COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

08/2018 VOL.61 NO.08

Multiparty Privacy in Social Media

Point/Counterpoint
Democracy and E-Democracy

How to Teach Computer Ethics through Science Fiction

Regulating Automated Decision Making

Association for Computing Machinery

SIGGRAPH ASIA 2018 TOKYO

CROSSOVER

The 11th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia

CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan

SA2018.SIGGRAPH.ORG

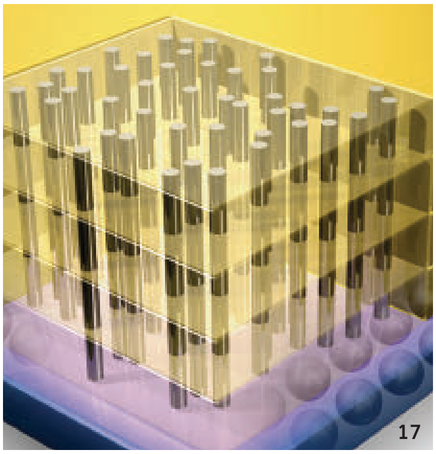
COMMUNICATIONS OF THE ACM

Departments

Last Byte

Association for Computing Machinery
Advancing Computing as a Science & Profession

News


17

Viewpoints

31

(L) IMAGE COURTESY OF NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY; (R) PAINTING BY JEAN-JACQUES-FRANÇOIS LE BARBIER

08/2018
VOL. 61 NO. 08

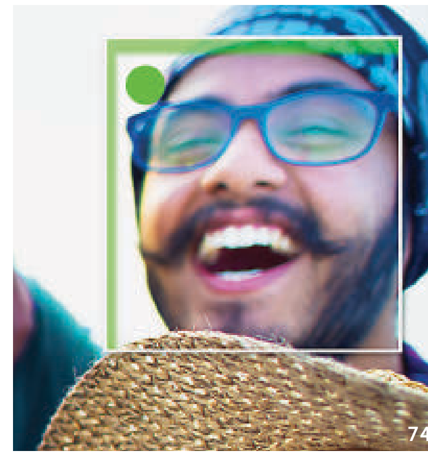
Practice

Articles' development led by acmqueue
queue.acm.org

Contributed Articles

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/how-to-teach-computer-ethics-with-science-fiction

Review Articles



Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/multiparty-privacy-in-social-media

Research Highlights

About the Cover:
Mulitparty privacy, as the phrase suggests, means different things to different parties. This month's cover story, by Jose Such and Natalia Criado (p. 74), traces the challenges of controlling privacy via social media when friends or family or associates have other plans. CGI cover illustration by Peter Crowther Associates; group shot by Rawpixel.com.

IMAGES BY (L) DARIUSH M/SHUTTERSTOCK; (C) 13_PHUNKOD/SHUTTERSTOCK; (R) CGI BY PETER CROWTHER ASSOCIATES; GROUP SHOT BY RAWPIXEL.COM

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.

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:10.1145/3231715

James Larus and Chris Hankin

Regulating Automated Decision Making

DISDAIN FOR REGULATION is pervasive throughout the tech industry. In the case of automated decision making, this attitude is mistaken. Early engagement with governments and regulators could both smooth the path of adoption for systems built on machine learning, minimize the consequences of inevitable failures, increase public trust in these systems, and possibly avert the imposition of debilitating rules.

Exponential growth in the sophistication and applications of machine learning is in the process of automating wholly or partially many tasks previously performed only by humans. This technology of automated decision making (ADM) promises many benefits, including reducing tedious labor as well as improving the appropriateness and acceptability of decisions and actions. The technology also will open new markets for innovative and profitable businesses, such as self-driving vehicles and automated services.

At the same time, however, the widespread adoption of ADM systems will be economically disruptive and will raise new and complex societal challenges, such as worker displacement; autonomous accidents; and, perhaps most fundamentally, confusion and debate over what it means to be human.

From a European perspective, this is a strong argument for governments to take a more active role in regulating the use of ADM. The European Union has already started to grapple with privacy concerns through the General Data Protection Regulation (GDPR), which regulates data protection and requires explanation of automated decisions involving people. However, widespread use of ADM will raise additional ethical, economic, and legal issues. Early attention to these questions is central to formulating regulation for autonomous vehicles. The German Ministry for Transport and Digital Infrastructure created an Ethics Commission, which identified 20 key principles to govern ethical and privacy concerns in automated driving.[a]

To raise these concerns more broadly, a group assembled by Informatics Europe and EUACM, the policy committee of the ACM Europe Council, recently produced a report entitled "When Computers Decide."[b] The white paper makes 10 recommendations to policy leaders:

1. Establish means, measures, and standards to assure ADM systems are fair.
2. Ensure ethics remain at the forefront of, and integral to, ADM development and deployment.
3. Promote value-sensitive ADM design.
4. Define clear legal responsibilities for ADM's use and impacts.
5. Ensure the economic consequences of ADM adoption are fully considered.
6. Mandate that all privacy and data acquisition practices of ADM deployers be clearly disclosed to all users of such systems.
7. Increase public funding for noncommercial ADM-related research significantly.
8. Foster ADM-related technical education at the university level.
9. Complement technical education with comparable social education.
10. Expand the public's awareness and understanding of ADM and its impacts.

Systems built on an immature and rapidly evolving technology such as machine learning will have spectacular successes and dismaying failures. Especially when the technology is used in applications that affect the safety and livelihood of many people, these systems should be developed and deployed with special care. Society must set clear parameters for what uses are acceptable, how the systems should be developed, how inevitable trade-offs and conflicts will be adjudicated, and who is legally responsible for these systems and their failures.

Automated decision making is not just a scientific challenge; it is simultaneously a political, economic, technological, cultural, educational, and even philosophical challenge. Because these aspects are interdependent, it is inappropriate to focus on any one feature of the much larger picture. The computing professions and technology industries, which together are driving these advances forward, have an obligation to start a conversation among all affected disciplines and institutions whose expertise is relevant and required to fully understand these complex issues.

Now is the time to formulate appropriately nuanced, comprehensive, and ethical plans for humans and our societies to thrive when computers make decisions. C

a https://www.bmvi.de/SharedDocs/EN/publications/report-ethics-commission.html
b https://dl.acm.org/citation.cfm?id=3185595

James Larus, a professor and Dean of the School of Computer and Communication Sciences at EPFL, is on the board of Informatics Europe.

Chris Hankin, chair of ACM Europe Council, is co-director of the Institute for Security Science and Technology and a professor of computing science at Imperial College London.

DOI:10.1145/3235764

Vinton G. Cerf

Traceability

At a recent workshop on cybersecurity at Ditchley House sponsored by the Ditchley Foundation in the U.K., a primary topic of consideration was how to preserve

the freedom and openness of the Internet while protecting against the harmful behaviors that have emerged in this global medium. That this is a significant challenge cannot be overstated. The bad behaviors range from social network bullying and misinformation to email spam, distributed denial of service attacks, direct cyberattacks against infrastructure, malware propagation, identity theft, and a host of other ills requiring a wide range of technical and legal considerations. That these harmful behaviors can and do cross international boundaries only makes it more difficult to fashion effective responses.

In other columns, I have argued for better software development tools to reduce the common mistakes that lead to vulnerabilities that are exploited. Here, I want to focus on another aspect of response related to law enforcement and tracking down perpetrators. Of course, not all harms are (or perhaps are not yet) illegal, but discovering those who cause them may still be warranted. The recent adoption and implementation of the General Data Protection Regulation (GDPR) in the European Union creates an interesting tension because it highlights the importance and value of privacy while those who do direct or indirect harm must be tracked down and their identities discovered.

In passing, I mention that cryptography has sometimes been blamed for protecting the identity or actions of criminals but it is also a tool for protecting privacy. Arguments have been made for "back doors" to cryptographic systems but I am of the opinion that such proposals carry extremely high risk to privacy and safety. It is not my intent to argue this question in this column.

What is of interest to me is a concept to which I was introduced at the Ditchley workshop, specifically, differential traceability. The ability to trace bad actors to bring them to justice seems to me an important goal in a civilized society. The tension with privacy protection leads to the idea that only under appropriate conditions can privacy be violated. By way of example, consider license plates on cars. They are usually arbitrary identifiers and special authority is needed to match them with the car owners (unless, of course, they are vanity plates like mine: "Cerfsup"). This is an example of differential traceability; the police department has the authority to demand ownership information from the Department of Motor Vehicles that issues the license plates. Ordinary citizens do not have this authority.

In the Internet environment there are a variety of identifiers associated with users (including corporate users). Domain names, IP addresses, email addresses, and public cryptography keys are examples among many others. Some of these identifiers are dynamic and thus ambiguous. For example, IP addresses are not always permanent and may change (for example, temporary IP addresses assigned at Wi-Fi hotspots) or may be ambiguous in the case of Network Address Translation. Information about the time of assignment and the party to whom an IP address was assigned may be needed to identify an individual user. There has been considerable debate and even a recent court case regarding requirements to register users in domain name WHOIS databases in the context of the adoption of GDPR. If we are to accomplish the simultaneous objectives of protecting privacy while apprehending those engaged in harmful or criminal behavior on the Internet, we must find some balance between conflicting but desirable outcomes.

This suggests to me that the notion of traceability under (internationally?) agreed circumstances (that is, differential traceability) might be a fruitful concept to explore. In most societies today, it is accepted that we must be identifiable to appropriate authorities under certain conditions (consider border crossings, traffic violation stops as examples). While there are conditions under which apparent anonymity is desirable and even justifiable (whistle-blowing, for example) absolute anonymity is actually quite difficult to achieve (another point made at the Ditchley workshop) and might not be absolutely desirable given the misbehaviors apparent anonymity invites. I expect this is a controversial conclusion and I look forward to subsequent discussion. C

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

ACM ON A MISSION TO SOLVE TOMORROW.

Dear Colleague,

Without computing professionals like you, the world might not know the modern operating system, digital cryptography, or smartphone technology to name an obvious few.

For over 70 years, ACM has helped computing professionals be their most creative, connect to peers, and see what's next, and inspired them to advance the profession and make a positive impact.

We believe in constantly redefining what computing can and should do.

ACM offers the resources, access and tools to invent the future. No one has a larger global network of professional peers. No one has more exclusive content. No one presents more forward-looking events. Or confers more prestigious awards. Or provides a more comprehensive learning center.

Here are just some of the ways ACM Membership will support your professional growth and keep you informed of emerging trends and technologies:

- Subscription to ACM's flagship publication ***Communications of the ACM***
- Online books, courses, and videos through the **ACM Learning Center**
- Discounts on registration fees to ACM Special Interest Group conferences
- Subscription savings on specialty magazines and research journals
- The opportunity to subscribe to the **ACM Digital Library**, the world's largest and most respected computing resource

Joining ACM means you dare to be the best computing professional you can be. It means you believe in advancing the computing profession as a force for good. And it means joining your peers in your commitment to solving tomorrow's challenges.

Sincerely,

Cherri M. Pancake
President
Association for Computing Machinery

Association for
Computing Machinery

Advancing Computing as a Science & Profession

DOI:10.1145/3235028

Encourage ACM to Address U.S. Election Integrity

In the spirit of Moshe Y. Vardi's call, in his "Vardi's Insights" column "Computer Professionals for Social Responsibility" (Jan. 2018), for ACM to "… be more active in addressing social responsibility issues raised by computing technology," we urge the ACM U.S. Public Policy Council to undertake a study of the technological infrastructure for U.S. elections. In a paper to be published in *the Proceedings of ETHICOMP 2018*,[3] we surveyed the widespread weaknesses in this infrastructure. We found, for historical and constitutional reasons, local control of elections, including equipment, processes, and procedures, is a prerogative jealously guarded. Practices and procedures even in neighboring counties can differ significantly, a factor in the presidential vote in Florida in 2000.

The bitterly contested aftermath of the related Florida recount led to federal legislation—the Help America Vote Act (HAVA) of 2002—concerning voting machines and registration procedures. Although intended to bring a measure of order and uniformity to the existing patchwork of state election systems, the legislation was hastily drafted and carelessly implemented, giving rise to problems that have plagued U.S. elections ever since.

Chronic problems with HAVA implementation have led to studies by the U.S. National Research Council and the U.S. Public Policy Committee of the ACM published in 2006[1] that had some effect in moving state and local officials toward adopting more reliable voting equipment and more secure processes for maintaining accurate voter-registration lists. Nevertheless, electronic voting machines and voter-registration lists remain vulnerable to attackers intent on interfering in U.S. elections. The profound shock administered by foreign actors trying to affect the result of the 2016 election is a call to action. ACM should once again mobilize the prestige and expertise of the computing profession to carry out a rigorous study to identify ways to restore confidence in the integrity of U.S. elections.

Some of the vulnerabilities we described in our paper, including politically motivated actions by state election officials and circulation of false information on social media, are not susceptible to easy solution. However, without jeopardizing the principle of local control, some problems in the U.S. election infrastructure can be eliminated or mitigated through sensible national standards and practices that represent the settled judgment of computing researchers and public-policy experts.

In the view of political scientist John Kingdon,[4] an issue can get on the political agenda only when three streams coincide: the problem; the solution; and the political will. It is clear that, in the U.S., we have a problem. The emerging consensus about standards for voting machines, computer databases of registered voters, electronic poll books, and risk-limiting audits constitutes a solution to several aspects of the problem. Kingdon underscores the critical role of policy entrepreneurs in building acceptance for solutions and creating couplings among the three critical streams. In today's polarized state of national discourse, the ACM U.S. Public Policy Council[2] is uniquely positioned to lend its trusted voice to the task of repairing civic confidence in this foundation of American democracy.

References

1. ACM. Statewide Databases of Registered Voters; https://www.verifiedvoting.org/wp-content/uploads/2013/04/acm-voter-registration-database-report-2006.pdf
2. ACM U.S. Public Policy Council; https://www.acm.org/public-policy/usacm
3. ETHICOMP conference. Sopot, Poland, Sept. 24–26, 2018; http://www.dmu.ac.uk/research/research-faculties-and-institutes/technology/centre-for-computing-and-social-responsibility/ethicomp/ethicomp-2018/ethicomp-2018.aspx
4. Kingdon, J.W. *Agendas, Alternatives, and Public Policies, Second Edition.* Longman Publishing Group, London, U.K., 2010.

William M. Fleischman and Kathleen V. Antaki, Villanova, PA, USA

USACM Responds:

We share the authors' concerns, as USACM has continued to engage in this area since its cited 2006 report through a variety of mechanisms, including congressional testimony, responses to formal government requests for information or comments, and letters to government bodies. USACM's most constrained resource is the time of its volunteers and limited staff. We thus try to avoid duplicating the efforts of others, including work being done by Verified Voting and the National Academies, both involving USACM members. There is, however, always more that can be done, and we would welcome the authors' contributions to USACM.

Stuart Shapiro, Chair, ACM U.S. Public Policy Council

Side with ACM Ethical Values

Bob Toxen's letter to the editor "Get ACM (and *Communications*) Out of Politics" (May 2018) said ACM was becoming too left leaning by taking decisions with more than a tinge of political motivation. In particular, Toxen said ACM should focus more squarely on technology. But politics is indeed inescapable when addressing policies that directly affect the field of computer science; for instance, immigration policy in the U.S., as well as every other country, has a direct effect on whether technology companies are able to attract and retain skilled workers, no matter where they might come from, in turn affecting the development of many technologies. Where would we be today if, say, Sergey Brin had been unable to emigrate to the U.S. from the Soviet Union in the 1970s or been separated from his parents at the border? Would Google even exist today if he had been forced to stay home? What would be the state of computing technology if Google had never existed? Likewise, passage in March 2016 of the Public Facilities Privacy & Security Act in North Carolina was a direct violation of the ACM Code of Ethics and Professional Conduct,[2] which obligates ACM and its membership to be fair to all and not discriminate. How could ACM in good conscience host a conference in a jurisdiction that had discriminated against some of its own membership? It would be just and within the ACM mandate to change conference venues to respect those values.

ACM must also recognize that systems can be deployed for harmful, as well as for good, purposes. A central pillar of the ACM Code[1] is to avoid harm to others, requiring ACM and its membership to take moral and ethical decisions on the use of technology that might seem to many otherwise reasonable professionals as political. Consider Google's work with the U.S. Department of Defense to develop AI systems that could enable drones to more effectively identify targets on the ground. Many Google employees have objected to the program due in part to the potential harm it might cause innocent civilians. Following this outcry from its own employees, as well as from the broader community, Google decided to not renew the program.[2]

ACM could, as Toxen suggested, remain narrowly focused on technology, leaving moral and ethical discussion to the political arena or engage in ways that might force it to take sides in the political arena. In his 1986 Nobel Peace Prize acceptance speech, Holocaust witness Elie Wiesel said, "We must always take sides. Neutrality helps the oppressor, never the victim. Silence encourages the tormentor, never the tormented." In light of recent political and social events and advances in technology, particularly AI and, potentially, autonomous systems, today might be the right time to build a community, perhaps even a special interest group, dedicated to issues of ethics and public policy.

