. The candidate will serve the School, College and University, as well as the community and the profession.

The College of Natural Sciences at CSUSB is committed to creating a welcoming and inclusive climate for people from diverse backgrounds and is committed to enhancing diversity. CNS also strives for excellence and fosters harmony.

Women and underrepresented minorities are strongly encouraged to apply. For more information about the School of Computer Science and Engineering, please visit http://cns.csusb.edu/cse.

ADLINE AND APPLICATION PROCESS:
Please submit 1) curriculum vitae; 2) statement of teaching philosophy; 3) description of research interests; 4) letters from 3 individuals qualified to comment (have letters of recommendation sent via email to facultyrecruitment@csusb.edu); 5) copies of transcripts of all post-secondary degrees (official transcripts will be required prior to appointment).

Also include a Diversity Statement detailing how your teaching and/or service and/or scholarship would support the success of students from racial, ethnic, and gender backgrounds that are underrepresented in your academic field. (Maximum 250 words).

Submit application at https://www.governmentjobs.com/careers/csusb/jobs/2204738.

Formal review of applications will begin Nov. 15, 2018 and continue until the position is filled.

Questions about this position can be directed to Dr. Haiyan Qiao, Director of School of Computer Science and Engineering, at hqiao@csusb.edu.

Calvin College

Tenure-Track Faculty Position

The Department of Computer Science at Calvin College invites applications for a tenure-track faculty position to begin August 2019, pending administrative approval, and strongly encourages applications from women and other underrepresented groups. Our department features supportive colleagues, excellent facilities, hardworking students, a dynamic colloquium series, and strong undergraduate programs in computer science, data science, digital communication, and information systems, including a BCS program accredited by ABET (abet.org).

Applicants should have a PhD (or be near completion) in computer science or a related area, or have a master's degree and 5 years of related experience. We are especially interested in expanding our expertise in the areas of data science & visualization, computer security, or 3D modeling & animation, but individuals from all computing-related areas are encouraged to apply.

Calvin is a Christian comprehensive liberal arts college located in Grand Rapids, Michigan; it is one of the largest Christian colleges in North America, and was named the #1 regional college in the Midwest for 2017 by U.S. News & World Report. Its faculty members are committed to establishing relationships and positive communication across multiple dimensions of diversity, including but not limited to ethnicity, gender, physical limitations, class, or religious perspectives.

For more information and application instructions, see: https://cs.calvin.edu/documents/Tenure_Track_Faculty_Position.

Columbia University

Open Rank Faculty Position in the Department of Electrical Engineering

Columbia Engineering is pleased to invite applications for a faculty position in the Department of Electrical Engineering at Columbia University in the City of New York. Applications at all ranks will be considered.

The Electrical Engineering department welcomes applications in all areas of electrical engineering, and especially encourages candidates with an interest in the school-wide initiatives that relate to engineering and medicine, autonomous systems, quantum computing and technology, and sustainability. Areas of emphasis for Electrical Engineering include (i) signals, information and data, and (ii) energy, including power systems, renewable energy and the optimization and control of the electrical grid. Candidates must have a Ph.D. or its professional equivalent by the starting date of the appointment. Applicants for this position must demonstrate the potential to do pioneering research and to teach effectively. The Department is especially interested in qualified candidates who can contribute, through their research, teaching, and/or service, to the diversity and excellence of the academic community.

The successful candidate is expected to contribute to the advancement of their field and the department by developing an original and leading externally funded research program, and to contribute to the undergraduate and graduate educational mission of the Department. Columbia fosters multidisciplinary research and encourages collaborations with academic departments and units across Columbia University.

For additional information and to apply, please see: http://engineering.columbia.edu/faculty-job-opportunities. Applications should be submitted electronically and include the following: curriculum vitae including a publication list, a description of research accomplishments, a statement of research and teaching interests and plans, contact information for three experts who can provide letters of recommendation, and up to three pre/reprints of scholarly work. All applications received by December 1st, 2018 will receive full consideration.

Applicants can consult http://www.ee.columbia.edu for more information about the department and http://academicjobs.columbia.edu/applicants/Central?quickFind=67066 for more details on the position and application.

Columbia University is an Equal Opportunity Employer / Disability / Veteran.

Georgia Institute of Technology, School of Computational Science and Engineering

Tenure-Track Faculty

The School of Computational Science and Engineering (CSE) of the College of Computing at the Georgia Institute of Technology seeks tenure-track faculty, at all levels, who may specialize in high-performance computing (HPC), data analytics, machine learning (ML), and modeling and simulation, to solve real-world problems in science, engineering, health, and social domains. Our school seeks candidates who may specialize in a broad range of application areas including biomedical and health informatics; urban systems and smart cities; social good and sustainable development; materials and manufacturing; and national security. Applicants must have an outstanding record of research, a sincere commitment to teaching, and interest in engaging in substantive interdisciplinary research with collaborators in other disciplines.

Georgia Tech is located in the heart of metro Atlanta, a home to more than 5.5 million people and nearly 150,000 businesses, a world-class airport, lush parks and green spaces, competitive schools and numerous amenities for entertainment, sports and restaurants that all offer a top-tier quality of life. From its diverse economy, global access, abundant talent and low costs of business and lifestyle, metro Atlanta is a great place to call "home." Residents have easy access to arts, culture, sports and nightlife, and can experience all four seasons – with mild winters that rarely require a snow shovel.

In mid-2019, CSE will move to its new home, the CODA Building. CSE will be the core academic unit in the building, co-located with institutes and centers focused on data engineering and science, ML, health informatics, cybersecurity, and HPC. The 750,000-square-foot mixed-use development represents a $375 million investment into the innovation district of Tech Square in Atlanta and will include an 80,000-square-foot HPC center alongside 620,000 square feet of office space. CSE is unique in that it will be the only school in its entirety to move all staff, operations, research, faculty, and students to this location. This unique placement positions CSE to become a direct partner with the greater CODA community.

Applications should be submitted online through https://academicjobsonline.org/ajo/jobs/11600. For best consideration, applications are due by December 15, 2018. The application material should include a full academic CV, a personal narrative on teaching and research, a list of at least three references and up to three sample publications. Georgia Tech is an Affirmative Action/Equal Opportunity Employer. Applications from women and under-represented minorities are strongly encouraged.

For more information about Georgia Tech's School of Computational Science and Engineering please visit: http://www.cse.gatech.edu/

Georgia Institute of Technology, School of Computer Science

Tenured/Tenure-Track Faculty Position

The School of Computer Science at the Georgia Institute of Technology is recruiting multiple tenure-track faculty. Our preference is for junior-level candidates at the Assistant Professor level, but exceptional candidates at all levels will be considered. We seek candidates who complement and enhance our research strengths in any area, and are especially interested in candidates whose research focuses on theory of computing, data science or security.