References

1. ACM Code of Ethics and Professional Conduct. Association for Computing Machinery; https://ethics.acm.org/
2. Wakabayashi, D. and Shane, S. Google will not renew Pentagon contract that upset employees. *New York Times* (June 1, 2018).

James Simpson, Chatham, ON, Canada

Communications welcomes your opinion. To submit a Letter to the Editor, please limit yourself to 500 words or less, and send to letters@cacm.acm.org.

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:10.1145/3231166 http://cacm.acm.org/blogs/blog-cacm

Assessing Responsibility for Program Output

We lack an easy way to indicate that algorithms do not make decisions and are not biased; programmers do, and are.

Robin K. Hill
Articulation of Decision Responsibility
http://bit.ly/2kDNgzY
May 21, 2018

Remember the days when record-keeping trouble, such as an enormous and clearly erroneous bill for property taxes, was attributed to "computer error?" Our technological society fumbles the assignment of responsibility for program output. It can be seen easily in exaggerations like this, from a tech news digest: "Google's Artificial Intelligence (AI) has learned how to navigate like a human being." Oh, my. See the *Nature* article by the Google researchers[2] for the accurate, cautious, description and assessment. The quote given cites an article in *Fast Company,* which states that "AI has spontaneously learned how to navigate to different places."[4] Oh, dear.

But this is not the root of the problem. In the mass media, even on National Public Radio, I hear leads for stories about "machines that make biased decisions." Exaggeration has been overtaken by simple inaccuracy. We professionals in Tech often let this pass, apparently on the belief the public really understands machines and algorithms have no such capacity as is normally connoted by the term "decision"; we think the speakers are uttering our own trade shorthand. When we say "the COMPAS system decides that offender B is more likely to commit another crime than is offender D"[1] (paraphrase mine), it is short for "the factors selected, quantified, and prioritized in advance by the staff of the software company Northpointe assign a higher numeric risk to offender B than to offender D." When the *Motley Fool* website[6] says "computers have been responsible for a handful of 'flash crashes' in the stock market since 2010," it means that "reliance on programs that instantaneously implement someone's predetermined thresholds for stock sale and purchase has been responsible ... etc."

The trouble is that there is no handy way to say these things. The paraphrases here expose the human judgments that control the algorithms, but the paraphrases are unwieldy. For decades of software engineering, we have adopted slang that attributes volition and affect to programs. Observations can be found on Eric S. Raymond's page on anthropomorphization[5]. I doubt many hackers ascribe the intentional stance to programs; I suspect rather that programmers use these locutions for expedience, as the "convenient fictions that permit 'business as usual'."[3] But the public misunderstanding is literal, and serious.

Algorithms are not biased, because a program does not make decisions. The program implements decisions made elsewhere. Programs are made up of assignments of value, evaluations of expressions, and branching to addresses for loading of instructions. There is no point of unpredictable choice, that is, a choice not determined by the code (even for "random" number generation), if we rule out quantum computation, which I am not qualified to consider. Certain scenarios may appear to challenge this bald determinism. Let's scrutinize those briefly.

Deductive closure includes propositions not immediately obvious.

> But even where the programmers are not sure what exactly will happen, because of obscure compound conditions, the algorithm does not "make a decision." What happens is an implication of the assertions in force (written into the code if the programmer bothered to formulate assertions), that is, an implication of the deductive closure. The question whether programmers can be held responsible for the distant eventualities is significant, noting that what we view as algorithmic bias does not often seem deliberate. In any case, the deciding agent is certainly not the machine.

Timing of interactions may result in unanticipated outcomes, as in passive investment through computerized stock trading.

> But unexpected states do not demonstrate demonic agency. Someone has decided in advance that it makes sense to sell a stock when it loses n% of its value. That's not what we would call a real-time decision on the spot, because it ignores (1) the real time and (2) the spot. We would correctly call that a decision made earlier and elsewhere by system designers, which played out into unforeseen results.

The pattern-matching of deep learning precludes the identification of symbolic variables and conditions.

> With no semantics available, no agent prominent, and no execution through a conditional structure traceable, the computer looks like the proximate decider. But no. If there are training cases, some complex combination of numeric variables has developed from given initial values which were adjusted over time to match a set of inputs with a set of outputs, where those matches were selected by the systems designers. In unsupervised learning, some sort of regularities are uncovered, regularities that were already there in the data. Although it may be tempting to say that no one is deciding anything, certainly no computer is making anything that could be called a decision. Someone has planned antecedently to seek those regularities.

Selection, recommender, and classification systems use the criteria implemented in their decision structure. We in the trade all know that whatever the algorithmic technique, the computer is not deciding. To explain to the public that computers are dumb may baffle and frustrate, rather than educate. The malapropisms that grant agency to algorithms confuse the determination of responsibility and liability, but also the public grasp of Tech overall. People may attempt to "persuade" the computer, or to try to fix, enhance, or "tame" the programs, rather than just rejecting their inappropriate deployment. At the extreme, people feel helpless and fearful when danger comes from beings like us—willful, arbitrary, capricious—except more powerful. Worse yet would be apathy: Society may ignore the difficulties and become resigned to the results, as if such programmed assessments were factive.

What would be the correct locution, the correct way to say it, passive toward machine and active toward programmer (or designer or developer or specification writer or whomever)? How should we note that "the deductive closure of home mortgage qualification criteria entails red-lining of certain neighborhoods"—other than to say those exact words, which are not compelling? How should we say that "The repeated adjustment of weighting criteria applied to a multi-dimensional function of anonymous variables, closely approximating an unknown function for which some correct outcomes have been identified by past users, associates this individual record to your own discrete declared criteria for a date"—without saying "the dating app has chosen this match for you"?

We have no other way of expressing such outcomes easily. We lack the verbs for computing that denote reaching states that look like decisions, and taking actions that look like choices. We need a substitute for "decides" in "the algorithm decides that X," something to fill in the blank in "the program ________ X." Perhaps "the program fulfills X." Perhaps "the program derives that X." Well ... this seems lame. The trouble really is that we have to avoid any verb that implies active mental function. This is new. This is unique to computing, as far as I can tell. The Industrial Revolution brought us many machines that seemed to have human capacities, but they also had material descriptions. For mechanical devices, verbs are available that describe physical functionality without the implication of cognition: "The wheel wobbles." "The fuel line clogged." We may say, jokingly or naively, that "the car chooses not to start today," but we are not forced into it by lack of vocabulary.

For this new technological requirement, the best locution I can come up with is, "the result of the programmed assumptions is that X." I have not heard anyone seriously appeal to "computer error" as a final explanation for some time; that seems like progress in understanding Tech. If we can forgo that locution, maybe we can forgo "biased algorithms."

Any other ideas?

References

1. Angwin, J., et al. Machine Bias. *ProPublica*. May 23, 2016, http://bit.ly/2sGcEbH.
2. Banino, A., et al. 2018. *Nature* 557; pages 429–433. Vector-based navigation using grid-like representations in artificial agents. doi:10.1038/s41586-018-0102-6.
3. Caporael, L.R. 1986. Anthropomorphism and Mechanomorphism: Two Faces of the Human Machine. *Computers in Human Behavior 2*; pages 215-234. https://doi.org/10.1016/0747-5632(86)90004-X.
4. Grothaus, M. Google's AI is learning to navigate like humans, *Fast Company*, May 15, 2018, http://bit.ly/2JuyYzI.
5. Raymond, E. Anthropomorphization. From *The Jargon File*, http://bit.ly/2kPmF2Z.
6. Williams, S. (TMFUltraLong). The Evolution of Stock Market Volatility, *The Motley Fool*, Apr 3, 2018, http://bit.ly/2xKrROY.

Robin K. Hill is adjunct professor in the Department of Philosophy, and in the Wyoming Institute for Humanities Research, of the University of Wyoming. She has been a member of ACM since 1978.

Science | DOI:10.1145/3231168

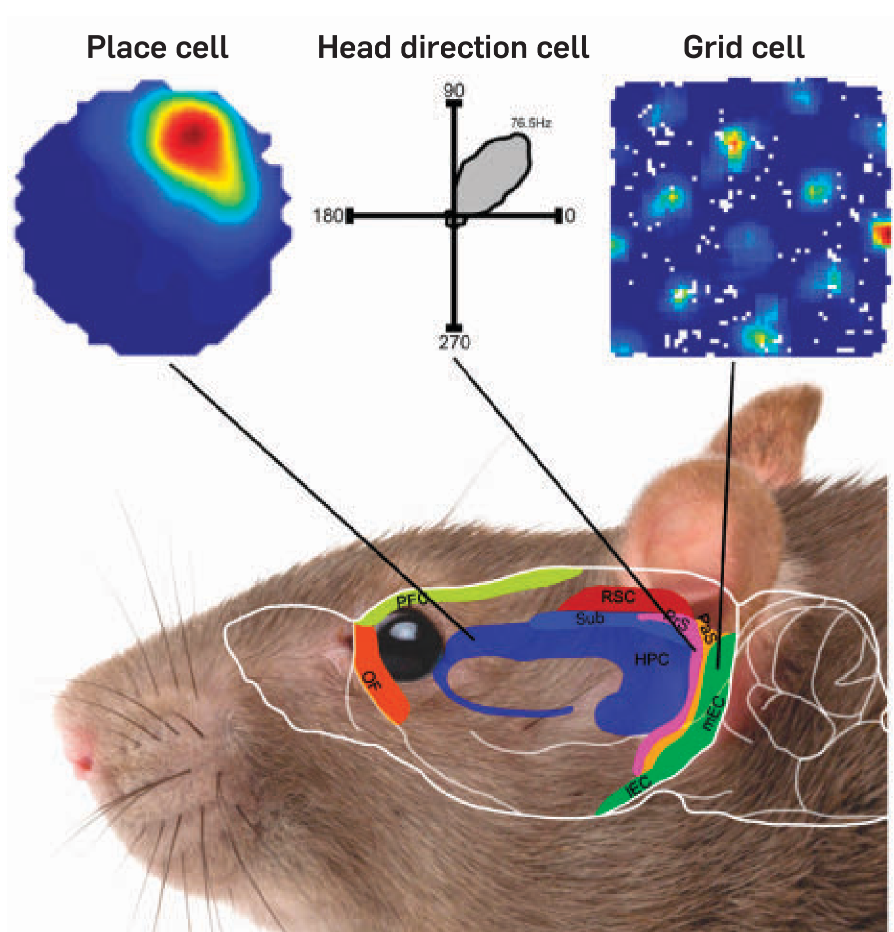
Chris Edwards

Animals Teach Robots to Find Their Way

Navigation research demonstrates bio-machine symbiosis.

A DEMONSTRATION VIDEO that veteran University College, London neuroscientist John O'Keefe often presents in lectures shows a rat moving around the inside of a box. Every time the rat heads for the top-left corner, loud pops play through a speaker; those sounds are the result of the firing of a specific neuron attached to an electrode. The neuron only fires when the rat moves to the same small area of the box. This connection of certain neurons to locations led O'Keefe and student Jonathon Dostrovsky to name those neurons "place cells" when they encountered the phenomenon in the early 1970s.

Today, researchers such as Huajin Tang, director of the Neuromorphic Computing Research Center at Sichuan University, China, are using maps of computer memory to demonstrate how simulated neurons fire in much the same way inside one of their wheeled robots. As it moves around a simple cruciform maze, the machine associates places with pictures of milk cartons, cheese, and apples that it encounters. When asked to find those objects, the same neurons fire. Although the robot looks in the direction of each object when



Example cells and a graphic representation of their anatomical distribution in the rat brain. At top left, the firing rate heat map of a place cell recorded as a rat explored a circular arena. Top center, a head direction cell firing rate plot. Top right, firing rate map of a grid cell.

FIGURE: THE REPRESENTATION OF SPACE IN THE BRAIN. (2016). GRIEVES, RODDY & JEFFERY, KATHRYN. BEHAVIOURAL PROCESSES. 10.1016/J.BEPROC.2016.12.012.

it moves to the center of the maze as part of its hunt, Tang says analysis of the simulated neuron shows "the movement is driven by this stored information, rather than visual recognition of the shape."

Researchers see synthetic models inside robots as crucial for guiding biological research, as well as for the design of more capable machines. Physical experiments can only measure the activity of a few neurons at a time, which makes it difficult to build a broad overview of how an animal thinks about a problem. Computer models make it possible to test hypotheses about the brain's behavior by seeing how similar a robot's reaction to a problem is to that of the animal. Neuron-level tests in the creature can then confirm or contradict the computer model.

Barbara Webb, a professor of biorobotics at the University of Edinburgh who has been investigating the navigational abilities of insects, favors building computer models even where biological data is limited. More than a decade ago, her team developed a computer model for path integration, a technique used by ants and bees among others to memorize a route. The idea had little anatomical basis at the time, but seemed to be a viable behavioral model. Recent experiments have confirmed similar activity taking place in collections of insect neurons.

Although insects have simple navigational structures, mammalian research has underpinned the key models used in robot development. Analogs of neural networks found in the rat's brain underpin what is today the most widespread model for biologically inspired navigation.

Michael Milford and colleagues at the Queensland University of Technology in Australia developed the RatSLAM architecture almost 15 years ago. Released in open source form, the relative accessibility of the techniques it uses has helped promulgate RatSLAM. Numerous experiments by Milford and other groups, such as the one based at Sichuan University, have demonstrated the ability of the system to work in many scenarios, up to the level of city streets. However, in such large-scale environments, it has to compete with more conventional GPS-enabled navigation systems.

Mammalian research has underpinned the key models used in robot development. Analogs of neural networks found in the rat's brain underpin the most widespread model for biologically inspired navigation.

Says Milford, "Where our work remains competitive is in areas where we don't have a lot of computing power, or in situations such as an underground mining site; places where you don't have access to satellites for GPS or access to the cloud. We have also regularly had conversations with manufacturers of products such as robot vacuum cleaners, or people who deploy autonomous robots in sites where you have limited sensing."

What the rat's brain brings to this research is the ability to navigate without external aids, and in dark places where the animal loses the ability to rely on visual cues. The rat seems to use information from its own movement, coupled with memories of past journeys, to work out how to get from one place to another.

The question for researchers is how close to an actual rat brain do robots need to be, to be as effective at navigation. Robots have the advantage of being able to sense their own motion far more accurately than an animal, and to take advantage of a wide range of accurate motion sensors, whereas a rat may make less-reliable estimates of how far and in which direction its legs have moved it.

The models that researchers build run the gamut from relatively simple structures to intensively detailed models. Milford's group opted for simplicity. "To model a single neuron to the

ACM Member News

SEEING THE BEAUTY IN COMPUTERS

"I was always afraid of chemistry because you could blow things up," says Gail Murphy, a professor in the Department of Computer Science, and vice president of Research & Innovation, at the University of British Columbia in Vancouver, Canada.

On the other hand, Murphy says, the beauty of computers is that you can do all types of experimentation, and if you don't get it right at first, you can just keep trying.

Murphy obtained her undergraduate degree in computing science from the University of Alberta, Canada, then spent five years working as a software developer before returning to graduate school and earning her master's degree and Ph.D. in computer science and engineering from the University of Washington.

Her research interests are in software engineering, with a focus on improving the productivity of knowledge workers and software developers. Murphy's goal is to make it easier for developers to find the information they need from large information spaces in order to do their work.

Murphy also is co-founder and chief science officer at Vancouver-based Tasktop Technologies, a company she helped launch in 2007 that helps connect software delivery organizations with automated deployment and integration technologies.

Currently, Murphy is working to identify and extract snippets of design information from software artifacts and developers' discussions, and making this information useful for developers as projects progress. She points out that this is especially important in open source development, because developers do not tend to write down the design, but it is implicit in all of the artifacts with which they interact.

—John Delaney

detail that we know takes incredible amounts of computational power. We didn't want to do that, as we wanted to create something useful in the short term. As we became very familiar with the navigation problem and mapping problem, we couldn't find a compelling reason to go to a higher level of fidelity," Milford says.

Milford and colleagues developed what they call "pose cells," which shared some characteristics with the place cells found by O'Keefe decades earlier, but which added information on the direction in which the robot faced, and the distance of travel recorded by internal sensors. Such pose cells can represent multiple physical locations; the robot determines the difference by adding information from cells that record the visual scene at each location.

The pose cells turned out to share characteristics with a class of neurons called "grid cells" discovered several years later by neuroscientists Edvard and May-Britt Moser, then working at the Norwegian University of Science and Technology. The Mosers shared the 2014 Nobel Prize in Physiology or Medicine with O'Keefe for their study of the multiple types of navigational cells of mammals.

"Grid cells display strikingly regular firing responses to the animal's locations in 2D (two-dimensional) space. Existing studies suggest place-cell responses may be generated from a subset of grid-cell inputs," says Tang, pointing to projects conducted by his team in which simulations of place and grid cells helped improve robot navigation. Grid cells appear to become more important as the area covered by the machine increases.