The School of Computer Science, one of three schools in the College of Computing, focuses on research that makes computing and communication smart, fast, reliable, and secure, with research groups in computer architecture, databases, machine learning, networking, programming languages, security, software engineering, systems, and theory. Faculty in the school are leaders in a variety of Georgia Tech initiatives, including: Institute for Data Engineering and Science (IDEaS), Institute for Information and Security (IISP), Center for Research into Novel Computing Hierarchies (CRNCH), and Algorithms and Randomness Center (ARC). The school is in a period of rapid growth with five tenure-track assistant professors hired last year.

Georgia Tech is adjacent to the Midtown district of Atlanta. Midtown is a walking, in-town neighborhood, burgeoning with many new cafes and restaurants, home to tech companies, and within walking distance of outdoor activities, including the Beltline, Piedmont Park, Botanical Gardens, and High Museum of Art. Georgia Tech's new CODA building will be located in Midtown and the School of Computer Science will have a strong presence in the new building. The greater Atlanta area is very cosmopolitan, with a variety of international communities and outdoor pursuits (beaches, mountains, etc.) within driving distance.

Applications will be considered until open positions are filled. However the review of applications will begin on December 1, 2018. Applicants are encouraged to clearly identify in their cover letter the area(s) that best describe their research interests. All applications must be submitted online at: https://academicjobsonline.org/ajo/jobs/11942.

Georgia Tech is an equal education/employment opportunity institution dedicated to building a diverse community. We strongly encourage applications from women, underrepresented minorities, individuals with disabilities, and veterans. Georgia Tech has policies to promote a healthy work-life balance and is aware that attracting faculty may require meeting the needs of two careers. More information about the School of Computer Science is available at: http://scs.gatech.edu/.

Purdue University

Head of the Department of Computer Science

The College of Science at Purdue University invites nominations and applications for the position of Head of the Department of Computer Science. The department seeks a dynamic research leader and innovative educator with creative vision and an outstanding record of achievement.

The department's teaching and research activities cover a broad range of topics including bioinformatics and computational biology, computational science and engineering, databases and data mining, distributed systems, graphics and visualization, information security and assurance, machine learning and information retrieval, networking and operating systems, programming languages and compilers, software engineering, and theory of computing and algorithms. For more information and the online version of the ad, see http://www.cs.purdue.edu/.

The successful candidate will have an exemplary record of scholarly achievement along with outstanding leadership potential. It is expected that candidates for this position will have an earned doctorate in computer science or a related field and a level of stature in the field sufficient at a minimum to merit appointment with tenure at the rank of Professor. The Head will work with faculty colleagues to build and achieve a compelling vision for the future of the Department, and to continue to accelerate the advancement of its nationally-ranked program through a commitment to excel in all aspects of the Department's mission. Highly desirable qualities include an understand-

Worcester Polytechnic Institute

Assistant Professor of Data Science

The Data Science program invites applications for a tenure track position with a research focus in Data Science to begin in Fall 2019 to strengthen this strategic interdisciplinary area. The new faculty will join the strong team of existing data science faculty working on interdisciplinary research related to big data on real-world grand challenge problems with societal impact.

The highly interdisciplinary Data Science program at WPI, a collaboration between Computer Science, Mathematical Sciences and the Robert A. Foisie School of Business, has undergone major growth since its inception in 2014 supported by a cluster hire of seven full-time faculty in Data Science and closely related disciplines. The signature Data Science program offers on-campus degree programs at all levels, including undergraduate minor in Data Science, MS degree in Data Science, and the first interdisciplinary PhD degree in Data Science in the nation.

WPI is interested in applicants with research and teaching expertise in all areas of Data Science, but in particular in applicants with strong background complementary to the existing expertise.

Areas of strength in Data Science at WPI include statistical learning, machine learning, deep learning, large-scale data management, compressed sensing, big data analytics, signal processing, visualization, artificial intelligence, and Data Science applications from digital health, cyber security, social media, material sciences, neuro-sciences, connected communities, urban computing, to bioinformatics. Outstanding candidates in any area of Data Science will receive full consideration.

Founded in 1865, WPI is one of the nation's first technological Universities known for its innovative project-based education. WPI's reputation as an innovative university rests on the shoulders of its faculty. A highly selective, private technological university and one of the nation's first, WPI believes that when great minds work together, great advances follow. At WPI the boundaries to multidisciplinary collaboration are low faculty members - students, and other partners work together on real-world projects and purposeful research. We are most proud of a recent No. 1 ranking for "faculty who best combine research and teaching." (Wall Street Journal/Times Higher Ed, 2016).

The successful candidate will hold an academic appointment in either the Mathematical Sciences or the Computer Science Department.

Candidates should have a Ph.D. in Computer Science, Mathematical Sciences, Statistics, Electrical Engineering or a closely related field, and the potential for excellence in teaching and research.

Candidates should include detailed research and teaching statements, vitae, and contact information for at least three references.

To apply, visit **http://apptrkr.com/1286303**. The deadline for applications is December 10, 2018. Applications will be considered after that date until the position is filled.

WPI is an Equal Opportunity Employer. All qualified candidates will receive consideration for employment without regard to race, color, age, religion, sex, sexual orientation, gender identity, national origin, veteran status, or disability. We are seeking individuals with diverse backgrounds and experiences who will contribute to a culture of creativity and collaboration, inclusion, problem solving and change making.

ing of the current needs and future direction of computer science as an academic discipline, a commitment to diversity and collaboration, and skills in academic leadership, student relations, mentoring, and alumni relations development.

Confidential nominations and inquires can be sent to head-search@cs.purdue.edu. Candidates should submit a letter of application articulating a vision for the future of academic computer science, a statement of research and teaching, and a complete curriculum vitae with names and email addresses of at least five references. Applications and nominations will be held in strict confidence; review of the same will begin immediately and continue until the position is filled. Application materials can be uploaded at: https://hiring.science.purdue.edu.

A background check will be required for employment in this position. Purdue University's Department of Computer Science is committed to advancing diversity in all areas of faculty effort, including scholarship, instruction, and engagement. Candidates should address at least one of these areas in their cover letter, indicating their past experiences, current interests or activities, and/or future goals to promote a climate that values diversity and inclusion.

Purdue University is an EOE/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.

San Diego State University

Department of Computer Science

Two Tenure-Track Assistant Professor Positions

The Department of Computer Science at SDSU seeks to hire two tenure-track Assistant Professors starting Fall 2019. The candidates should have PhD degrees in Computer Science or closely related fields. One position is in Cybersecurity (see https://apply.interfolio.com/53552); the other position is in Algorithms & Computation (see https://apply.interfolio.com/53547). Questions about the position may be directed to COS-CS-Search@sdsu.edu. Top candidates in other areas will also be considered. SDSU is an equal opportunity/Title IX employer.

Texas State University

Department of Computer Science

The Department of Computer Science invites applications for three faculty positions:

1. One tenure-track Assistant Professor to start on September 1, 2019. Review date is February 4, 2019.
2. Two non-tenure track Senior Lecturers to start on September 1, 2019. Review date is March 4, 2019.

Please consult the department's webpage at www.cs.txstate.edu/employment/faculty/ for job duties, required and preferred qualifications, application procedures, and information about the university and the department.

Texas State University is committed to an inclusive education and work environment that provides equal opportunity and access to all qualified persons. Texas State, to the extent not in conflict with federal or state law, prohibits discrimination or harassment on the basis of race, color, national origin, age, sex, religion, disability, veterans' status, sexual orientation, gender identity or expression. Texas State University is a member of The Texas State University System. Texas State University is an EOE.

Trinity College, Hartford, Connecticut

Assistant Professor of Computer Science

Applications are invited for a tenure-track position in computer science at the rank of Assistant Professor to start in the fall of 2019.

Candidates must hold a Ph.D. in computer science at the time of appointment. We are seeking candidates with teaching and research interests in applied areas associated with data analytics, such as database and information systems, data mining and knowledge discovery, machine learning, and artificial intelligence, but other related areas will also be seriously considered.

Trinity College is a coeducational, independent, nonsectarian liberal arts college located in, and deeply engaged with, Connecticut's capital city of Hartford. Our approximately 2,200 students come from all socioeconomic, racial, religious, and ethnic backgrounds across the United States, and seventeen percent are international. We emphasize excellence in both teaching and research, and our intimate campus provides an ideal setting for interdisciplinary collaboration. Teaching load is four courses per year for the first two years and five courses per year thereafter, with a one-semester leave every four years. We offer a competitive salary and benefits package, plus a start-up expense fund. For information about the Computer Science Department, visit: http://www.cs.trincoll.edu/.

Applicants should submit a curriculum vitae and teaching and research statements and arrange for three letters of reference to be sent to: https://trincoll.peopleadmin.com/.

Consideration of applications will begin on December 15, 2018, and continue until the position is filled.

Trinity College is an Equal-Opportunity/Affirmative-Action employer.

Women and members of minority groups are encouraged to apply.

The University of Alabama in Huntsville

Assistant Professor

The Department of Computer Science at The University of Alabama in Huntsville (UAH) invites applicants for a tenure-track faculty position at the Assistant Professor level beginning August 2019. All applicants with a background in traditional areas of computer science will be considered; however, special emphasis will be given to applicants with expertise in cybersecurity, software engineering, cloud computing, and systems related areas.

A Ph.D. in computer science or a closely related area is required. The successful candidate will have a strong academic background and be able to secure and perform funded research in areas typical for publication in well-regarded academic conference and journal venues. In addition, the candidate should embrace the opportunity to provide undergraduate education.

The department has a strong commitment to excellence in teaching, research, and service; the candidate should have good communication skills, strong teaching potential, and research accomplishments.

UAH is located in an expanding, high technology area, in close proximity to Cummings Research Park, the second largest research park in the nation and the fourth largest in the world. Nearby are the NASA Marshall Space Flight Center, the Army's Redstone Arsenal, numerous Fortune 500 and high tech companies. UAH also has an array of research centers, including information technology and cybersecurity. In short, collaborative research opportunities are abundant, and many well-educated and highly technically skilled people are in the area. There is also access to excellent public schools and inexpensive housing.

UAH has an enrollment of approximately 9,500 students. The Computer Science department offers BS, MS, and PhD degrees in Computer Science and contributes to interdisciplinary degrees. Faculty research interests are varied and include cybersecurity, mobile computing, data science, software engineering, visualization, graphics and game computing, multimedia, AI, image processing, pattern recognition, and distributed systems. Recent NSF figures indicate the university ranks 30th in the nation in overall federal research funding in computer science.

Interested parties must submit a detailed resume with references to info@cs.uah.edu or Chair, Search Committee, Dept. of Computer Science, The University of Alabama in Huntsville, Huntsville, AL 35899. Qualified female and minority candidates are encouraged to apply. Initial review of applicants will begin as they are received and continue until a suitable candidate is found.

The University of Alabama in Huntsville is an affirmative action / equal opportunity employer / minorities/ females / veterans / disabled.

Please refer to log number: 19/20 - 538

University of Illinois at Urbana-Champaign

Positions in Computing

The Department of Electrical and Computer Engineering (ECE ILLINOIS) at the University of Illinois at Urbana-Champaign invites applications for faculty positions at all areas and levels in computing, broadly defined, with particular emphasis on Embedded Computing Systems and the Internet of Things; Data-Centric Computing Systems and Storage; Networked and Distributed Computing Systems; AI/Autonomous Systems; Robotics; Machine Vision; Quantum Computing. Applications are encouraged from candidates whose research programs specialize in core as well as interdisciplinary areas of electrical and computer engineering. Ideal candidates include those who demonstrate evidence of a commitment to diversity, equity, and inclusion through research, teaching, and/or service endeavors.

From the transistor and the first computer implementation based on von Neumann's architecture to the Blue Waters petascale computer (the fastest computer on any university campus), ECE ILLINOIS has always been at the forefront of computing research and innovation. ECE ILLINOIS is in a period of intense demand and growth, serving over 3000 students and averaging 7 new tenure-track faculty hires per year in recent years. It is housed in its new 235,000 sq. ft. net-zero energy design building, which is a major campus addition with maximum space and minimal carbon footprint.

Qualified senior candidates may also be considered for tenured full Professor positions as part of the Grainger Engineering Breakthroughs Initiative (graingerinitiative.engineering.illinois.edu), which is backed by a $100-million gift from the Grainger Foundation.

Please visit http://jobs.illinois.edu to view the complete position announcement and application instructions. Full consideration will be given to applications received by December 1, 2018, but applications will continue to be accepted until all positions are filled.

Illinois is an EEO Employer/Vet/Disabled www.inclusiveillinois.illinois.edu.

The University of Illinois conducts criminal background checks on all job candidates upon acceptance of a contingent offer.

University of Memphis
Department of Computer Science
Assistant Professors

The Department of Computer Science at the University of Memphis is seeking candidates for multiple Assistant Professor positions beginning Fall 2019. Exceptionally qualified candidates in all areas of computer science are invited while candidates with core expertise in cyber-human systems (including computer vision, speech recognition, computer graphics, and human computer interaction (HCI)) and CS education, are particularly encouraged to apply. Candidates from minority and underrepresented groups are highly encouraged to apply. Successful candidates are expected to develop externally sponsored research programs, teach both undergraduate and graduate courses and provide academic advising to students at all levels.