A key facet of grid cell behavior for large-scale navigation is its ability to store information about multiple locations. "The assumption is that this is a very clever way to map data into a very compact storage representation. The data so far suggest you can do immense amounts of data compression," Milford says, pointing to work his group is doing for the U.S. Air Force in this subject area.

As well as the functions of individual types of neurons, a common link between robot design and biology lies in the way they are structured. Rat-

In addition to the functions of individual types of neurons, a common link between robot design and biology lies in the way they are structured.

SLAM is one of a number of systems that use the competition between groups of simulated neurons to move activity to the most appropriate location. In these attractor networks, neurons excite those close to them and inhibit those further away. However, sometimes new sensor information causes activity to rise elsewhere until that group of neurons takes over and, in turn, inhibits its competitors.

Clusters of neurons that seem to operate as attractor networks have now been found in the navigation centers of insects that help with path integration and steering. Insects lack the rich collections of cells that mammals use for navigation, but Lund University biology researcher Stanley Heinze is impressed by the way insects can recall complex routes that are sometimes miles long, making it possible to find their way home easily. Working with Webb's team from the University of Edinburgh and colleagues at Lund in Sweden, Heinze developed a robot to test ideas of how honeybees navigate.

Webb says ants, bees, and other insects appear to use a combination of path integration and visual memory to store routes. She points out that if you move an ant away from one of the routes it has memorized and drop it in a new location, it will adopt a search pattern; as soon as it encounters a point on one of its known paths, it will orient itself and find its way home.

In the cluttered environments through which they fly, bees appear to rely more on direction and speed than the local landmarks that guide ants. The species chosen for study by Heinze and Webb has receptors in its eyes that respond to polarized light, and tend to forage at times when this polarization is most apparent. Tests with grid patterns demonstrated how bees can use these cells to sense speed accurately even when a strong wind forces them to one side.

Heinze and colleagues built versions of the path-integration and speed-sensor cells into a ring-shaped attractor network to reduce noisy inputs from multiple sources into a single packet of activity that could shift around the ring. Sent out on random routes, the network helped the machine find its way back to the starting point, demonstrating the viability of the concept.

Through such simple models, researchers hope to continue the long journey towards understanding how intelligence works and how it can be emulated in computers and robots. Milford says, "I always regard spatial intelligence as a gateway to understanding higher-level intelligence. It's the mechanism by which we can build on our understanding of how the brain works." C

Further Reading

Milford, M., Jacobsen, A., Chen, Z., and Wyeth, G.
RatSLAM: Using Models of the Rodent Hippocampus for Robot Navigation and Beyond. *Robotics Research: The 16th International Symposium* (2013).

Galluppi, F., Davies, S., Furber, S., Stewart, T., and Eliasmith, C.
Real Time On-Chip Implementation of Dynamical Systems with Spiking Neurons. IEEE World Congress on Computational Intelligence (WCCI) 2012, Brisbane, Australia.

Hu, J., Tang, H., Tan, K.C., and Li, H.
How the Brain Formulates Memory: A Spatial-Temporal Model. *IEEE Computational Intelligence*, (2016) Volume 11, Issue 2.

Stone, T., Webb, B., Adden, A., Weddig, N.B., Honkanen, A., Templin, R., Wcislo, W., Scimeca, L., Warrant, E., and Heinze, S.
An Anatomically Constrained Model for Path Integration in the Bee Brain. *Current Biology* (2017), Volume 27, Issue 20.

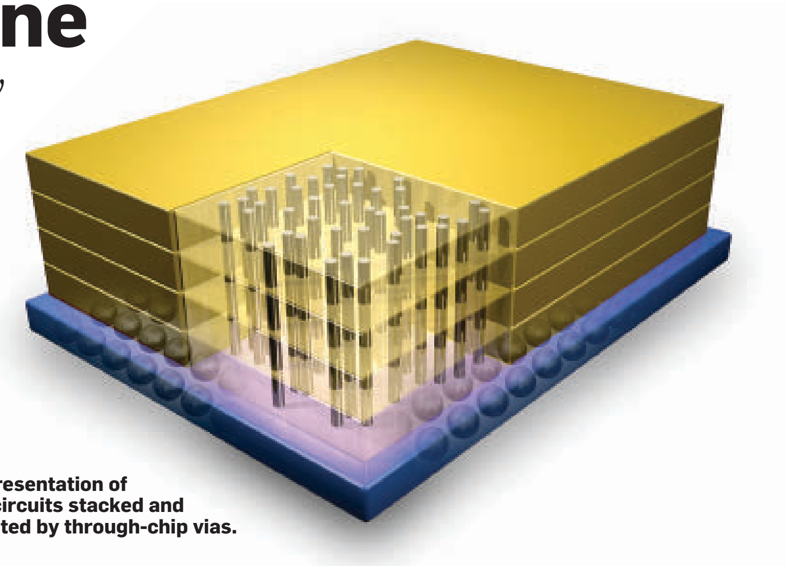
Chris Edwards is a Surrey, U.K.-based writer who reports on electronics, IT, and synthetic biology.

Technology | DOI:10.1145/3231207

Don Monroe

Electronics Are Leaving the Plane

Stacking chips and connecting them vertically increases both speed and functionality.



Artist's representation of integrated circuits stacked and interconnected by through-chip vias.

FOR DECADES, INTEGRATED circuits have been confined to a veneer on semiconductor chips, with transistors and wiring devices packed ever more densely within this thin sheet. As in-plane shrinkage has become more challenging, however, electronics companies are looking to stack multiple circuit layers vertically to boost speed and functionality, while reducing power consumption and size.

"The performance of a system is not controlled by the individual components, but by the way that you can assemble these different components," said Paolo Gargini, head of the International Roadmap for Devices and Systems, an IEEE Standards Association Industry Connections program that has supplanted the more device-focused semiconductor roadmap. Over time, stacking will give way to true monolithic growth of three-dimensional (3D) chips for some applications, like memory.

Historically, chips were electrically connected with long, wide metal traces on a printed circuit board, which take a lot of energy and time to charge and discharge. Engineers have long known that stacking chips and connecting them vertically improves both power and speed by reducing the electrical path between them. Memory technology has led the way in exploiting this trick, but the potential benefits affect everything from power-sensitive mobile devices to power-hungry processors in online data centers.

For high-performance computing, "you can save 60%, 90% of the power required, because a lot of it is in communication from a processor and getting access to the memory and doing the compute locally," said John Knickerbocker of IBM in Yorktown Heights, NY.

Stacking also compactly connects chips made using incompatible processes. At the International Electron Devices Meeting in December, for example, Sony reported sandwiching a logic layer, a DRAM layer, and a CMOS imaging layer in a stack that was only 130 microns thick.

A further advantage of combining separate chips is that sensors "include an analog circuit that prefers a higher voltage in many cases. Logic circuits prefer a lower voltage for power consumption and speed," said Fumiaki Yamada, who worked on the Japanese 3D "Dream Chip" project exploring potential technology for 3D chips, and is now an independent consultant.

Stacking could also be the ultimate way to pack diverse functions into small devices like smart watches, or to drive the nascent "Internet of Things" (IoT). So far, however, many mobile devices still use a more mature technology called package-on-package, which stacks the chips only after they have been packaged. The packages can still be stacked vertically, and they are equipped with an array of solder balls to make many contacts to a common substrate, but the modularity allows manufacturers to design them independently and test them before assembly.

Challenges

True 3D stacking exploits wafer-processing-style tools in service of packaging. One key element is deep etching of uniform vertical channels into a wafer—or even all the way through it. These channels are then filled with metal to form "through-silicon vias" (TSVs) that connect the top and bottom of a chip, "which helps the performance because you don't have to convey the data to the edge of the chip," said Yamada. These methods resemble the long-established "flip-chip" face-to-face bonding, but they can extend to many chips.

Another key capability is thinning of wafers to less than the thickness of a human hair, which is useful both for compact stacking and for facilitating drilling holes through them. These sheets, which may be as wide as an entire wafer, then need to be precisely aligned with and bonded to other processed circuits. Yamada notes that reliably handling these thin layers, often after temporarily gluing them to another substrate for handling, is in some ways "still a problem to be solved," although process engineers have made significant progress.

Stacking also faces other challenges that make it expensive and have so far limited its use. For one thing, the modified structure requires significant changes in design. The array of contacts throughout the active circuitry takes up significant real estate in the middle of the chip that could otherwise be used for transistors. In ad-

IMAGE COURTESY OF THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, U.S. DEPARTMENT OF COMMERCE dition, the different chips in the stack need to be designed with matching pin layouts, which requires a high degree of coordination in the design of the various chips, and limits the manufacturer's flexibility to modify the layout or use alternative suppliers.

Another major issue is heat removal, which is already a major issue for traditional chips. Combining multiple layers of heat-generating devices, and burying them further from the surface, makes the problem worse. Still, Knickerbocker said, "for low-power applications like some mobile applications and some IoT applications, the power levels are so low that getting the power in and cooling it is not a problem at all," especially since stacking reduces the total power substantially. For high-performance computing, "even though the power delivery and the cooling challenges go up substantially, there's still tremendous benefit at the system-performance level to make it worthwhile," Knickerbocker said, adding that advanced power delivery may be needed, as well as cooling technologies such as flowing liquids through the chip stack or using materials that absorb heat by undergoing a phase transition.

Multiple chips also complicate manufacturing yield, which is critical to the economics of electronics manufacturing. When individual components can be proven functional before assembly, the yield of a multi-chip device can be better than that of a single chip combining the same components. However, without assurance of such "known-good die," a failure of any layer will require trashing the whole stack.

Memory First

So far, the greatest benefit of 3D chips has been on memory. One reason is that memory consists of identical repeated units, and designers have long taken advantage of the interchangeability of memory blocks to bypass occasional defective ones. (Field-programmable gate arrays have a similar redundancy.) In addition, although heating is a major challenge for stacking of logic chips, in memory chips many transistors are inactive much of the time.

Equally important is the seemingly insatiable demand for memory in all sorts of systems. Indeed, two distinct flavors of vertically stacked DRAM have become important in the last few years. High-bandwidth memory (HBM) has an aggressive champion in AMD, and is already in its second generation, HBM2. A competing technology, Hybrid Memory Cube (HMC), has been developed by Micron. Although there are important differences, both feature very high data rates with over 1,000 or more connections between layers. "For the HBM stacks, the density of the interconnect is now down at like 55μm pitch between connections," Knickerbocker said.

In this rapidly evolving field, stacking is not the only route manufacturers are exploring for 3D memory, however. Samsung, for example, in addition to its HBM products, has developed a monolithic flash-memory technology called V-NAND, which features strings of dozens of floating gate transistors connected vertically in series along deep etched trenches refilled with silicon, grown over a wafer of control and sensing circuitry.

Micron also has teamed with Intel to develop their own monolithic multilayer flash memory. Although they announced an end to this collaboration in January, the two companies are still collaborating on different monolithic technology, a multilayer resistive memory called 3D-XPoint.

Arms Race

In a more general stacking configuration, combining different types of chip remains challenging. "You've got to have the design, you've got to have the assembly, you've got to have either the same die size or thin chips are hanging out. It's not so easy," Knickerbocker cautions. As an intermediate step, "a lot of people over the past five or years have been using what's called 2.5D, like a silicon interposer, and put multiples of these chips side by side, or some combination of chip stacks and chips next to them," he said. "You can get lots and lots of connections for adjacent chips in a way that allows that product to be rolled out very quickly without doing the design consistency across many different technologies that go into a full chip stack."

For example, Nvidia's latest devices for artificial intelligence applications combine a high-density interconnect on a silicon interposer wafer with HBM memory stacks close to their graphics processor unit (GPU). "That's a good start," Knickerbocker said, "but I still think 3D and full chip stacking for many applications will give the best and highest performance at the system level."

On the other hand, stacking will always be competing with the approach of growing new devices on a wafer during fabrication, but it is hard to develop a monolithic process that does not disrupt the layers below. "Packaging is a shortcut," said Gargini, who oversaw many generations of this arms race during his decades in technology development at Intel, including the first integration of modest cache memories onto the same die with a processor. "The packaging side buys you performance a couple of generations ahead of technology, then the monolithic part catches up," Gargini said. "At each point in time, you take the best trade-off between cost and performance."

In the end, customers care more about the price, performance, and size of the entire packaged device than about how many components are in it or how it is assembled inside. As long as advanced packaging, whether by stacking or other means, provides an advantage, "these companies are absolutely ready to do this stuff," Gargini said. "They have had these capabilities for a long time." C

Further Reading

Lapedus, M.
A New Memory Contender?, Semiconductor Engineering, January 2, 2018, http://bit.ly/2DbliT2

Kondo, K., Kada, M., and Takahashi, K. (Eds.)
Three-Dimensional Integration of Semiconductors, Springer International Publishing, 2015, http://bit.ly/2G6uXxd

3D Stacked Memory: Patent Landscape Analysis Lexinnova Technologies LLC, http://bit.ly/2tuC0Ny

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

Society | DOI:10.1145/3231170

Esther Shein

Broadening the Path for Women in STEM

Organizations work to address 'a notable absence of women in the field.'

In 2018, girls and women are getting the message they belong in computer science as much as boys and men, thanks to a greater push for STEM (science, technology, engineering, and mathematics) curricula in schools and a vast number of programs available to them outside of school.

Yet the numbers remain discouraging. Although computer science jobs are projected to grow 15% to 20% through 2020, the majority of these positions will be pursued and filled by men, according to Women in Computer Science (WiCS).

In 2016, 26% of professional computing jobs in the U.S. workforce were held by women; 20% of the Fortune 100 chief information officer (CIO) positions were held by women, and 23% of Advanced Placement (AP) computer science test takers were female, based on data from the National Center for Women and Information Technology (NCWIT).

"As STEM-related industries on a whole add over 1.7 million jobs in the coming years, there continues to be a notable absence of women in the field,'' according to the WiCS website.

All that has not discouraged Tahsina Saosun, 20, a computer science major at Barnard College and events coordinator for the Barnard/Columbia University chapter of WiCS (CUWiCS).

Saosun became interested in studying computer science after participating in the program Girls Who Code the summer before her senior year of high school. After the eight-week session in which she was introduced to various programming languages, and learned how to declare variables and write code in loops, she was hooked.

Saosun's experience "has been kind of mixed.'' She says she found support in introductory computer science courses, but not as much in upper-level classes. Most of her professors have been male.

"Overall, I haven't felt uncomfortable,'' she says, "but I would give credit to my involvement in CUWiCS." It also helps to be studying at Barnard, a women's college, she adds. "There's lots of support and resume advice and career advice. That helps a lot."

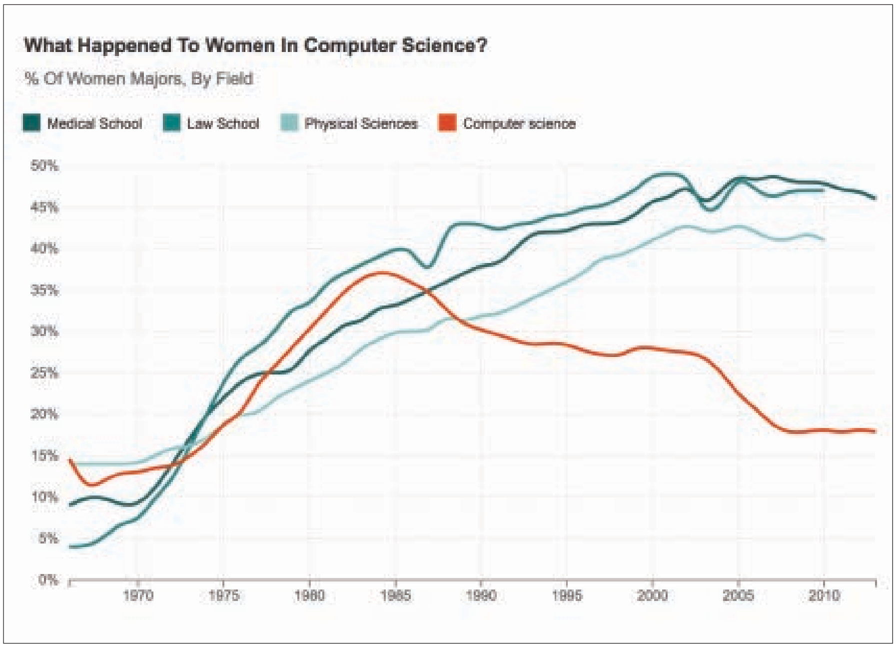
That is something Wendy DuBow is working to replicate for others. DuBow, senior research scientist and director of evaluation at NCWIT, says the organization focuses on generating awareness of computer science to girls in grades K–12, as well as in secondary education and industry.