Applicants should hold a PhD in Computer Science, or related discipline, and be committed to excellence in both research and teaching. Salary is highly competitive and dependent upon qualifications.

The Department of Computer Science (http://www.memphis.edu/cs/) offers B.S., M.S., and Ph.D. programs as well as graduate certificates in Data Science and Information Assurance, and participates in an M.S. program in Bioinformatics (through the College of Arts and Sciences). The Department has been ranked 55th among CS departments with federally funded research. The Department regularly engages in large-scale multi-university collaborations across the nation. For example, CS faculty lead the NIH-funded Big Data "Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K)" and the "Center for Information Assurance (CfIA)". In addition, CS faculty work closely with multidisciplinary centers at the university such as the "Institute for Intelligent Systems (IIS)".

Known as America's distribution hub, Memphis ranked as America's 6th best city for jobs by Glassdoor in 2017. Memphis metropolitan area has a population of 1.3 million. It boasts a vibrant culture and has a pleasant climate with an average temperature of 63 degrees.

Screening of applications begins immediately. For full consideration, application materials should be received by November 25, 2018. However, applications will be accepted until the search is completed.

To apply, please visit https://workforum.memphis.edu/. Include a cover letter, curriculum vitae, statement of teaching philosophy, research statement, and three letters of recommendation. Direct all inquiries to Corinne O'Connor (cconnor2@memphis.edu).

A background check will be required for employment. The University of Memphis is an Equal Opportunity/Equal Access/Affirmative Action employer committed to achieving a diverse workforce.

University of Michigan
Multiple Tenure-Track and Teaching Faculty Positions

Computer Science and Engineering (CSE) at the University of Michigan invites applications for multiple tenure-track and teaching faculty (lecturer) positions. We seek exceptional candidates at all levels in all areas across computer science and computer engineering. We also have a targeted search for an endowed professorship in theoretical computer science (the Fischer Chair). Qualifications include an outstanding academic record, a doctorate or equivalent in computer science or computer engineering, and a strong commitment to teaching and research. Candidates are expected, through their research, teaching, and/or service, to contribute to the diversity and excellence of the academic community.

The University of Michigan is one of the

上海科技大学
ShanghaiTech University

TENURE-TRACK AND TENURED POSITIONS

ShanghaiTech University invites highly qualified candidates to fill multiple tenure-track/tenured faculty positions as its core founding team in the School of Information Science and Technology (SIST). We seek candidates with exceptional academic records or demonstrated strong potentials in all cutting-edge research areas of information science and technology. They must be fluent in English. English-based overseas academic training or background is highly desired.

ShanghaiTech is founded as a world-class research university for training future generations of scientists, entrepreneurs, and technical leaders. Boasting a new modern campus in Zhangjiang Hightech Park of cosmopolitan Shanghai, ShanghaiTech shall trail-blaze a new education system in China. Besides establishing and maintaining a world-class research profile, faculty candidates are also expected to contribute substantially to both graduate and undergraduate educations.

Academic Disciplines: Candidates in all areas of information science and technology shall be considered. Our recruitment focus includes, but is not limited to: computer architecture, software engineering, database, computer security, VLSI, solid state and nano electronics, RF electronics, information and signal processing, networking, security, computational foundations, big data analytics, data mining, visualization, computer vision, bio-inspired computing systems, power electronics, power systems, machine and motor drive, power management IC as well as inter-disciplinary areas involving information science and technology.

Compensation and Benefits: Salary and startup funds are highly competitive, commensurate with experience and academic accomplishment. We also offer a comprehensive benefit package to employees and eligible dependents, including on-campus housing. All regular ShanghaiTech faculty members will join its new tenure-track system in accordance with international practice for progress evaluation and promotion.

Qualifications:
- Strong research productivity and demonstrated potentials;
- Ph.D. (Electrical Engineering, Computer Engineering, Computer Science, Statistics, Applied Math, Artificial Intelligence, Statistics or related field);
- A minimum relevant (including PhD) research experience of 4 years.

Applications: Submit (in English, PDF version) a cover letter, a 2-page research plan, a CV plus copies of 3 most significant publications, and names of three referees to: sist@shanghaitech.edu.cn. For more information, visit http://sist.shanghaitech.edu.cn/NewsDetail.asp?id=373

Deadline: The positions will be open until they are filled by appropriate candidates.

Penn Engineering

MULTIPLE FACULTY POSITIONS
Department of Electrical and Systems Engineering

The School of Engineering and Applied Science at the University of Pennsylvania is growing its faculty by 33% over the next five years. As part of this initiative, the **Department of Electrical and Systems Engineering** is engaged in an aggressive, multi-year hiring effort for multiple tenure-track positions at all levels. Candidates must hold a Ph.D. in Electrical Engineering, Computer Engineering, Systems Engineering, or related area. The department seeks individuals with exceptional promise for, or proven record of, research achievement, who will take a position of international leadership in defining their field of study, and excel in undergraduate and graduate education. Leadership in cross-disciplinary and multi-disciplinary collaborations is of particular interest. We are interested in candidates in all areas that enhance our research strengths in

1. **Nanodevices and nanosystems** (nanoelectronics,MEMS/NEMS, power electronics, nanophotonics, nanomagnetics, quantum devices, integrated devices and systems at nanoscale),
2. **Circuits and computer engineering** (analog, RF, mm-wave, digital circuits,emerging circuit design, computer engineering, IoT, embedded and cyber-physical systems), and
3. **Information and decision systems** (control, optimization, robotics, data science, network science, communications, information theory, signal processing).

Prospective candidates in all areas are strongly encouraged to address large-scale societal problems in energy, transportation, health, food and water, economic and financial networks, social networks, critical infrastructure, and national security. We are especially interested in candidates whose interests are aligned with the school's strategic plan, ***https://www.seas.upenn.edu/about/penn-engineering-2020/***

Diversity candidates are strongly encouraged to apply. Interested persons should submit an online application at ***https://www.ese.upenn.edu/faculty-staff/*** and include curriculum vitae, statement of research and teaching interests, and at least three references. Review of applications will begin on December 1, 2018.

The University of Pennsylvania is an Equal Opportunity Employer.
Minorities/Women/Individuals with Disabilities/Veterans are encouraged to apply.

world's leading research universities, consisting of highly ranked departments and colleges across engineering, sciences, medicine, law, business, and the arts. CSE is a vibrant and innovative community, with over 70 world-class faculty members, over 300 graduate students, and a large and illustrious network of alumni. Ann Arbor is known as one of the best small cities in the country, offering cosmopolitan living without the hassle. The University of Michigan has a strong dual-career assistance program.

We encourage candidates to apply as soon as possible. For best consideration for Fall 2019, please apply by December 1, 2018. Positions remain open until filled and applications can be submitted throughout the year.

For more details on these positions and to apply, please visit http://cse.umich.edu/jobs.

Michigan Engineering's vision is to be the world's preeminent college of engineering serving the common good. This global outlook, leadership focus, and service commitment permeate our culture. Our vision is supported by a mission and values that, together, provide the framework for all that we do. Information about our vision, mission and values can be found at: http://strategicvision.engin.umich.edu/.

The University of Michigan has a storied legacy of commitment to Diversity, Equity and Inclusion (DEI). The Michigan Engineering component of the University's comprehensive, five-year, DEI strategic plan—with updates on our programs and resources dedicated to ensuring a welcoming, fair, and inclusive environment—can be found at: http://www.engin.umich.edu/college/about/diversity.

The University of Michigan is a Non-Discriminatory/Affirmative Action Employer.

The University of Texas at San Antonio (UTSA)
Department Chair

The Department of Computer Science at the University of Texas at San Antonio (UTSA) is seeking a dynamic Department Chair that can lead a department of preeminence in an extraordinary diverse University that is focused on a significant expansion of its research mission.

The Department seeks exceptional candidates with (1) a record of high quality scholarship and competitive research with federal, state, and industry funding, (2) experience and leadership in institutions of higher education, industry, or professional organizations, (3) an understanding of pedagogies that will lead to student success and excellence in undergraduate and graduate teaching, (4) experience leading interdisciplinary teams, and (5) mentorship experience and a commitment to inclusion and diversity.

The University of Texas at San Antonio is designated a National Center of Academic Excellence in Cyber Operations and has just been approved for $70 million in funding to construct two new facilities – A National Security Collaboration Center and a proposed School of Data Science. The Computer Science Department has 23 full-time faculty, 8 full-time lecturers, 1,300 undergraduate students, 70 M.S., and 60 Ph.D. students.

The successful candidate must have a doctorate in computer science or closely related field, with outstanding research and teaching records that warrant an appointment at the rank of full professor with tenure. Tenure is contingent upon Board of Regents approval.

See http://apptrkr.com/1295200 for information on the Department and application instructions.

Screening of applications will begin on November 15, 2018. The search will continue until the position is filled or the search is closed.

The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer. Women, minorities, veterans, and individuals with disabilities are encouraged to apply.

The University of Texas at San Antonio (UTSA)
Faculty Position in Computer Science

The Department of Computer Science at The University of Texas at San Antonio (UTSA) invites applications for one tenure-track or tenured open rank (Assistant, Associate or Full Professor) position, starting in Fall 2019. This position is targeted towards faculty with expertise and interest in artificial intelligence (AI). Outstanding candidates from all areas of AI will be considered, and preference will be given to applicants with expertise in cyber adversarial learning, AI for resource-constrained systems (such as IoTs and embedded systems), or AI (such as natural language processing, computer vision and deep learning) as it relates to health-related applications. This position is part of the university-wide cluster hiring in Artificial Intelligence.

See http://www.cs.utsa.edu/fsearch for information on the Department and application instructions. Screening of applications will begin immediately.

Application received by January 2, 2019 will be given full consideration. The search will continue until the positions are filled or the search is closed. The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer. Women, minorities, veterans, and individuals with disabilities are encouraged to apply.

Department of Computer Science
RE: Faculty Search
The University of Texas at San Antonio
One UTSA Circle
San Antonio, TX 78249-0667
Phone: 210-458-4436

University of Toronto
Assistant Professor, Tenure Stream

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering (ECE) at the University of Toronto invites applications for up to four full-time tenure-stream faculty appointments at the rank of Assistant Professor. The appointments will commence on July 1, 2019.

Within the general field of electrical and computer engineering, we seek applications from candidates with expertise in one or more of the following strategic research areas: 1. Computer Systems and Software; 2. Electrical Power Systems; 3. Systems Control, including but not limited to autonomous and robotic systems.

Applicants are expected to have a Ph.D. in Electrical and Computer Engineering, or a related field, at the time of appointment or soon after.

Successful candidates will be expected to initiate and lead an outstanding, innovative, independent, competitive, and externally funded research program of international calibre, and to teach at both the undergraduate and graduate levels. Candidates should have demonstrated excellence in research and teaching. Excellence in research is evidenced primarily by publications or forthcoming publications in leading journals or conferences in the field, presentations at significant conferences, awards and accolades, and strong endorsements by referees of high international standing. Evidence of excellence in teaching will be demonstrated by strong communication skills; a compelling statement of teaching submitted as part of the application highlighting areas of interest, awards and accomplishments, and teaching philosophy; sample course syllabi and materials; and teaching evaluations, as well as strong letters of recommendation.

Eligibility and willingness to register as a Professional Engineer in Ontario is highly desirable.

Salary will be commensurate with qualifications and experience.

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto ranks among the best in North America. It attracts outstanding students, has excellent facilities, and is ideally located in the middle of a vibrant, artistic, diverse and cosmopolitan city.

Additional information may be found at http://www.ece.utoronto.ca.

Review of applications will begin after September 1, 2018, however, the position will remain open until November 29, 2018.

As part of your online application, please include a cover letter, a curriculum vitae, a summary of your previous research and future research plans, as well as a teaching dossier including a statement of teaching experience and interests, your teaching philosophy and accomplishments, and teaching evaluations. Applicants must arrange for three letters of reference to be sent directly by the referees (on letterhead, signed and scanned), by email to the ECE department at search2018@ece.utoronto.ca. Applications without any reference letters will not be considered; it is your responsibility to make sure your referees send us the letters while the position remains open.

You must submit your application online while the position is open, by following the submission guidelines given at http://uoft.me/how-to-apply. Applications submitted in any other way will not be considered. We recommend combining attached documents into one or two files in PDF/MS Word format. If you have any questions about this position, please contact the ECE department at search2018@ece.utoronto.ca.

The University of Toronto is strongly committed to diversity within its community and especially welcomes applications from racialized persons / persons of colour, women, Indigenous / Aboriginal People of North America, persons with disabilities, LGBTQ persons, and others who may contribute to the further diversification of ideas.

As part of your application, you will be asked to complete a brief Diversity Survey. This survey is voluntary. Any information directly related to you is confidential and cannot be accessed by search committees or human resources staff. Results will be aggregated for institutional planning purposes. For more information, please see http://uoft.me/UP.

All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

University of Toronto
Assistant Professor, Teaching Stream

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering (ECE) at the University of Toronto invites applications for a full-time teaching-stream faculty appointment at the rank of Assistant Professor, Teaching Stream, in the general area of Computer Systems and Software. The appointment will commence on July 1, 2019.