While some research indicates girls should be exposed to computer science in middle school in order to best pique their interest, other research says "the best thing that could happen is that rigorous computer science be offered in high school so all students are exposed to it … the way they're exposed to English, math, and science,'' says DuBow. "Exposure is a huge influencer and predictor of who will go on to major or minor in [computer science] in college. So, we work on all fronts."

DuBow believes computer science should be a graduation requirement, but points out there are still high schools that do not offer a single course in the discipline. Even when it is offered, she says, "only certain students will take it, so it doesn't do anything to broaden participation in computing." Students will get steered away from computer science unless they show a predilection or fit a stereotype, she says.

"So if you haven't had exposure [to computer science] and people don't see you as someone who does computing from an early age, you don't see yourself that way, either.''

NCWIT offers a program to edu-



Source: National Science Foundation, American Bar Association, American Association of Medical Colleges.

cate high school guidance counselors about computer science, and hopes to it expand to community colleges.

A female student might take a computer science class in college, but "sometimes there's still a 'weeding-out mentality' going on in introduction to computer science classes," DuBow says. "That's not a welcoming environment, especially if you perceive people around you have had more exposure, and if it's not an inclusive classroom, it's going to be a turnoff."

DuBow says it is important not to think of a computer science major as the indicator of success, since there are interdisciplinary majors from which students emerge with a deep understanding of computer science, such as bioinformatics, biomedical engineering, computational media, game design, and multimedia computing.

The overarching issue remains young women's lack of exposure to computer science, DuBow says.

"As a society, we still have these stereotypes about who ought to be in what kind of field, so there's still really strong biases against women going into computer science, and that gets inculcated in kids at an early age and instigated by parents, and also counselors.'' She adds that there are also "lots of advisors that will steer girls and people of color one way and white boys another way."

United Nations Secretary-General Antonio Guterres has weighed in on the issue. During a February speech for the International Day of Women and Girls in Science, Guterres said, "Although both girls and boys have the potential to pursue their ambitions in science and mathematics, in school and at work, systematic discrimination means that women occupy less than 30% of research and development jobs worldwide."

Guterres said it is important to the world that girls and women be encouraged to achieve their full potential as scientific researchers and innovators. He called for "concerted, concrete efforts" to overcome stereotypes and biases .

Eve Riskin has been an electrical engineer long enough to remember when biases against women in STEM were more obvious. Now associate dean for Diversity and Access in the College of Engineering at the University of Washington (UW), Riskin earned her bachelor's degree from the Massachusetts Institute of Technology (MIT), a school she says her mother picked for her. Her influences were strong; both of Riskin's parents were programmers, and her siblings also worked in computer science.

Riskin decided she wanted to an electrical engineering professor during graduate school at Stanford University, but when she finished graduate school, "there will still very few women faculty, and I was number four in the electrical engineering department [at UW], which was a huge number in 1990."

In 2001, the U.S. National Science Foundation launched the ADVANCE program to increase the participation and advancement of women in academic STEM careers. UW received an award to aid those efforts, says Riskin, who is also faculty director of UW ADVANCE.

The program provides professional development for women and junior faculty. This is important, Riskin says, "because if you're a professor, you live in the department and if your chair is thoughtful and doesn't have biases, your life will be better—as opposed to one who gives smaller salaries and lousy teaching assignments."

Riskin says when she was studying, it was not uncommon to hear comments like "Why should I have you in my class when you're just going to get married and have babies?" There are still stragglers from the old days on faculties, she adds.

Yet progress is being made. In 2016, 27% of the graduates in UW's College of Engineering were women; today, Riskin says, there are eight or nine female faculty in her department.

U.N. Secretary-General Antonio Guterres has called for "concerted, concrete efforts" to overcome stereotypes and biases that dissuade women pursuing careers in STEM.

Riskin says there is more to be done to help women advance in computer science, because institutional cultures are ingrained and hard to change, especially when it comes to hiring practices.

"The way people are screened and hired is a problem," Riskin says. When people's credentials are questioned, they begin to feel they do not belong, and that perpetuates. "Venture capital firms generally don't fund women,'' Riskin says. "We all have biases and affinity toward people like ourselves."

People need to be aware of those biases and not have knee-jerk reactions, she says. "We have to be more thoughtful in how we interview candidates."

Of course there are exceptions, and women who have risen to management roles in STEM fields, like Angie Duong, who became the first female engineer at Irvine, CA-based transmission control protocol solutions provider Badu Networks. Today, Duong is software development manager at Badu, where she leads a team of 12 male engineers.

Despite her achievements, Duong still sees biases in the workplace to be overcome. "When you work with all men, you have to know what you're doing, you have to establish your reputation, you need to step up and make the decisions,'' she says. "Here [at Badu Networks] no one looks down at women, but just because they don't say it doesn't mean they don't think it."

In the meantime, efforts are on-going to help young women become interested in computer science as a career, and to make computer science welcoming to women rather than exclusionary.

In 2017, the Girl Scouts announced its first-ever cybersecurity badge for girls in grades K–12.

This year, 16 states and one U.S. territory partnered with the SANS Institute, a computer security training and certification organization, on the first "Girls Go Cyberstart," a national competition to attract young women to cybersecurity.

"There are big barriers to women getting into this field, and we want to give them an on-ramp that is their own," says Alan Paller, director of research at

Bethesda, MD-based SANS Institute.

Paller was pleased that 6,647 girls from 1,000 U.S. schools participated in the competition. Participants performed tasks including cracking codes, plugging security gaps, and creating software tools.

Creating greater gender parity in STEM-oriented professions will take more than improving science education for girls and promoting overall gender equality, according to the 2018 report "The Gender-Equality Paradox in Science, Technology, Engineering and Mathematics Education." The study by the journal *Psychological Science* looked at almost 500,000 adolescents from 67 countries in the Program for International Student Assessment (PISA), the world's largest educational survey. It found that girls were at least as strong in science and math as boys in 60% of the PISA countries, and that they were capable of college-level STEM studies.

Yet the gender gap in STEM fields persists.

"The generally overlooked issue of intraindividual differences in academic competencies and the accompanying influence on one's expectancies of the value of pursuing one type of career versus another need to be incorporated into approaches for encouraging more women to enter the STEM pipeline," the study notes. "In particular, high-achieving girls whose personal academic strength is science or mathematics might be especially responsive to STEM-related interventions."

Whether a girl has the desire to be involved in computer science, encouragement and exposure to the field remain focal points. In a time when the #MeToo movement has gained momentum, the push to empower young women to feel welcome in computer science continues as well.

These efforts have gone global. In January, for example, a female coputer engineer and some colleagues created Jiggen Tech Hub, West Africa's first tech hub for women.

"It's not about individual women changing their perspectives or doing something different," says Dubow. "It's about departments and school systems and industry and hiring practices that have to change to make a difference on this issue."

A study of almost 500,000 adolescents in 67 countries found girls were at least as strong in science and math as boys in 60% of PISA countries, and were capable of college-level STEM studies.

ACM-W, ACM's Council on Women in Computing, which advocates internationally for the engagement of women in all aspects of the computing field, sponsors ACM Celebrations of Women in Computing, providing monetary and other support in order to connect women working/studying in technical fields and break down feelings of isolation.

The intention of ACM-W in supporting these celebrations, says the organization's chair, Jodi Tims, is to reach the broadest possible populations of women through an international network of self-sustaining small conferences, dovetailing when possible with ACM-W chapters.

Tims says 87 such Celebrations have been held since 2013, with a total 10,500 attendees through 2017. She says attendance has grown from about 1,500 in 2013-2014 to 5,800 for the first half of this year, and the number of countries in which Celebrations take place has grown from five at the outset to 16 this year.

Tims suggested a variety of things individuals can do to make certain their environments are inclusive, such as:

- Ensure everyone in a meeting, regardless of gender, have thechance to contribute to a discussion.
- Encourage young women to push back against negative peer pressure from both women and men to dissuade them from staying in computing.
- Mentor a female student interested in computing.
- Make certain hiring, tenure, and promotion committees, as well as teaching faculty and managers, understand how unconscious bias can affect their decisions, and help them to develop mechanisms that will disrupt those biases.

Tims points out that ACM "has the potential to set the standard for what it means to be an organization committed to solving issues of gender diversity in computing."

Further Reading

Stoet, G., and Geary, D.C.
The Gender-Equality Paradox in Science, Technology, Engineering, and Mathematics Education, *Psychological Science*, 2018.

Stout, J.G., and Blaney, J.M.
"But it *doesn't* come naturally": How effort expenditure shapes the benefit of growth mindset on women's sense of intellectual belonging in computing. *Computer Science Education* (pp. 1–14), http://bit.ly/2EIZMKP

Blaney, J.M., and Stout, J.G.
Examining the relationship between introductory computing course experiences, self-efficacy, and belonging among first-generation college women. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 69–74). New York, NY: ACM.

Sax, L.J., Zimmerman, H.B., Blaney, J.M., Toven-Lindsey, B., and Lehman, K.J.
Diversifying computer science departments: How department chairs become change agents for women and underrepresented minority students. *Journal of Women and Minorities in Science and Engineering*, 23(2), 101-119.

DuBow, W., Farmer, R. Wu, Z., and Fredrickson, M.
Bringing Young Women into Computing through the NCWIT Aspirations in Computing Program, *Communications of the ACM*, December 2013.

Tims, J.L.
Achieving Gender Equity: ACM-W Can't Do It Alone, *Communications of the ACM*, February 2018.

Barr., V.
Gender Diversity in Computing: Are We Making Any Progress? *Communications of the ACM*, April 2017.

Esther Shein is a freelance technology and business writer based in the Boston area.

DOI:10.1145/3232037

Kashif Ali and Kurtis Heimerl

▸ Michael L. Best, Column Editor

Global Computing
Designing Sustainable Rural Infrastructure Through the Lens of OpenCellular

Understanding the unique local context, as well as technical considerations, are essential components of successful project deployment.

Rural areas are defined, in part, by their lack of infrastructure. In many parts of this column author Kurtis Heimerl's home state of Alaska, communities lack power infrastructure and learn to set up and use generators as a solution; in column author Kashif Ali's home country of Pakistan, filters are deployed to provide clean water in areas without a generalized potable source. Building sustainable infrastructure solutions for these types of places—which exist in some way in all countries with substantive rural areas—is a complex problem. While many (if not most) issues are not technical in nature (instead involving things like local buy-in), research has shown designing the technical portion of an intervention with a deep understanding of the local context is important to the success of the project.[2] This understanding includes the capabilities, affordances, knowledge, and infrastructure available within a community; as well as the ability to leverage that understanding to build technologies that are inexpensive, robust, and understandable by rural users.

OpenCellular (OC),[8] an initiative of the Telecom Infrastructure Project (TIP),[10] is an open source hardware and software infrastructure platform that implements a cellular access point, either GSM or LTE. It was created to provide coverage to the hundreds of millions of people living in areas currently without cellular coverage and includes a variety of optimizations across hardware, software, and business models to better fit the needs and capabilities of these rural communities. OC's design leverages the extensive research literature on operating in rural areas as well as our personal experiences in places like Kenya, Indonesia,[5] and the Philippines.[1] In this column, we describe four important design choices in the context of the OpenCellular project: platform, power and networks, business models, and customizability. While the OC developers are still conducting their initial deployments and the eventual impact of the project is not known, we believe these designs are crucial to the long-term success of the platform.

User Platform

The first design choice is what device to leverage for your intervention. A common viewpoint is that we live in a world of ubiquitous access; everyone has access to at least basic cellular connectivity. Unfortunately, this is not the case.

Figure 1. An OpenCellular operator working in remote location.

Figure 2. OC operator in community location.

Recent estimates have GSM cellular coverage at 80% of the world.[11] These networks have stopped expanding as operators instead invest in higher-revenue urban 3G or LTE installations. Ironically, while ubiquitous connectivity may be a myth, the ubiquity of the cellular phone itself is not. In our travels, we have yet to come to a location where these devices were not already woven into the fabric of the community. For example, when we first set up our network in Papua, Indonesia, the network recorded over 3,000 unique mobile phones despite being a four-hour drive from the closest cellular network.[9] It may seem counterintuitive for there to be cellular phones where there is no network, but the devices are more than just phones. They are also rugged rural entertainment consoles, with built-in battery power, speakers, and headphone jacks. Where there is no cell network there is often no radio and instead, people bring their entertainment with them. All the while, many members of the rural community regularly travel to dense urban areas where there is coverage and use their phones there.

This situation—a large installed base of mobile phones in areas without any cellular access—provides an opportunity for novel connectivity solutions. OC is designed to meet the current capabilities and needs of users in these communities. The first revision, OC-SDR (OpenCellular Software Defined Radio), is a GSM cellular base station with support for basic GPRS/Edge data connectivity. The second revision, OC-LTE is an LTE-based extension of the platform. These two access points allow locals to get on the network using their existing phones. Then the operator can determine various services the community would like (for example, using IVR for low-literate populations) and eventually upgrade to broadband as economics and availability of other associated infrastructure (such as power for smartphones) matures.

Power and Backhaul

To keep costs low, one must leverage what infrastructure is available in the remote communities. Surana et al.[9] found that grid power was unreliable in rural India, producing both brownouts and voltage spikes capable of destroying equipment. They also learned that the lack of general Internet access was a huge issue in diagnosing failures. OpenCellular mitigates these problems by building backhaul and power solutions into the system. The platform includes power cleaning, variable input voltages, and support for Power-over-Ethernet (PoE). The system also supports PoE's power sourcing equipment (PSE) standard, allowing the OpenCellular access point to "daisy-chain" power to phone chargers or even another OC instance. To better support renewable energy sources, OC also features two internal solar charge controllers for external sealed lead acid and internal lithium ion batteries. The internal lithium-ion battery, with a built-in UPS system, works as a backup to allow the system to fail gracefully when the local grid fails. This suite of power cleaning and support systems allows OpenCellular systems to coexist in the chaotic reality of rural power.

While ubiquitous connectivity may be a myth, the ubiquity of the cellular phone itself is not.

For network, OpenCellular includes a built-in out-of-band satellite backhaul. This is not designed to be used for daily communications (as the cost could be prohibitive) but instead to reduce the cost of debugging a failed network. Operators will not need to send an engineer out to the system (see Figures 1 and 2) if the backhaul has failed and instead can use the satellite link to gather critical data about the issues with the system, such as the status of individual hardware components or stability of power or backhaul subsystems. This collected information can also be relayed to the local maintainer (potentially by the access point itself) who can then assist in maintenance (for example, cleaning the solar panels) even when the main power and backhaul is down.

The highly variable power and network situation in rural areas invites another design imperative: OC optimizes for low cost over reliability. In rural areas, the vast majority of downtime will be due to failures in the related infrastructure, including power and network. Increasing the reliability of the OC system itself will only marginally increase the overall uptime of the network. Instead, it is better to be cheap and easy to replace or repair.

FIGURE 1 IMAGE BY MIKE DARCY OF AMN; FIGURE 2 IMAGE BY EUAN MILLAR OF ENSEMBLE

Business Models

Thirdly, you must design your intervention to sustain. That involves creating business models that encourage local participation and support. While existing business models for cellular exist (and are quite lucrative), many rural areas remain underserved. Even with the power and network advances mentioned here, it is always going to be expensive to send engineers and equipment to remote parts of the country for installations and maintenance. For coverage to reach the entire world in a sustainable manner, OC must support a variety of different business models that can cover the diversity of the rural world. OC enables two key business models: community-focused and traditional (see Figures 3–5).

In many remote rural areas, much of the infrastructure is owned and operated by local agents. Co-production[7] is one model that makes use of this fact, with core infrastructure such as power and water built and operated in close collaboration with the populations served. Galperin[4] suggested extending local ownership to cellular, with smaller local telecoms providing service. OpenCellular supports these local business models. CommunityCellularManager (CCM)[3] is one OC-supported software suite that allows small communities to operate their own small OpenCellular-based networks. It provides both client and cloud support for management, routing, and interconnect. With CCM, the local rural community can then personally maintain and operate the network.

OC can also operate as a traditional cellular access point, supporting a variety of open and closed-source basebands and cellular stacks that when configured can connect to traditional core networks (EPC, in case of OC-LTE). This allows existing incumbents to utilize OC to decrease the cost of their rural installations while requiring minimal changes to the rest of their infrastructure.

Figure 3. Handwritten field notes: Current model.

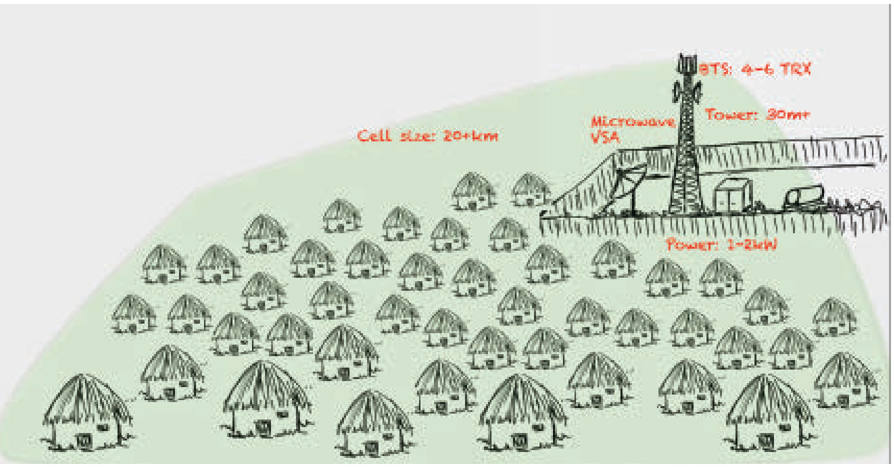


Figure 4. Handwritten field notes: De facto deployment.