Applicants are expected to have a Ph.D. in Electrical and Computer Engineering, or a related field, at the time of appointment or soon after.

Successful candidates will have demonstrated excellence in teaching and pedagogical inquiry, including in the development and delivery of undergraduate courses and laboratories and supervision of undergraduate design projects. This will be demonstrated by strong communication skills, a compelling statement of teaching submitted as part of the application highlighting areas of interest, awards and accomplishments and teaching philosophy; sample course syllabi and materials; and teaching evaluations, as well as strong letters of reference from referees of high standing endorsing excellent teaching and commitment to excellent pedagogical practices and teaching innovation.

Eligibility and willingness to register as a Professional Engineer in Ontario is highly desirable.

Salary will be commensurate with qualifications and experience.

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto ranks among the best in North America. It attracts outstanding students, has excellent facilities, and is ideally located in the middle of a vibrant, artistic, diverse and cosmopolitan city. Additional information may be found at http://www.ece.utoronto.ca.

Review of applications will begin after September 1, 2018, however, the position will remain open until November 29, 2018.

As part of your online application, please include a cover letter, a curriculum vitae, and a teaching dossier including a summary of your previous teaching experience, your teaching philosophy and accomplishments, your future teaching plans and interests, sample course syllabi and materials, and teaching evaluations. Applicants must arrange for three letters of reference to be sent directly by the referees (on letterhead, signed and scanned), by email to the ECE department at search2018@ece.utoronto.ca. Applications without any reference letters will not be considered; it is your responsibility to make sure your referees send us the letters while the position remains open.

You must submit your application online while the position is open, by following the submission guidelines given at http://uoft.me/how-to-apply. Applications submitted in any other way will not be considered. We recommend combining attached documents into one or two files in PDF/MS Word format. If you have any questions about this position, please contact the ECE

Exploring new frontiers in computer science.
Mathematics of Big Data
Spreadsheets, Databases, Matrices, and Graphs
Jeremy Kepner and Hayden Jananthan
Foreword by Charles E. Leiserson
BITS to BITCOIN
HOW OUR DIGITAL STUFF WORKS
MARK STUART DAY
THE PROBLEM WITH SOFTWARE
WHY SMART ENGINEERS WRITE BAD CODE
ADAM BARR
ROBOTICS THROUGH SCIENCE FICTION
Artificial Intelligence Explained Through Six Classic Robot Short Stories
edited by ROBIN R. MURPHY
mitpress.mit.edu/ACM

department at search2018@ece.utoronto.ca.

The University of Toronto is strongly committed to diversity within its community and especially welcomes applications from racialized persons / persons of colour, women, Indigenous / Aboriginal People of North America, persons with disabilities, LGBTQ persons, and others who may contribute to the further diversification of ideas.

As part of your application, you will be asked to complete a brief Diversity Survey. This survey is voluntary. Any information directly related to you is confidential and cannot be accessed by search committees or human resources staff. Results will be aggregated for institutional planning purposes. For more information, please see http://uoft.me/UP.

All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

University of Toronto
Associate Professor, Tenure Stream

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering (ECE) at the University of Toronto invites applications for up to four full-time tenure-stream faculty appointments at the rank of Associate Professor. The appointments will commence on July 1, 2019.

Within the general field of electrical and computer engineering, we seek applications from candidates with expertise in one or more of the following strategic research areas: 1. Computer Systems and Software; 2. Electrical Power Systems; 3. Systems Control, including but not limited to autonomous and robotic systems.

Applicants are expected to have a Ph.D. in Electrical and Computer Engineering, or a related field, and have at least five years of academic or relevant industrial experience.

Successful candidates will be expected to maintain and lead an outstanding, independent, competitive, innovative, and externally funded research program of international calibre, and to teach at both the undergraduate and graduate levels. Candidates should have demonstrated excellence in research and teaching. Excellence in research is evidenced primarily by sustained and impactful publications in leading journals or conferences in the field, awards and accolades, presentations at significant conferences and a high profile in the field with strong endorsements by referees of high international standing. Evidence of excellence in teaching will be demonstrated by strong communication skills, a compelling statement of teaching submitted as part of the application highlighting areas of interest, awards and accomplishments, and teaching philosophy; sample course syllabi and materials; and teaching evaluations, as well as strong letters of recommendation.

Eligibility and willingness to register as a Professional Engineer in Ontario is highly desirable.

Salary will be commensurate with qualifications and experience.

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto ranks among the best in North America. It attracts outstanding students, has excellent facilities, and is ideally located in the middle of a vibrant, artistic, diverse and cosmopolitan city. Additional information may be found at http://www.ece.utoronto.ca.

Review of applications will begin after September 1, 2018, however, the position will remain open until November 29, 2018.

As part of your online application, please include a cover letter, a curriculum vitae, a summary of your previous research and future research plans, as well as a teaching dossier including a statement of teaching experience and interests, your teaching philosophy and accomplishments, and teaching evaluations. Applicants must arrange for three letters of reference to be sent directly by the referees (on letterhead, signed and scanned), by email to the ECE department at search2018@ece.utoronto.ca. Applications without any reference letters will not be considered; it is your responsibility to make sure your referees send us the letters while the position remains open.

You must submit your application online while the position is open, by following the submission guidelines given at http://uoft.me/how-to-apply. Applications submitted in any other way will not be considered. We recommend combining attached documents into one or two files in PDF/MS Word format. If you have any questions about this position, please contact the ECE department at search2018@ece.utoronto.ca.

The University of Toronto is strongly committed to diversity within its community and especially welcomes applications from racialized persons / persons of colour, women, Indigenous / Aboriginal People of North America, persons with disabilities, LGBTQ persons, and others who may contribute to the further diversification of ideas.

As part of your application, you will be asked to complete a brief Diversity Survey. This survey is voluntary. Any information directly related to you is confidential and cannot be accessed by search committees or human resources staff. Results will be aggregated for institutional planning purposes. For more information, please see http://uoft.me/UP.

All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

University of Zurich
Professorship in Human and Crowd Computing (Assistant/Associate/Full)

The Faculty of Business, Economics and Informatics of the University of Zurich invites applications for a Professorship in Human and Crowd Computing (Assistant/Associate/Full) starting in fall 2019.

Candidates should hold a PhD in Computer Science, Informatics or a related discipline, and have an excellent research record in the area of "Human and Crowd Computing," ideally with a focus in one or multiple of the following subareas: Social Computing, Human Computation, Collective Intelligence, Human-Agent Interaction, Computer-Supported Cooperative Work, Incentive Design/Mechanism Design, Information Elicitation, and the Design/Analysis of Social Platforms. We expect the candidate to be committed to excellence in teaching both at the undergraduate and the graduate levels. German language skills are not required.

The successful candidate is expected to conduct high-quality research within the area of Human and Crowd Computing and establish her or his research group within the Department of Informatics and internationally. The successful candidate is also expected to actively interface with the other groups at the department and the faculty, and seek collaborations with researchers across faculties within the Digital Society Initiative of the University of Zurich as well as internationally.

The Faculty of Business, Economics and Informatics offers a stimulating research environment and rich opportunities for collaboration. The Human and Crowd Computing professorship is another step towards establishing the strengths of the Department of Informatics within its three focus areas of people-oriented computing, computing and economics, and big data analytics.

The University of Zurich is one of the leading research universities in Europe and offers the widest range of study courses in Switzerland to over 26,000 students. Through its educational and research objectives, the University of Zurich aims at attracting leading international researchers who are willing to contribute to its development and to strengthening its reputation. The University of Zurich is an equal opportunity employer and strongly encourages applications from female candidates.

Please submit your application, including a CV, contact information for at least three references, three papers (published or unpublished), and a record of teaching effectiveness (taught courses and evaluations) via https://www.facultyhiring.oec.uzh.ch/position/10092545 before 1st November, 2018.

Documents should be addressed to Prof. Dr. Harald Gall; Dean of the Faculty of Business, Economics and Informatics; University of Zurich; Switzerland.

For questions regarding the open position please contact Prof. Dr. Thomas Fritz (fritz@ifi.uzh.ch) or Prof. Dr. Abraham Bernstein (bernstein@ifi.uzh.ch) or Prof. Dr. Sven Seuken (seuken@ifi.uzh.ch).

US Air Force Academy
Assistant Professor of Computer Science

The Department of Computer Science at the US Air Force Academy seeks to fill up to two faculty positions at the Assistant Professor level. The department is particularly interested in candidates with backgrounds in artificial intelligence, computer and network security, operations research, or unmanned aerial systems, but all candidates with a passion for undergraduate computer science teaching are encouraged to apply.

The Academy is a national service institution, charged with producing lieutenants for the US Air Force. Faculty members are expected to exemplify the highest ideals of professionalism and character. USAFA is located in Colorado Springs, an area known for its exceptional natural beauty and quality of life. The United States Air Force Academy values the benefits of diversity among the faculty to include a variety of educational backgrounds and professional and life experiences.

For information on how to apply, go to usajobs.gov and search with the keyword 509715400.

[CONTINUED FROM P. 176] the camera. And next to her … I squinted to see better … hand on her waist, was *me*. I gasped. Was this proof that in the future I would build a machine to travel back in time? How else could I appear in an image produced in the 19th century?

Moments later I was backpedaling. Plenty of Englishmen had beards at the time. It could be a passing resemblance. But genuine or not, the photograph inspired me to recall the field equations of the general theory. Going forward in time is easy; we do it every moment of every day. Special relativity tells us all we need to do is move … and the faster we go, the quicker we get to the future. But traveling back is far more complicated—even theoretically. Ideas for backward time travel usually involve an effect called "frame dragging," whereby rotating masses like black holes twist the fabric of space-time as if they were spoons in honey until time loops back on itself.

I realized that to make backward time travel practical I might try another oddity of relativity, that energy, like mass, produces gravity. If I could pick up on the quantum fluctuations of energy in empty space I could set up a feedback loop whereby the energy produced gravity, feeding back to produce yet more energy … warping space-time sufficiently to make the journey *without* a black hole. But there was a problem. As the great physicist Richard Feynman pointed out in 1982, you can only fully simulate quantum systems with a quantum computer. Without quantum computing, the process would be impossible to control—and at the time of my first visit, quantum computing was little more than a theoretical concept.

Still, that old photograph gave me hope that in the future I might yet use such a computer to travel into the past. To test my hypothesis, I needed a prediction I could check. I'd explored the history and science of Fox Talbot's earliest negative many times. It featured no human figures, but I knew exactly where and when it was taken. So, if I could build a time machine, I'd make sure I was outside the window at precisely the moment Fox Talbot exposed his chemically sensitive paper. If I now looked at the negative and I was in it, holding some clearly identifiable object, say, star-shaped—the first shape that occurred to me—I would *know* the mechanism worked.

If I fail, I'll simply cease to exist.

I had to see the image again, recalling it was in Fox Talbot's house, by the window where it had been taken. As I hurried from the museum onto the curving driveway, I caught a glimpse of that jewel of a building—Lacock Abbey—glowing in the sunlight. I went round the corner, momentarily losing it in a clump of trees, passing a small, surprisingly lifelike statue of an Egyptian sphinx. Round another corner and up a shallow flight of steps leading into a vaulted hall. I couldn't allow myself to pause to enjoy a notice saying NO PHOTOGRAPHY, here of all places, continuing through the dining room into the abbey's South Gallery.

It was the smallest center oriel window that provided Fox Talbot's subject. When I arrived, a lively discussion was under way between the room's guide and a lone visitor, with the guide pointing out that the gallery was narrow, making it difficult for Fox Talbot to project an image. But the visitor paid little attention, peering instead through the window, taking in the same view as in the photograph, now nearly 200 years on.

"Were those trees there when Fox Talbot took the picture?" he asked the guide. The windowpane had the faded translucency of age, making it difficult to see out. "You see," he said, "it's this blob." We stepped up to the enlarged version of the Fox Talbot negative framed on the gallery wall. Sure enough, I could make out a shape on its right-hand side. My heart leaped. "I've always wondered what the blob on the right is," said the visitor.

I knew there had never been a blob in the photograph. Until now …

"Someone could have chopped down a tree," said the guide helpfully.

I approached the negative, restraining myself from touching it and leaving a mark. "Is it a *person*?" I said. "Could the blob be a person?" As I looked more closely, I could make out some detail. It was surely a bearded man. And in his right hand he was holding (my stomach clenched) a star.

It was going to happen. Here was the evidence I could build my machine to harness the feedback loop and travel back. Desperate to channel my thoughts, I hurried back to the hire car where I'd left my notebook. The exit route took me though a dark part of the old abbey, a dismal, dusty place with empty stone coffins lined up on the floor as if waiting for their future occupants. It was here I began to ask myself whether quantum computers would ever be capable of such sophistication in my lifetime. Had I imagined everything? Instead of heading for the car, I hurried back to the museum … to see again the first photograph in which I'd appeared.

There they stood, as they always had, the group in front of the house. I recognized the woman's impish smile. But the man next to her, hand on her waist, was nothing like me, but shorter, rounder, clean-shaven.

So each year I return, hoping the latest developments in quantum computing are about to make my kind of time travel possible. I run my simulations and, prompted by my app, time myself and count my steps, walk different routes, pause different lengths of time at each exhibit, yet fail to appear in the images.