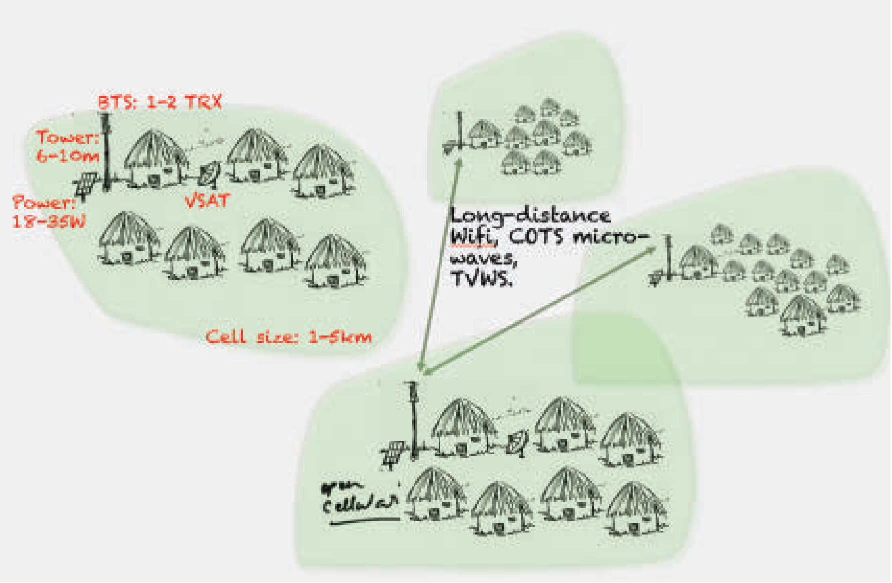


Figure 5. Handwritten field notes: Bottom-up model.

Customizability

While OpenCellular has been designed with our own rural experiences in mind, the appropriateness of specific technologies will vary widely across areas. For example, OC's built-in satellite backhaul may be appropriate where wireless is used for backhaul[6] but overly expensive if the installation is backhauled over a more robust medium such as fiber. Similarly, new technologies such as 5G or LoRa[6] may see rapid uptake in the next few years and overtake LTE. For this reason, OpenCellular needs to be extensible and customizable to enable new access models and new technologies. Enabling this customizability in OpenCellular consists of two distinct design choices: modularity and open source.

The OpenCellular hardware is designed in a modular fashion, with individual components like the power subsystem separated from the rest of the system. This allows organizations building OC devices to "pick and choose" the features that are relevant to their context. For instance, daughter-boards allow for the base system to expand to radio technologies outside of GSM and LTE and into future technologies like LoRA. Similarly, the components listed here, such as the built-in battery backup, can be removed from the board during manufacturing to save cost if the grid power is expected to be clean. Lastly, new subsystem modules, such as an inexpensive WiFi hotspot, could be added before manufacturing.

Lastly, we need to enable these new designs to scale. To do this, OC has been released as open source hardware, including all the schematics, layout, CAD, BoM, and firmware needed to enable large-scale industrial manufacturing. Additionally, all testing software is also open source, so anyone (either an OEM/CM or university students) can replicate and produce OC hardware at the same quality level as current industrial partners; the software is available at https://bit.ly/2xREpUD. Because of the rich suite of software and hardware supporting the platform, motivated organizations can extend OpenCellular to meet their needs and then manufacture the equipment at scale locally in the country, increasing local capacity, reducing costs, and stimulating the local economy.

Conclusion

Designing infrastructure for rural areas that can leverage the local context—the skills, knowledge, and affordances of the communities that live there—is a difficult task. With OC we chose to focus on four key elements. The first is the user platform, ensuring the intervention uses technologies that are common and available. The next is ensuring we support a diverse range of power and network technologies as well as business models—some of the key differentiators between rural communities. Lastly, we recognize the limitations of our own designs and capabilities by releasing OC as open source hardware, complete with all of the designs necessary to modify and manufacture the solution. It is our hope that, through the lens of OpenCellular, readers can see how to similarly design their own interventions with these concerns in mind. While we are hopeful that OpenCellular itself brings connectivity to the world, rural access problems always require holistic solutions that are driven by the needs, abilities, and limitations of the communities themselves. As such, we also aspire to allow motivated individuals and organizations take OpenCellular, expand it to fit their needs, and create a diverse ecosystem of rural access solutions. Join us at https://bit.ly/2JoFa8Y to participate in this process.

OpenCellular enables two key business models: community-focused and traditional.

References

1. Barela, M.C. Towards building a community cellular network in the Philippines: Initial site survey observations. In *Proceedings of the International Conference on Information and Communication Technologies and Development*, 2016.
2. Chetty, M., Tucker, W., and Blake, E. Developing locally relevant software applications for rural areas: A South African example. In *Proceedings of the 2004 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, 2004.
3. CommunityCellularManager; https://bit.ly/2Lvgxs5.
4. Galperin, H. and Bar, F. The Microtelco opportunity: Evidence from Latin America. *Information Technologies & International Development 3*, 2 (Feb. 2006).
5. Heimerl, K. et al. Local, Sustainable, small-scale cellular networks. In *Proceedings of the Sixth International Conference on Information and Communication Technologies and Development: Full Papers—Volume 1*, New York, NY, 2013.
6. Mikhaylov, K., Petaejaejaervi. J., and Haenninen, T. Analysis of capacity and scalability of the LoRa low power wide area network technology. In *European Wireless* 2016.
7. New Economics Foundation. Co-production: A manifesto for growing the core economy. New Economics Foundation, 2008.
8. OpenCellular Telecom Infra Project; https://bit.ly/2hJE0u2
9. Surana, S. et al. Beyond pilots: Keeping rural wireless networks alive. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA, 2008.
10. Telecom Infrastructure Project; https://bit.ly/2sF1hlj
11. Touchard, G. *The State of Connectivity in Emerging Markets.* OpenCellular Workshop, Nairobi, Kenya, 2017.

Kashif Ali (kashi@fb.com) is an engineer at Facebook Inc., USA.

Kurtis Heimerl (kheimerl@cs.washington.edu) is an assistant professor at the University of Washington, USA.

The Telecom Infra Project is offering grant opportunities for organizations looking to use OpenCellular; see https://oc.telecominfraproject.com/opencellular-grant-program/ for more information. Applications will be due in Fall 2018.

Calendar of Events

August 10–12
HPG '18: High-Performance Graphics,
Vancouver, BC, Canada,
Sponsored: ACM/SIG,
Contact: Steven Molnar,
Email: molnar@nvidia.com

August 11
DigiPro '18: The Digital Production Symposium,
Vancouver, BC,
Sponsored: ACM/SIG,
Contact: Corban John Gossett,
Email: corban@nimblecollective.com

August 12–16
SIGGRAPH '18: Special Interest Group on Computer Graphics and Interactive Techniques Conference,
Vancouver, BC,
Sponsored: ACM/SIG,
Contact: Roy Anthony,
Email: barcodemen@gmail.com

August 12–16
ICER '18: International Computing Education Research Conference,
Espoo, Finland,
Sponsored: ACM/SIG,
Contact: Andrew K. Petersen,
Email: andrew.petersen@utoronto.ca

August 19–23
KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
London, U.K.,
Co-Sponsored: ACM/SIG,
Contact: Faisal Farooq,
Email: ffarooq@gmail.com

August 20–25
SIGCOMM '18: ACM SIGCOMM 2018 Conference,
Budapest, Hungary,
Sponsored: ACM/SIG,
Contact: János Tapolcai,
Email: tapolcai@tmit.bme.hu

August 23–29
ICFP '18: ACM SIGPLAN International Conference on Functional Programming,
St. Louis, MO,
Sponsored: ACM/SIG,
Contact: Robby Findler,
Email: robby@plt-scheme.org

DOI:10.1145/3232039

Mark Guzdial and Amy Bruckman

Education

Providing Equitable Access to Computing Education

Seeking the best measures to reach advantaged and less-advantaged students equally.

A SEAT IN a computer science classroom is one of the hottest tickets on American campuses today. Undergraduate enrollment in computer science is at an all-time high, and many of the students in those classes (even beyond the introductory level) are non-CS majors. This surge is so large and unprecedented that the U.S. National Academy of Science wrote a report to document the surge and suggest strategies for managing the growth.[a]

Interest in learning computer science extends into primary and secondary schools as well. Several countries have national efforts to provide CS education to every student in every school. Among the 50 U.S. states, 36 have statewide policies promoting CS education. We are struggling to deal with all this interest, but it is a good problem to have. We have something that everyone wants. The problem is who is getting it.

Undergraduate education is still mostly the domain of the rich. Low- and middle-income families are much less likely to get access to higher education than the rich, as reported by the *Equality of Opportunity* project at Stanford.[b] Most high schools in the U.S. are not offering computer science, and wealth is a significant predictor of whether a school offers CS.[3]

IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

One obvious solution is educational technology. We could offer online CS courses, at all levels from primary and secondary school, through undergraduate and graduation education, and beyond to life-long learning. Computer scientists invented MOOCs (Massive Open Online Courses) to provide CS education to as broad an audience as possible. The first MOOCs were invented by CS faculty at Stanford to offer CS courses online. The first start-ups offering MOOCs—Coursera and Udacity—were led by Stanford CS faculty. The authors of this column are both

a https://bit.ly/2sDPJ1t
b https://bit.ly/19bjppE

faculty at Georgia Institute of Technology, where our Online-MS in CS (OMS CS) was praised by President Barack Obama for its innovative accessibility and low cost.

In the last six years, we have also come to understand who is taking MOOCs. We now know that MOOC students tend to be older than traditional college students, have above-average wealth, and are well educated. MOOCs do not serve the masses. They do not serve to replace traditional education, but to augment it. They do not "democratize education" as many had hoped.

Recent innovations in online learning are proving to have a "rich get richer" effect—those already likely to succeed benefit, and those left behind are left at an increased disadvantage. We argue that *our current technology is further undermining educational equity*. Computer science departments have an ethical mandate to do better.

Rawlsian Justice

Is it OK to work to help the already advantaged? Certainly, the latest smartphone and the latest luxury car are unapologetically created for the privileged. Our current education system is regrettably not so different from a luxury car—for a price, deluxe experiences are available that advantage the children of the rich and help reproduce their privilege. However, we have higher aspirations. We hope education can serve as a leveler, helping everyone to reach their full potential.

Some difference in privilege is necessary and even desirable to create a thriving culture. How much privilege is OK? What are our obligations to work toward greater equity, for a society that aspires to be just? These profound questions were most eloquently addressed by the philosopher John Rawls. Rawls was an ethicist who argued that for a just society, "social and economic inequalities are to be arranged so that they are both to the greatest benefit for the least advantaged and attached to offices and positions open to all under conditions of fair equality of opportunity."[6] Rawls called this "the difference principle."

Most undergraduate computer science majors are taught about Rawlsian Justice. ABET-accredited programs must include a course in computing and society, which includes ethical frameworks. This definition of justice is ours. It is the one we teach our own students.

We used to think MOOCs were going to change higher education and would democratize education.

Using Evidence to Tell Us If We Are Reaching the Least Advantaged

We used to think MOOCs were going to change higher education and would democratize education. In 2012, a reasonable person might have seen development of MOOCs as a way to bridge social and economic inequities. By creating MOOCs, CS departments could reasonably claim they were using their privilege to provide great benefit to the least-advantaged members of society.

Today, we have evidence MOOCs do not work like that.

People who take MOOCs already have access to education and tend to be wealthy. Over 60% of MOOC participants already have undergraduate degrees.[1] People who take MOOCs tend to be wealthy. A 2015 paper[5] reports, "[MOOC] registrants on average live in neighborhoods with median incomes approximately 45 standard deviations higher than the U.S. population."

Analyses of Georgia Tech's OMS CS shows the students who apply to the program are demographically different from those who take a face-to-face MS CS program.[4] The average OMS CS applicant is a 34-year-old mid-career American, while the average in-person applicant is a 24-year old non-American. MOOCs reach a population that would be unlikely to get a master's degree in another way. The program is transformative for mid-career professionals—people who are already successful, but aspire to more in their careers. As information technology is increasingly becoming critical to every aspect of our society, OMS CS is playing a global role in preparing us for the future. MOOCs are well worth offering, but the population being served tends not to be the least advantaged.

We now know that MOOCs as we have used them so far violate Rawls' Difference Principle—we are further advantaging the already advantaged. We have an ethical mandate to do better.

We Have to Check If We Are Doing Better

How do we reach the least-advantaged students? Around the U.S., we can see CS departments trying a lot of ways to provide access to CS education. They are offering summer camps, putting their undergraduates into high school and elementary classrooms to help teach CS, or creating "road shows" to demonstrate computer science to elementary or secondary school students who may not know what computer science is. Some of these work. Many do not.

Often, providing computing educational opportunities to "everyone" operationally means only the most-advantaged students actually get access. Free and open summer camps are often filled first by the most-privileged students who tend to hear about the camps and fill them before less-privileged students get a chance.

It is challenging to figure out how to make free and open resources available to less-advantaged students. For example, our colleague Betsy DiSalvo found that many free CS learning resources are never discovered by disadvantaged families simply because the families do not know the right terms to search for.[2] The for-profit companies are better at tailoring their websites so their resources are the first to appear for the terms that (for example) immigrant families use when searching for learning resources.

Researchers are still working to understand why MOOCs fail students from less-advantaged backgrounds. Access is part of the problem. An experiment offering Udacity MOOCs to San Jose State University students was ended early because the online students had disappointing performance compared to the face-to-face students. Part of the problem there was that the online students did not always have access to broadband Internet when at

CS departments should offer interventions that measurably reach advantaged and less-advantaged students equally.

home. Other researchers are exploring the use of "nudges" to convince students they can succeed and MOOCs are worth the effort.

We do not believe MOOCs are fundamentally unsuited to struggling learners. We need to continue the design work to make scalable MOOC or MOOC-like solutions work more effectively for less-advantaged students. We also need to design and offer non-MOOC alternatives for students who need greater support.

Our Proposal: Match the Learning Opportunities

We propose that CS departments who offer MOOCs must balance the opportunities they are offering to advantaged students (like MOOCs) by pairing them with opportunities for less-advantaged students. CS MOOCs fill a need and should be offered and even expanded. But they do not meet the definition of Rawlsian justice. CS departments should offer interventions that measurably reach advantaged and less-advantaged students equally. Dollar for dollar, student for student, initiatives that reach more advantaged students need to be matched with those that reach less-advantaged ones.

The U.S. National Science Foundation has launched a new pilot effort to expand engagement in broadening participation in computing (BPC) activities by awardees in their Computer and Information Science and Engineering (CISE) directorate.[c] They aim to increase the number of computer scientists who are working to make computing education more accessible. Some CISE proposals already require BPC plans, and more proposals will be required in the future. Proposal writers will be provided a set of resources, and they will be encouraged to participate in meaningful activities that have successfully reached underrepresented populations. Example programs include the Distributed Research Experiences for Undergraduates (DREU) program[d] from the Computing Research Association's Committee on the Status of Women in Computing Research (CRA-W) and the NCWIT Aspirations award.[e] There are things we can do that have a measurable impact on increasing equitable access to computing education, and it is the responsibility of the entire CS community to do them and assess whether they are working.

Conclusion

CS learning opportunities are highly sought after. CS departments have an ethical obligation to ensure access to these opportunities is equitable. We propose the use of empirical measures, to ensure we are reaching advantaged and less-advantaged students equally.

c https://bit.ly/2sPbUB7

d https://bit.ly/2kdd2hk

e https://bit.ly/1EBXw2o

References

1. Chuang, I. and Ho, A. HarvardX and MITx: Four years of open online courses—Fall 2012–Summer 2016 (Dec. 23, 2016); https://ssrn.com/abstract=2889436
2. DiSalvo, B., Reid, C., and Khanipour Roshan, P. They can't find us: The search for informal CS education. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. ACM, New York, NY, USA, 2014, 487–492; DOI: http://dx.doi.org/10.1145/2538862.2538933
3. Ericson, B. and Guzdial, M. Measuring demographics and performance in computer science education at a nationwide scale using AP CS data. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. ACM, New York, NY, USA, 2014, 217–222; DOI: http://dx.doi.org/10.1145/2538862.2538918
4. Goodman, J., Melkers, J. and Pallais, A. *Can Online Delivery Increase Access to Education?* Faculty Research Working Paper No. w22754. National Bureau of Economic Research, 2016.
5. Hansen, J.D. and Reich, J. Socioeconomic status and MOOC enrollment: Enriching demographic information with external datasets. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge (LAK '15)*. ACM, New York, NY, USA, 2015, 59–63. DOI=http://dx.doi.org/10.1145/2723576.2723615
6. Rawls, J. *A Theory of Justice*. Harvard University Press, Cambridge, MA, 1971.

Mark Guzdial (guzdial@cc.gatech.edu) is a professor in the School of Interactive Computing at Georgia Institute of Technology, and in Computer Science & Engineering at the University of Michigan, MI, USA.

Amy Bruckman (asb@cc.gatech.edu) is a professor and associate chair in the School of Interactive Computing at Georgia Institute of Technology, Atlanta, GA, USA.

DOI:10.1145/3232041

George V. Neville-Neil

Kode Vicious

Every Silver Lining Has a Cloud

Cache is king. And if your cache is cut, you are going to feel it.

IMAGE COLLAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

Dear KV,

My team and I have spent the past eight weeks debugging an application performance problem in a system we moved to a cloud provider. Now, after celebrating that achievement, we thought we would tell you the story and see if you have any words of wisdom.

In 2016, our management decided that—to save money—we would move all our services from self-hosted servers in two racks in our small in-office datacenter to the cloud so we could take advantage of the elastic pricing available from most cloud providers. Our system uses fairly generic, off-the-shelf, open source components, including Postgres and Memcached, to provide the back-end storage to our Web service.

Over the past two years we built up a good deal of expertise in tuning the system for performance, so we thought we were in a good place to understand what we needed when we moved the service to the cloud. What we found was quite the opposite.

Our first problem was very inconsistent response times to queries. The long tail of long queries of our database began to grow the moment we moved our systems into the cloud service, but each time we went to look for a root cause, the problem would disappear. The tools we would normally use to diagnose the issues we found on bare metal also gave far more varied results than expected. In the end, some of the systems could not be allocated elastically but had to be statically allocated, so the service would behave in a consistent manner. The savings management expected were never realized. Perhaps the only bright side is that we no longer have to maintain our own deployment tools, because deployment is handled by the cloud provider.

We wonder, is this really a common problem, or could we have done something that would have made this transition less painful?

Rained on Our Parade

Dear Rained,

Clearly, your management has never heard the phrase, "You get what you pay for." Or perhaps they heard it and did not realize it applied to them. The savings in cloud computing comes at the expense of a loss of control over your systems, which is summed up best by the popular sticker "The Cloud Is Just Other People's Computers."

All the tools you built during those last two years work only because they have direct knowledge of the system components down to the metal, or at least as close to the metal as possible. Once you move a system into the cloud, your application is sharing resources with other, competing systems, and if you are taking advantage of elastic pricing, then your machines may not even be running until the cloud provider deems them necessary. Request latency is dictated by the immediate availability of resources to answer the incoming request. These resources include CPU cycles, data in memory, data in CPU caches, and data on storage. In a traditional server, all these resources are controlled by your operating system at the behest of the programs running on top of the operating system; but in a cloud, there is another layer, the virtual machine, which adds another turtle to the stack, and even when it is turtles all the way down, that extra turtle is going to be the source of resource variation. This is one reason you saw inconsistent results after you moved your system to the cloud.

Let's think only about the use of CPU caches for a moment. Modern CPUs gain quite a bit of their overall performance from having large, efficiently managed L1, L2, and sometimes L3 caches. The CPU caches are shared among all programs, but in the case of a virtualized system with several tenants, the amount of cache available to any one program—such as your database or Memcached server—decreases linearly with the addition of each tenant. If you had a beefy server in your original colocation facility, you were definitely gaining a performance boost from the large caches in those CPUs. The very same server running in a cloud provider is going to give your programs drastically less cache space with which to work.

With less cache, fewer things are kept in fast memory, meaning your programs now need to go to regular RAM, which is often much slower than cache. Those accesses to memory are now competing with other tenants that are also squeezed for cache. Therefore, although the real server on which the instances are running might be much larger than your original hardware—perhaps holding nearly a terabyte of RAM—each tenant receives far worse performance in a virtual instance of the same memory size than it would if it had a real server with the same amount of memory.

Let's imagine this with actual numbers. If your team owned a modern dual-processor server with 128 gigabytes of RAM, each processor would have 16 megabytes—not gigabytes—of L2 cache. If that server is running an operating system, a database, and Memcached, then those three programs share that 16 megabytes. Taking the same server and increasing the memory to 512 gigabytes, and then having four tenants, means the available cache space has now shrunk to one-fourth of what it was—each tenant now receives only four megabytes of L2 cache and must compete with three other tenants for all the same resources it had before. In modern computing, cache is king, and if your cache is cut, you are going to feel it, as you did when trying to fix your performance problems.

Most cloud providers offer systems that are non-elastic, as well as elastic, but having a server always available in a cloud service is more expensive than hosting one at a traditional colocation facility. Why is that? It is because the economies of scale for cloud providers work only if everyone is playing the game and allowing the cloud provider to dictate how resources are consumed.

Some providers now have something called Metal-as-a-Service, which I really think ought to mean a 1980s-era metal band shows up at your office, plays a gig, and smashes the furniture—but alas, it is just the cloud providers' way of finally admitting cloud computing is not really the right answer for all applications. For systems that require deterministic performance guarantees to work well, you really must think very hard about whether or not a cloud-based system is the right answer, because providing deterministic guarantees requires quite a bit of control over the variables in the environment. Cloud systems are not about giving you control; they are about the owner of the systems having the control.

KV

Related articles on queue.acm.org

Cloud Calipers
Kode Vicious
https://queue.acm.org/detail.cfm?id=2993454

20 Obstacles to Scalability
Sean Hull
https://queue.acm.org/detail.cfm?id=2512489

A Guided Tour through Data-center Networking
Dennis Abts and Bob Felderman
https://queue.acm.org/detail.cfm?id=2208919

George V. Neville-Neil (kv@acm.org) is the proprietor of Neville-Neil Consulting and co-chair of the *ACM Queue* editorial board. He works on networking and operating systems code for fun and profit, teaches courses on various programming-related subjects, and encourages your comments, quips, and code snips pertaining to his *Communications* column.

DOI:10.1145/3213766; 10.1145/3231050

Ehud Shapiro/Douglas Schuler

Point/Counterpoint
Democracy and E-Democracy

A discussion of the possibility of supplanting traditional representative democracy with e-democracy.

DOI: 10.1145/3213766

Point: Foundations of E-Democracy

Considering the possibility of achieving an e-democracy based on long-established foundations that strengthen both real-world democracies and virtual Internet communities.

Ehud Shapiro

IMAGE BASED ON PAINTING BY JEAN-JACQUES-FRANÇOIS LE BARBIER, 1789 (PUBLIC DOMAIN)

THE INTERNET REVOLUTION of democracy, which will transform earthly representative democracies by employing the communication and collaboration capabilities of the Internet, has yet to come. For this *Communications* Point/Counterpoint discussion, I enlist the wisdom of our forefathers to lead the way. By consulting the *1789 Declaration of the Rights of Man and of the Citizen*,[4] I distill core values of democracy and derive from them requirements for the foundations of e-democracy. Building on these foundations can usher in the urgently needed revolution of democracy.

Representative democracy is in retreat worldwide,[1,5,6] as many democracies transform into oligarchies, plutocracies, or even kleptocracies. A key reason is lack of respect of democracy's basic tenet—equality of rights—as the rich, the powerful, and the connected increasingly dominate who gets nominated, who gets elected, and what the elected do. The forefathers of democracy have identified this to be "… the sole cause of public calamities and of the corruption of governments."[4]

The Internet, on the other hand, is revolutionizing industry after industry, leaving older ways of human conduct in the dustbin of history. Yet, it has not changed the basic workings of democracy: Representative democracy today functions essentially as it did 200 years ago (Internet-enabled disruptions of elections notwithstanding).

How could this be? Why has an Internet revolution of democracy not yet occurred, despite the pressing need for it and the apparent clear ability of the Internet to deliver it? I believe a key reason is that amalgamating "Internet" and "Democracy" into an Internet democracy, or *e-democracy*, is more difficult than it seems.

E-democracy has at least two meanings: Using the Internet to strengthen real-world democracies,[1,14] and democratic conduct of virtual Internet communities.[3] When viewed as objectives they coalesce, as one entails or requires the other.

Amalgamating "Internet" and "Democracy" presupposes universal Internet access as well as Net neutrality and freedom; their absence undermines the legitimacy of e-democracy, as a regime can exclude an oppressed minority, or a service provider can make e-democracy a super-premium service, excluding the poor.

Even if the Internet infrastructure is universally accessible, neutral, and fair, utilizing an existing Internet application such as Facebook and its siblings as a foundation for e-democracy is a non-starter: They are prone to duplicate and fake accounts and, crucially, to nondemocratic oversight, control, and arbitrary intervention by their owners. Even Wikipedia, a hallmark of Internet participation, is governed neither by its readers nor by its editors, but by an appointed board that has full legal authority to shut it down, for example, to avert bankruptcy.

Hence, new *foundations for e-democracy* are needed. I envision these foundations to simultaneously support the democratic conduct of all types of communities: Associations, clubs, unions, cooperatives, organizations, movements, and political parties; and at all levels—local, national, transnational, and international; eventually including cities, states, and federations; and, ultimately, uniting the entire humanity in a global e-democracy.

Among these communities, the pivot for revolutionizing earthly democracies may be Internet-resident democratic political parties, or e-parties. Only by winning real-world elections, e-parties can export the participatory practices of e-democracy from their inner workings to real-world governments, enacting legislation that gradually supplants traditional representative democracy by e-democracy.

But what are these foundations? Who could guide us in their construction? A standard method in requirements engineering is to interview the prospective customer. The prospective "customer" for e-democracy is humanity at large. Hence, in lieu of an interview, I enlist one of humanity's most inspiring documents: The 1789 French *Declaration of the Rights of Man and of the Citizen*[4] (henceforth: *Declaration*), which offers a concise, clear, and bold expression of the essence of democracy. I study its Articles, extract from them core democratic values, and derive from these values requirements for the foundations of e-democracy.

Core Values of Democracy

Here, I list the core democratic values extracted from the Articles (marked by **A**) of the Declaration (Interpreting Man→Person, Citizen→Member, and Nation→Community):

1. **Sovereignty:** The *Declaration*'s **Article III (A3)** states "The principle of any sovereignty resides essentially in the Nation. No body, no individual can exert authority that does not emanate expressly from it." We interpret this principle to mean that **the members of an e-democracy are its sovereign**.

> **The prospective "customer" for e-democracy is humanity at large.**

2. **Equality**: **A1** states that "Men are born and remain free and equal in rights. ...". Together with **A3** they imply that sovereignly must be equally shared, often stated as *one person-one vote*. But there is more to equality than the right to vote. **A4** states that the law is the expression of the general will and that all people have the right to contribute to its formation; and equally so, according to **A1**. **A6** further states that all people, being equal in the eyes of the law, are equally admissible to all public posts. Equality extends not only to rights but also to obligations: **A12–14** ascertain the need for public services and for equally sharing their financing among members, but progressively, according to their ability to pay.

To summarize, all members of a democracy must have equal capacity to act as voters, discussants, proposers and public delegates, as well as share progressively the burden of public expenditures.

3. **Freedom**: **A1** states that "men are born and remain free." The nature of this freedom is further elaborated in other articles: **A10–11** espouse the **freedom of expression** within the limits if the law. **A5** proclaims the freedom to take any action that is not harmful to others. Among those implied freedoms I note the **freedom of assembly**[3] granting any group of people the freedom to assemble, and the **subsidiary principle**, granting such a group the freedom to make decisions that pertain to them.

4. **Transparency: A14–15** require that the conduct of public agents and the collection and expenditure of public funds be transparent. Furthermore, **A2** states that the goal of any political association must be the conservation of the natural and imprescriptible rights of man: **liberty, property, safety and resistance against oppression**. This can be ascertained in an ecosystem of e-democracies only if the decisions of each are transparent to the others.

5. **Property and Privacy: A17** recognizes the right for property and its private use, which, extended to our times, incorporates the right for the ownership and privacy of information. The right to safety and resistance against oppression (**A2**) entails voter privacy to resist coercion.

6. **Justice:** Revolt against unjust rulers was crucial to the emergence of democracy, and justice is the focus of early charters of democracy such as the English Magna Carta[12] and the French *Declaration*. Indeed, **A1** and **A4–13** address the equal and just conception, application, and enforcement of the law. Furthermore, **A16** states that a **constitution** is needed to guarantee the rights of citizens and the separation of the powers of government.

Requirements of Foundations of E-Democracy

I now aim to derive from these core democratic values requirements for the foundations of e-democracy.

1. **Sovereignty:** Internet communities today, from the local bulletin board to almighty Facebook, are dictatorial, with an omnipotent administrator who determines who gets in, who is thrown out, and what actions each member may take. The administrator also has the capacity to shut down the community and annihilate its recorded history at will. Furthermore, communities like Facebook employ rule-by-decree like bygone Middle Ages fiefdoms. The owner, like a feudal lord, sets the rules (sometimes in secrecy), tries members for breaching them, and executes the punishment. The members, like serfs, toil for the financial benefit of the lord while having no (intellectual) property, civil rights, or voting rights. They have no say on their remuneration or tax, on community rules of conduct or their enforcement, or on the election of community leadership. In the event of a bankruptcy or hostile takeover, the entire community and its recorded history may be annihilated, with community members being helpless bystanders. All this clearly

violates all of democracy's core values: sovereignty, equality, freedom, transparency, property, privacy, and justice.

First, I consider the question of ownership. Any seemingly sovereign e-democracy that resides on computers operated by a third party could be unplugged at its will, or its default, rendering sovereignty meaningless. Hence, in the context of an e-democracy, sovereignty requires ownership.

How can the members of an e-democracy be the sovereign and hence necessarily the owner? Advances in cryptocurrencies and blockchain technology provide the first example. In a DAO (Decentralized Autonomous Organization),[3] built on top of Ethereum, the dictatorial system administrator is replaced by a *smart contract*, namely an autonomous, incorruptible, transparent, and persistent software agent, programmed to obey democratic decisions (albeit with *one coin–one vote*, not *one person–one vote*). The DAO operates on a distributed computer network with no central ownership. A few caveats: First, an early DAO venture capital fund had a bug that allowed a malicious member to syphon its funds. Smart-contract programming in general and the DAO architecture in particular have yet to mature to offer a sound foundation for e-democracy. Second, Ethereum and Bitcoin, while having distributed control in theory, have a core group of miners that could control and subvert them should they decide to join forces, a risk that a future e-democracy at the national or global scale cannot afford. Third, current proof-of-work consensus protocols of public blockchains incentivize inconceivable and unsustainable waste of energy, which cannot be endorsed by any moral person or organization. Fourth, a replicated ledger such as Ethereum and Bitcoin could not support the high-throughput transaction rate and response time required by a national or global e-democracy; a distributed ledger architecture is needed. Fifth, to foster participation rather than greed, a democratic cryptocurrency should reward participation,[8] rather than capital-intensive coin-mining; the globally unique digital identities required for e-democracy, discussed later, may afford an egalitarian cryptocurrency.[4,8] The economy of a democratic cryptocurrency could be programmed with democratically instituted taxes and budgets[9,15] to operate the e-democracy.

In summary, a distributed public ledger employing a democratic cryptocurrency and programmed to adhere to democratic control could ensure the members of an e-democracy are its sovereign and owner.

2. **Equality:** Equality entails *one person–one vote*. Yet e-democracies consist of digital identities, not people. Requiring *one digital identity–one vote* is not enough, as most existing systems allow a person to create as many digital identities as one wishes.

To support equality in an e-democracy, a new notion of digital identity must be devised that is truthful, unique, persistent, and owned by the person it represents. Otherwise, if fake—the owner may vote on behalf of a non-existent person; if non-unique—the owner may cast multiple votes; if not persistent—the owner may terminate and shed an obligated identity and acquire a fresh one clear of obligations, eluding accountability; and if not owned by the person it represents—it grants its owner an extra vote at the expense of the person it represents.

While truthfulness is a common requirement, for example in credit card and mobile phone contracts, uniqueness and persistency are not, as a person may obtain numerous credit cards, mobile phones, and email accounts and terminate them at will. Government-issued identity numbers, often complemented with biometric attributes and incorporated in digital identity cards (such as e-Estonia or India's Aadhaar) may serve as a unique and persistent digital identity attribute.