But now I've found records of an ancestor of mine who lived near Lacock in Fox Talbot's time, sharing *my* name and birthday. If I'm in that photograph again, with my hand on the woman's waist—I'll have travelled back to become my own great, great, great grandfather. If I fail, I'll simply cease to exist. I'll keep uncertainty from my mind and fix the reality of the future and the past, like Fox Talbot developing a photographic print. I'm sure I can do it. It's only a matter of time.

Brian Clegg (www.brianclegg.net) is a science writer based in the U.K. His most recent books are *Are Numbers Real?*, an exploration of the relationship between math and reality, and *The Reality Frame*, an exploration of relativity and frames of reference.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/3280370

Brian Clegg

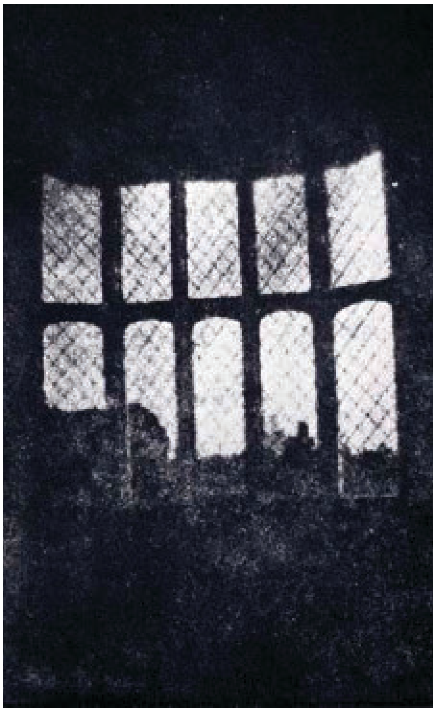
Future Tense
Between the Abbey and the Edge of Time

A photo marks my place, then and now.

IT'S MY 25TH year visiting Lacock Abbey in Wiltshire, England, arriving early as usual. As I sit in the hire car, I replay in my mind the day I realized how I might manipulate time. Ever since my doctorate in astrophysics, I'd been researching the equations of general relativity, focusing on the crossover between relativity and the quantum world. But my true interest was "closed timelike loops"—what some might call a time machine.

It was here, 25 years prior, I had worked out how a quantum computer could use a loophole in Einstein's gravitational field equations to make time travel possible. I was in England for a conference on quantum gravity and took the opportunity to visit the home of a personal hero, 19th-century photography pioneer William Henry Fox Talbot. Every moment remains etched in my memory and I have tried to repeat them each year since.

The first anniversary I brought a sheaf of printer output from Monte Carlo simulations run on a Thinking Machines CM-5 in Cambridge, permutating the variables I could alter. A step here, a pause there, attempting to recreate what happened that day. Each year since I have tried as many walkthrough variants as I could. And each year I have failed. I need to use the rest of my time keeping my career alive, so I limit myself to the anniversary. Now I've written a smartphone app to guide me, but the approach is the same, with each detail recreated as best I can. But this year has to be different.



Oriel window in Lacock Abbey photographed by William Fox Talbot in 1835.

Special relativity tells us all we need to do is move ... and the faster we go, the quicker we get to the future.

I began that first visit at the Fox Talbot museum near the abbey. The exhibits led up to the moment on Lake Como in 1834 when Fox Talbot had his inspiration. He enjoyed clever toys, trendy optical devices like the *camera lucida* for projecting an image onto sketching paper to guide an artist. I'd like to say, cue the light bulb over Fox Talbot's head, except electric light hadn't been invented yet. Fox Talbot was studiously tracing an image. Yet he knew that compounds of silver darkened when exposed to light. So why not soak the paper in silver salts and let the drawing produce itself?

In August 1835, Fox Talbot set up a *camera obscura*, projecting an image of a window in the South Gallery of Lacock Abbey onto treated paper. The result (or at least a copy) is on the wall beside the window where the picture was taken. It's tiny—only 1.2 inches by one inch—the world's first photographic negative. Other early processes produced one-off images, but Fox Talbot's negatives provided unlimited prints. Forget the idea that the Victorian information age started with Charles Babbage's mechanical computers. What Fox Talbot invented was *visual* information processing, but his bits were silver crystals, in a mechanism that became the mainstay of photography for over a century and a half.

A display of photographs from Fox Talbot's time stood near the exit. That was where my time journey began. One showed a group outside a country cottage. Most were stiffly Victorian, but one, a young woman, smiled engagingly at

[CONTINUED ON P. 175]

IMAGE BY WILLIAM FOX TALBOT, FROM WIKIPEDIA.ORG CC-PD-MARK

ETRA

ACM SYMPOSIUM ON EYE TRACKING RESEARCH & APPLICATIONS

June 25-28, 2019
Crowne Plaza
1450 Glenarm Place
Denver, Colorado

The ETRA conference series focuses on eye movement research & applications across a wide range of disciplines including computer science, human-computer interaction, visualization, biomedical research, virtual reality & psychology.

Join us in Denver to celebrate another year of eye tracking research!

Important dates

Papers & Notes

Dec 14, 2018	Paper abstracts due
Dec 19, 2018	Long & short papers due
Jan 23, 2019	Reviews due to authors
Jan 28, 2019	Rebuttals + revised papers due
Feb 18, 2019	Final notifications to authors
Mar 01, 2019	Camera ready papers due

Demo/Video & Doctoral Symposium

Mar 08, 2019	Extended abstracts due
Mar 15, 2019	Notifications due to authors
Mar 22, 2019	Camera ready papers due

Challenge Track

Mar 15, 2019	Challenge report due
Mar 29, 2019	Notifications due to authors
Apr 05, 2019	Camera ready papers due

General Chairs: Bonita Sharif (University of Nebraska, Lincoln)
& Krzysztof Krejtz (SWPS University of Social Sciences and Humanities, Poland)

@ETRAConference @ETRAConference WEBSITE: http://etra.acm.org

Association for Computing Machinery SIGCHI ACMSIGGRAPH

SIGGRAPH ASIA 2018 TOKYO

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan

CROSSOVER

The 11th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia

Platinum Sponsor

FORUM 8®

Gold Sponsors

CyberAgent

dhu

IMAGICA GROUP

LUMINOUS PRODUCTIONS

unity

Bronze Sponsors

BANDAI NAMCO
BANDAI NAMCO Studios

DeNA

KHRONOS GROUP

清华大学-腾讯联合实验室
TSINGHUA UNIVERSITY- TENCENT JOINT LABORATORY

Sponsored by

acm

REGISTER ONLINE TODAY!
sa2018.siggraph.org/registration

Organized by

koelnmesse