However, e-democracies may transcend national boundaries, for example, in regional and international organizations. Realizing equality in global e-democracies is a bigger challenge: First, unhindered Internet access should be a recognized basic civil right and be provided universally. Second, some people, notably refugees, may have no verifiable national identity, yet should be granted participation in a global e-democracy. Third, people may have multiple citizenships, and without an additional notion of "global citizenship" with an associated globally unique digital identity, one may have multiple votes, violating equality. Fourth, malicious nondemocratic regimes may produce an arbitrary number of fake national identities and use them (in a Sybil attack[6]) to sway the vote of a global e-democracy in favor of their national interest.

A trustworthy notion of *global citizenship*; a mechanism to endow each global citizen with a truthful, persistent, and globally unique *global digital identity*; and a *global judiciary* empowered to revoke fake or duplicate global digital identities and to transfer stolen identities back to their rightful owners, as well as to prosecute the perpetrators of these crimes, are all needed to ensure equality in a global e-democracy.

3. **Freedom:** As **freedom of expression** is granted within the limit of the law, its realization requires a *constitution* that determines these limits and a *judiciary* that enforces them, discussed here. **Freedom of assembly** can by realized by a software architecture that allows the unhindered formation of one e-democracy within another. To uphold the **subsidiary principle**, each subsidiary democracy should be able to undertake decisions that pertain to it, within the law.

4. **Transparency:** The structure of an e-democracy, its rules of conduct, its underlying technology, the source code of its software, as well as the decisions of its communities, the actions of its public delegates and its finances must all be transparent to all. (It is acknowledged that in an extreme scenario, resisting an oppressive regime may necessitate compromised transparency.)

5. **Property and Privacy:** The ownership of private data and its measured disclosure to third parties only as needed can be supported with self-sovereign identities.[11] Ensuring privacy of voters and avoiding coercion require advanced cryptographic techniques such as anonymous credentials[2] and coercion avoidance.

6. **Justice and Accountability:** To advance from the Internet Middle Ages and supplant Internet fiefdoms with e-democracies, we must offer **justice**—subject to democratic amendment, and a democratically elected *judiciary* that rules according to the constitution.

E-democracies will come under criminal attack through identity forgery and theft, voter coercion, misinformation, hate crimes, and other offenses. They can be redressed by the judiciary via a public warning, public condemnation, temporary gag, and fines. As suspension or, worse, expulsion, violate the basic civil right to vote, it may be considered too extreme. Imagine a future in which a person is a member of multiple e-democracies, which have a joint judicial system. A temporarily limit on participation in all these democracies simultaneously, analogous to jail time in the real world, may be severe indeed. But for such a punishment to be effective, **accountability** must be ensured: it is not sufficient that the offending digital identity be truthful; it has to be unique and persistent, lest the offender sheds the punishment by abandoning one identity in favor of another.

7. **Hysteresis:** Democracy's forefathers did not foresee the immediacy with which the general will can be ascertained on the Internet. Eventually, the general will must prevail lest we violate sovereignty. But it should go through reasonable checks and balances until it does, lest mob dynamics prevail. To this end we enlist **hysteresis**, a characteristic of systems in which the output is not an immediate function of the input.

While a multiyear election cycle confers natural hysteresis on earthly democracies, e-democracies require hysteresis to be engineered, so that swings in people's opinions may not immediately result in decisions that accommodate such swings. Examples include minimal periods for proposal making and deliberation; minimal endorsements for proposals to be considered; minimal quorum for a decision to be binding; and special majority needed for certain actions, for example, change of constitution.

Conclusion

It is my opinion that representative democracies are in dire straits because of their failure to uphold core democratic values, notably equality and transparency, and that e-democracy may offer the only feasible remedy. I have derived requirements for the foundations of e-democracy from the 1789 French *Declaration of the Rights of Man and of the Citizen*. The next urgent step is to build such foundations so the desperately needed Internet revolution of earthly democracies would commence. C

References

1. Anttiroiko, A.-V. Building strong e-democracy: The role of technology in developing democracy for the information age. *Commun. ACM 46*, 9 (Sept. 2003), 121–128.
2. Camenisch, J. and Lysyanskaya, A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *Advances in Cryptology—EUROCRYPT 2001* (2001).
3. Create a Democracy contract in Ethereum, 2017; https://www.ethereum.org/dao
4. *Déclaration des droits de l'homme et du citoyen*; National Constituent Assembly (1789); English translation: https://en.wikipedia.org/wiki/Declaration_of_the_Rights_of_Man_and_of_the_Citizen
5. Democracy Index. Wikipedia, 2017; https://en.wikipedia.org/wiki/Democracy_Index
6. Foa, R.S. and Mounk, Y. The danger of deconsolidation. *The Democratic Disconnect 27*, 3 (2016).
7. Gilad, Y., Hemo, R., Micali, S., Vlachos, G. and Zeldovich, N. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. Cryptology ePrint Archive, Report 2017/454 (2017).
8. Jeffries, D. Why Everyone Missed the Most Mind-Blowing Feature of Cryptocurrency. Hacker Noon, 2017.
9. Lee, D. T., Goel, A., Aitamurto, T. and Landemore, H. Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions. In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP-14)* (Pittsburgh, PA, 2014). AAAI Press, 2014.
10. Levine, B.N., Shield, C., and Margolin, N.B. *A Survey of Solutions to the Sybil Attack*. University of Massachusetts Amherst, Amherst, MA, 2006.
11. Lewis, A. A gentle introduction to self-sovereign identity (2017).
12. Magna Carta Libertatum, 1215; https://en.wikipedia.org/wiki/Magna_Carta
13. Mill, J.S. *On Liberty*. Longman, Roberts & Green, Bartleby.com, 1859, 1999.
14. Schuler, D. Creating the world citizen parliament: Seven challenges for interaction designers. *Interactions 20*, 3 (Mar. 2013), 38–47.
15. Shapiro, E. and Talmon, N. A Condorcet-Optimal Participatory Budgeting Algorithm. arXiv (2017).

Ehud Shapiro (ehud.shapiro@weizmann.ac.il) is the incumbent of the Harry Weinrebe Chair of Computer Science and Biology, Department of Computer Science and Applied Math, Weizmann Institute of Science, Rehovot, Israel.

The author thanks Ofir Raz, Amos On, Luca Cardelli, Jeffrey Sachs, Stan Letovsky, Yaniv Erlich, Benny Daon, Shani Gershi, Nimrod Talmon, Ariel Procaccia, Liav Orgad, and Raffaele Marchetti for discussions and helpful comments and Michal Golan-Mashiach and Ouri Poupko for their help.

DOI: 10.1145/3231050

Counterpoint: E-Democracy Won't Save Democracy. Democracy Will Save Democracy

Increased technology is not the solution to the fundamental issue of declining democratic culture.

Douglas Schuler

Democracy is radical. It exists when people are involved in their own governance: participating in public problem-solving and checking power. It entails awesome responsibilities that citizens don't always embrace. But shirking these responsibilities invites catastrophe: decisions would be made by the most powerful to enrich the few at the expense of the many and the natural environment. Also, as the trend persisted, the ability for citizens to engage wisely and effectively would degrade. More obstacles to engagement would be erected by those who make the decisions. And so on in a downward spiral.

I agree with Ehud Shapiro's statement in his "Point" column, "Foundations of e-Democracy," that democracy worldwide is threatened and degraded. Many countries are becoming less democratic and citizens around the world are losing confidence in democracy.[5] I disagree, however, with many of his prescriptions including the assertion that "e-democracy may offer the only feasible remedy." Declining democratic culture—not lack of technology—is the best indicator for declining democratic participation. When people see governance as irrelevant and unresponsive, they become cynical and withdrawn and the general ability to help address shared challenges withers. Moving the mechanics of democracy to the Internet ignores these core realities.

Saving Democracy

Shapiro observes that "many democracies transform into oligarchies, plutocracies, or even kleptocracies" because they are dominated by "the rich, the powerful, and the connected." Beyond that there are is little analysis of the problems that could help us see the benefits of his prescriptions. His support for e-democracy seemingly rests on the Internet's near-magical properties. In building a case for an "Internet revolution of democracy" he asserts "the pressing need" for it and states there exists "apparent clear ability of the Internet to deliver it." A variety of other critical questions are begged by the presumption that e-democracy is necessary—even inevitable.

The big problems we face including lack of government leadership, media freedom, and critical civic education, are problems that technology alone cannot fix. Other nagging problems such as professional dissembling, influence of money, corruption, gerrymandering, and voter suppression also share that feature and addressing them *non-technologically* could help give rise to a democracy that was amenable to intelligently integrating online opportunities.

According to Shapiro, e-democracy "presupposes universal Internet access as well as Net neutrality." This seems to imply that his prescriptions are of no use in many settings (in the U.S., for example, as well as most of the world) where those attributes do not exist and, unfortunately, may never exist. About 20% of adults in the U.S.—often the most disadvantaged citizens—have neither broadband at home nor smartphones (https://pewrsr.ch/2kQtkrM; https://pewrsr.ch/2inUJzB) and Net neutrality is threatened.[6] If those conditions must already exist (and I would propose adding "non-surveilled" Internet access) Shapiro's proposal becomes utopian, mostly irrelevant in the near term.

It was unclear to me from Shapiro's "Point" column whether it is representative democracy that is in "retreat worldwide" or whether it is the political processes practiced in the world's putative democratic societies. In other words, I was not clear whether representation itself is to be dispensed with. Nevertheless, I would still mention the seminal 1789 text *Declaration of the Rights of Man and of the Citizen*[7] upon which his "Point" column is based supports that right (**A6, A14**). Although he does not use the term "direct democracy" he endorses a trajectory that "gradually supplants traditional representative democracy by e-democracy." This objective should not be seen as obvious, nor necessarily desirable. Perhaps the citizenry will want to employ "representatives" who have governing expertise? Moreover, the goal of direct democracy may be unsound on practical grounds: How much time would the average person want to expend in a given day to consider every relevant proposal?

Thinking that democracy can be reduced to a computer problem can be a dangerous distraction.

Missing Aspects

Shapiro takes an innovative approach by using the *Declaration* as a proxy "customer" for "humanity at large" to derive requirements for future democratic systems. While the *Declaration* is surprisingly relevant and thorough, it says little about recent developments in our understanding of democracies and 21st-century realities. Although individual rights are fundamental to democracy it is only through collective efforts that non-trivial objectives are realized. Democracies need spaces (or settings) where people can assemble and procedures with which they can discuss, deliberate, and make decisions. John Dewey pointed out that the process of coming to a decision is actually more important than the decision itself. But this rich aspect of democracy is often overlooked by developers and funders. Citizens interact with formal governmental processes and within non-governmental organizations such as labor unions, nonprofits, and social movements. In the future citizens may also participate in global decision making. (And we could be experimenting more with that right now.) Improving the ability of citizens to organize into various types of collectivities could help provide a more democratic playing field. Increasing the involvement of people who are marginalized including undocumented people, people in occupied territories, rural people, refugees, prisoners, and people without access to the Internet is critical. The bottom line is that types and missions of various collectivities—as well as their social contexts—are exceedingly diverse and while the *Declaration* focuses on "universal" rights, the exercise of these rights (and the struggles for them) will take a multitude of forms. At least some of these forms will be online and the cooperation and commitment of the computer science community will be necessary.

The focus on democratic societies alone is limiting: It implies Shapiro's ideas are applicable to about half of the world's population. This is a huge number but the other half could also benefit from additional democratization. Democracy comes in shades of gray and processes that degrade or enrich democracy are perpetually at play in all countries. Hence, determining the level of e-democracy readiness is not trivial although the need to do so is essential. Moreover, the problems humankind faces are global even if the negative consequences of these problems are borne unequally. But opening up e-democracies to the people of the world would likely be problematic as governments (and media monopolies and other powerful entities) might feel inclined to nudge their citizens to vote their way.

Shapiro also makes several disconcerting technological recommendations although limited editorial space and my lack of knowledge of the technological particulars prevent an extensive analysis. Technology is embedded within social contexts that cannot be separated from the technology in use. Even democratic functions that seem most conducive to automation such as voting have not yet demonstrated the necessary legitimacy to warrant universal adoption. And the idea of conducting the necessary discussion and deliberation without surveillance and harassment seems impossibly utopian in this era of mass harvesting of personal information. Beyond that there are deep inherent risks in staking future democracies on unproven technologies including blockchain, cryptocurrencies, and smart contracts. And handing over decision making to an "autonomous, incorruptible, transparent, and persistent software agent" is essentially nondemocratic, even if it is "programmed to obey democratic decisions."

Finally, Shapiro does not consider the process of achieving e-democracy in any depth. Thinking about how we get there is crucial, non-trivial, and political—not merely technological. Improving democracy is not a matter of building a system based on a set of requirements and switching it on. Democracy requires participation and the design and development of participatory systems are best undertaken with participation. He suggests "e-parties" will "export their participatory practices of their inner workings to real-world governments" but this is a narrow view of social innovation (and our experiences with e-parties thus far have not been entirely reassuring). It is relevant to note that women in France—but not all—were only granted voting rights in 1944, a full 155 years after the 1789 *Declaration* that asserted the equality of all.

Technology is embedded within social contexts that cannot be separated from the technology in use.

Conclusion

I appreciate Shapiro's focus on foundations. My critique could be seen as providing additional foundations including political realities, critique, and provisos. I fear Shapiro's discussion on technologies goes beyond the foundation orientation into the realm of technological determinism or fetishism. Thinking that democracy can be reduced to a computer problem can be a dangerous distraction. The reality is that many of the "answers" we seek can only be determined through seeing how new systems are used, and this use is likely to vary from cultural context to cultural context.

But this critique is not intended to discourage new citizen approaches, including ones that use the new affordances the Internet provides. On the contrary, many initiatives such as participatory budgeting,[3] deliberative polling,[4] online deliberation,[2] citizen juries,[10] and many others suggest promising directions for transforming our democratic systems incrementally.

To get this right we must experiment. Our systems must evolve and this means engagement with real people. While the technological contribution is necessary, civil society, librarians, artists, government officials, activists, and "ordinary" people must also assume important roles. In an article I wrote for *ACM Interactions*,[8] I proposed a "global parliament" as a suitable grand challenge in which the community of computer professionals could collaborate with many others to design and build a system (or systems) that facilitated global citizen communication. Computer professionals need to keep in mind the broad social goals—foundations—such as strengthening social and cultural support and interest in democracy; increasing access to information and dialogue and deliberation; and giving voice to marginalized people. This means working to ensure the right mixture of people, policy, institutions, processes, education, and, of course, technology.

The media landscape at the time of the *Declaration* bears little resemblance to the ubiquitous, monopolistic digital empires of today with their global reach, massive data mining, and influence on public opinion. And governments of the 18^{th} century did not hire hackers and digital mercenaries. Thus more control over the existing media and more access to and support for publicly owned media will be necessary for genuine democracy in the 21^{st} century. I agree with Shapiro that Facebook is not an appropriate platform for this, nor could any for-profit, proprietary, closed system. A project of this magnitude requires a deep, long-term commitment by civil society, government, professional societies, and others. The first principle of the ACM with its "obligation to protect fundamental human rights" suggests it should be involved. And a project of this magnitude would require sustained support.

Good democratic governance should not be confused with "thin democracy,"[1] where citizens assume minimal roles. We need systems that help people be more engaged, better informed, and more adept at public problem solving, a capacity I refer to as civic intelligence.[9] The point is not merely to make people's lives more convenient but to make their lives richer, including their ability to contribute to the common good.

Democracy is—and always will be—a work in progress. It is by definition imperfect. At its core it is an artifact of rules and procedures animated by human beings. It is necessarily both open and closed, constrained and free. It necessarily includes non-sanctioned activities such as peaceful protest and civil disobedience. Let's use Shapiro's ideas as provocations, hypotheses, or proposals as we move forward. But if we use the ideas and approach he proposes and advocates in his "Point" column (or any single proposal) as the blueprint, we will miss the opportunity to improve the governance approaches we need for current and future realities. It is a critical time for this community to engage in deep and ongoing discussion and activism on the roles of computers—and computer professionals—in society. **C**

References

1. Barber, B. *Strong Democracy: Participatory Politics for a New Age.* University of California Press, 2003.
2. Davies, T. and Gangadharan, S.P. *Online Deliberation: Design, Research, and Practice.* University of Chicago Press, 2009.
3. Dias, N. *Hope for Democracy: 25 Years of Participatory Budgeting Worldwide.* In Loco Association, São Brás de Alportel, 2014; https://bit.ly/2Jw4hdy
4. Fishkin, J.S. and Luskin, R.C. Experimenting with a democratic ideal: Deliberative polling and public opinion. *Acta Politica 40*, 3 (Mar. 2005), 284–298.
5. Foa, R.S. and Mounk, Y. The democratic disconnect. *Journal of Democracy 27*, 3 (Mar. 20106), 5–17.
6. Kang, C. F.C.C. repeals Net neutrality rules. *The New York Times* (Dec. 14, 2017); https://nyti.ms/2CkTbRR
7. National Constituent Assembly France. *Declaration of the Rights of Man and of the Citizen* (1789); https://bit.ly/2uJrA8A
8. Schuler, D. Creating the world citizen parliament: Seven challenges for interaction designers. *ACM interactions 20*, 3 (May+June 2013).
9. Schuler, D. Cultivating society's civic intelligence: Patterns for a new 'world brain'. *Information, Communication, and Society 4*, 2 (Feb. 2001), 157–181.
10. Smith, G. and Wales, C. Citizens' juries and deliberative democracy. *Political Studies 48*, 1 (Jan. 2000), 51–65.

Douglas Schuler (douglas@publicsphereproject.org) is faculty emeritus at The Evergreen State College in Olympia, WA, USA.

practice

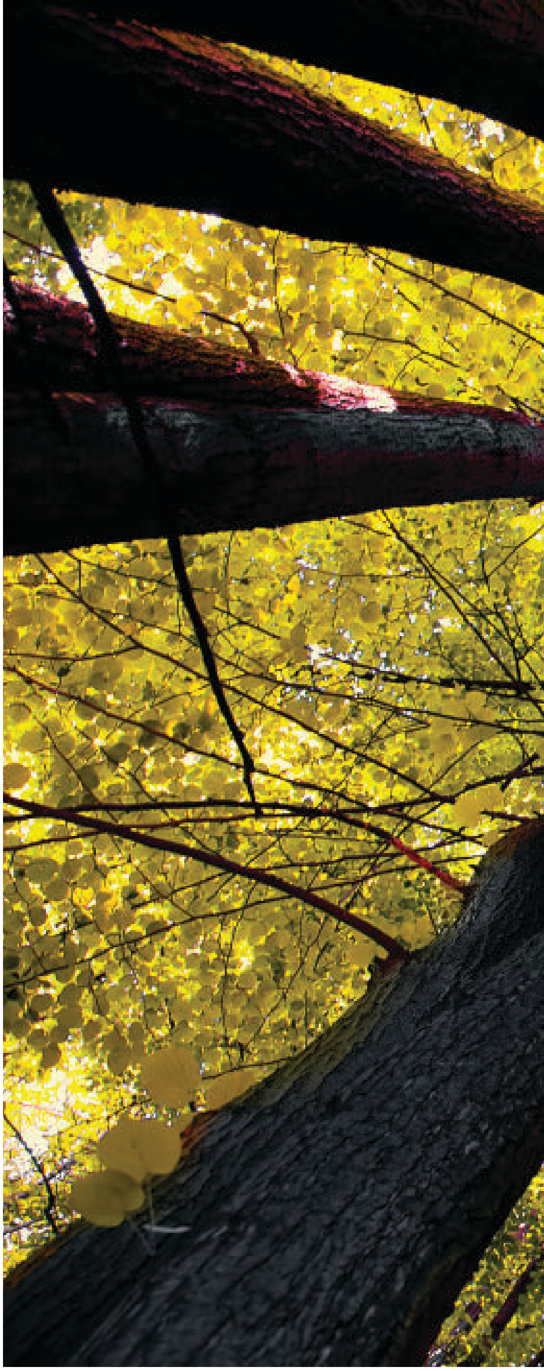
DOI:10.1145/3209210

Article development led by acmqueue
queue.acm.org

Different uses for read-optimized B-trees and write-optimized LSM-trees.

BY ALEX PETROV

Algorithms Behind Modern Storage Systems



THE AMOUNTS OF data processed by applications are constantly growing. With this growth, scaling storage becomes more challenging. Every database system has its own trade-offs. Understanding them is crucial, as it helps in selecting the right one from so many available choices.

Every application is different in terms of read/write workload balance, consistency requirements, latencies, and access patterns. Familiarizing yourself with database and storage internals facilitates architectural decisions, helps explain why a system behaves a certain way, helps troubleshoot problems when they arise, and fine-tunes the database for your workload.

It is impossible to optimize a system in all directions. In an ideal world there would be data structures guaranteeing the best read and write performance with no storage overhead but, of course, in practice that is not possible.

This article takes a closer look at two storage system design approaches used in a majority of modern databases—read-optimized B-trees[3] and write-optimized LSM (log-structured merge)-trees[8]—and describes their use cases and trade-offs.

B-Trees

B-trees are a popular read-optimized indexing data structure and general-

IMAGE BY DARIUSH M/SHUTTERSTOCK

ization of binary trees. They come in many variations and are used in many databases (including MySQL InnoDB[7] and PostgreSQL[10]) and even file systems (HFS+,[1] HTrees in ext4[6]). The B in B-tree stands for *Bayer*, the author of the original data structure, or *Boeing*, where he worked at that time.

In a binary tree every node has two children (referred as a left and a right child). Left and right subtrees hold the keys that are less than and greater than the current node key, respectively. To keep the tree depth to a minimum, a binary tree has to be balanced: when randomly ordered keys are being added to the tree, it is natural that one side of the tree will eventually get deeper than the other.

One way to rebalance a binary tree is to use so-called rotation: rearrange nodes, pushing the parent node of the longer subtree down below its child and pulling this child up, effectively placing it in its parent's position. Figure 1 is an example of rotation used for balancing in a binary tree. On the left, a binary tree is unbalanced after adding node 2 to it. In order to balance the tree, node 3 is used as a pivot (the tree is rotated around it). Then node 5, previously a root node and a parent node for 3, becomes its child node. After the rotation step is done, the height of the left subtree decreases by one and the height of the right subtree increases by one. The maximum depth of the tree has decreased.

Binary trees are most useful as in-memory data structures. Because of balancing (the need to keep the depth of all subtrees to a minimum) and low fanout (a maximum of two pointers per node), they do not work well on disk. B-trees allow for storing more than two pointers per node and work well with block devices by matching the node size to the page size (for example, 4KB). Some implementations today use larger node sizes, spanning across multiple pages in size.

B-trees have the following properties:

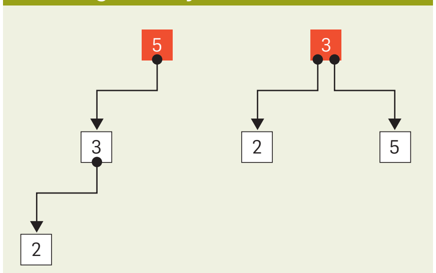
- *Sorted.* This allows sequential scans and simplifies lookups.
- *Self-balancing.* There is no need to balance the tree during insertion and

deletion: When a B-tree node is full, it is split in two, and when the occupancy of the neighboring nodes falls below a certain threshold, the nodes are merged. This also means that leaves are equally distant from the root and can be located using the same amount of steps during lookup.

▸ *Guarantee of logarithmic lookup time.* This makes B-trees a good choice for database indexes, where lookup times are important.

▸ *Mutable.* Inserts, updates, and deletes (also, subsequent splits and merges) are performed on disk in place. To make in-place updates possible, a certain amount of space overhead is required. A B-tree can be organized as a clustered index, where actual data is stored on the leaf nodes or as a heap file with an unclustered B-tree index.

Figure 1. Example of rotation used for balancing in binary tree.



This article discusses the B+tree,[5] a modern variant of the B-tree often used for database storage. The B+tree is different from the original B-tree[3] in that it has an additional level of linked leaf nodes holding the values, and these values cannot be stored on internal nodes.

Anatomy of the B-tree. Let's first take a closer look at the B-tree building blocks, illustrated in Figure 2. B-trees have several node types: root, internal, and leaf. Root (top) is the node that has no parents (that is, it is not a child of any other node). Internal nodes (middle) have both a parent and children; they connect a root node with leaf nodes. Leaf nodes (bottom) carry the data and have no children. Figure 2 depicts a B-tree with a branching factor of four (four pointers, three keys in internal nodes, and four key/value pairs on leaves).

B-trees are characterized by the following:

▸ *Branching factor:* The number (*N*) of pointers to the child nodes. Along with the pointers, root and internal nodes hold up to *N-1* keys.

Figure 2. Example of a B-tree.

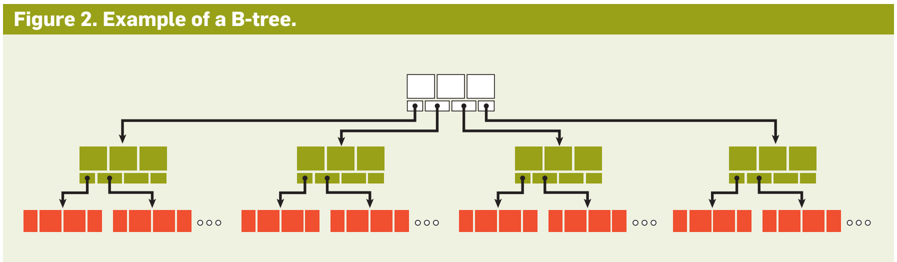


Figure 3. Single root-to-leaf pass.

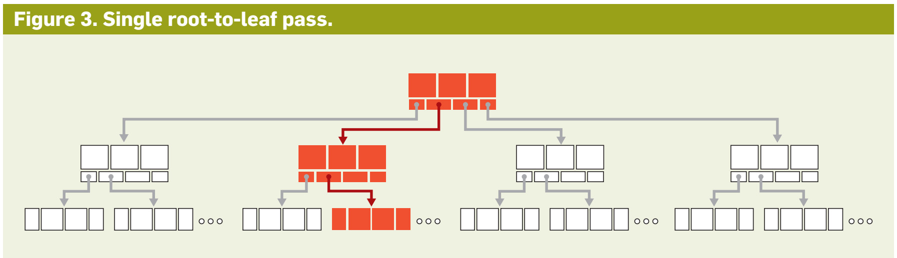
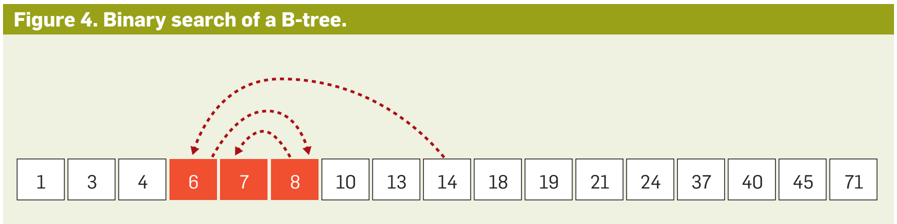


Figure 4. Binary search of a B-tree.



▸ *Occupancy:* How many pointers to child items the node is currently holding, out of the maximum available. For example, if the tree-branching factor is *N*, and the node is currently holding *N/2* pointers, occupancy is 50%.

▸ *Height:* The number of B-tree levels, signifying how many pointers have to be followed during lookup.

Every non leaf node in the tree holds up to *N-1* keys (index entries), separating the tree into *N* subtrees that can be located by following a corresponding pointer. Pointer *i* from an entry K_i points to a subtree in which all index entries are such that $K_{i-1} <= K_{searched} < K_i$ (where *K* is a set of keys). The first and last pointers are special cases, pointing to subtrees in which all the entries are less than or equal to K_0 in the case of the leftmost child, or greater than K_N in the case of the rightmost child. A leaf node may also hold a pointer to the previous and next nodes on the same level, forming a doubly linked list of sibling nodes. Keys in all the nodes are always sorted.

Lookups. When performing lookups, the search starts at the root node and follows internal nodes recursively down to the leaf level. On each level, the search space is reduced to the child subtree (the range of this subtree includes the searched value) by following the child pointer. Figure 3 shows a lookup in a B-tree making a single root-to-leaf pass, following the pointers "between" the two keys, one of which is greater than (or equal to) the searched term, and the other of which is less than the searched term. When a point query is performed, the search is complete after locating the leaf node. During the range scan, the keys and values of the found leaf, and then the sibling leaf's nodes, are traversed, until the end of the range is reached.

In terms of complexity, B-trees guarantee *log*(*n*) lookup, because finding a key within the node is performed using binary search, shown in Figure 4. Binary search is easily explained in terms of searching for words beginning with a certain letter in the dictionary, where all words are sorted alphabetically. First you open the dictionary exactly in the middle. If the searched letter is alphabetically "less than" (appears earlier than) the one opened, you continue your search in the left half of the dictionary; otherwise, you continue in the right half. You keep reducing the remaining page range by half and picking the side to follow until you find the desired letter. Every step halves the search space, making the lookup time logarithmic. Searches in B-trees have logarithmic complexity, since on the node level keys are sorted, and the binary search is performed in order to find a match. This is also why it is important to keep the occupancy high and uniform across the tree.

Insertions, updates, and deletions. When performing insertions, the first step is to locate the target leaf. For that, the aforementioned search algorithm is used. After the target leaf is located, key and value are appended to it. If the leaf does not have enough free space, this situation is called overflow, and the leaf has to be split in two. This is done by allocating a new leaf, moving half the elements to it and appending a pointer to this newly allocated leaf to the parent. If the parent does not have free space either, a split is performed on the parent level as well. The operation continues until the root is reached. When the root overflows, its contents are split between the newly allocated nodes, and the root node itself is overwritten in order to avoid relocation. This also implies the tree (and its height) always grows by splitting the root node.

LSM-trees. The log-structured merge-tree is an immutable disk-resident write-optimized data structure. It is most useful in systems where writes are more frequent than lookups that retrieve the records. LSM-trees have been getting more attention because they can eliminate random insertions, updates, and deletions.

Figure 5. Structure of an LSM tree.

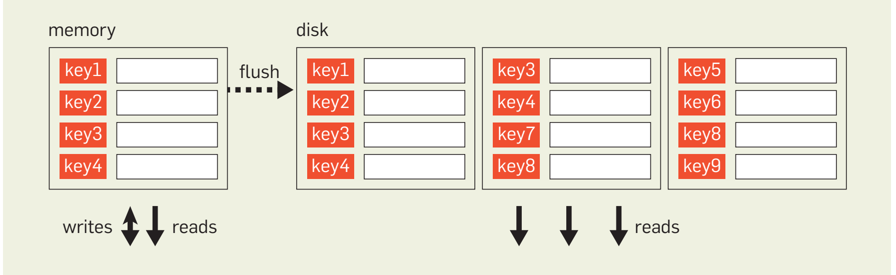


Figure 6. Structure of an SSTable.

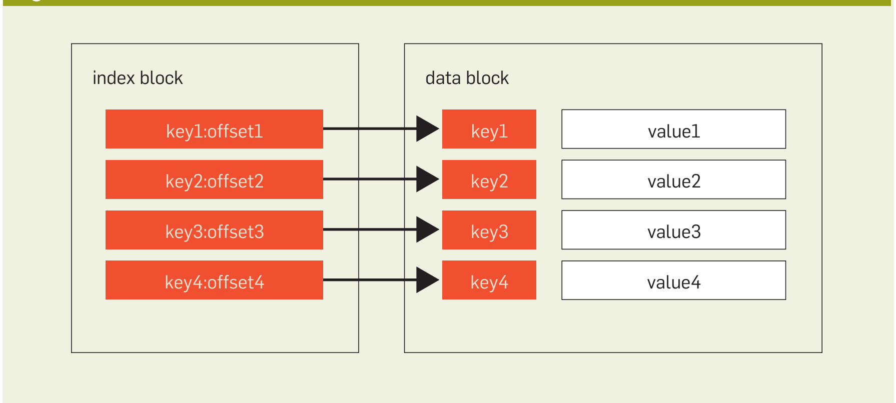
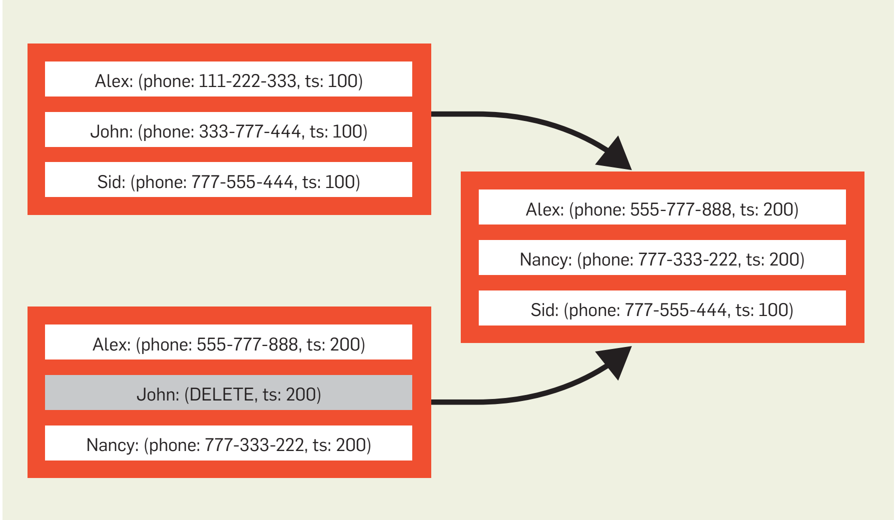


Figure 7. Example of a merger step.



Anatomy of the LSM-tree. To allow sequential writes, LSM-trees batch writes and updates in a memory-resident table (often implemented using a data structure allowing logarithmic time lookups, such as a binary search tree or skip list) until its size reaches a threshold, at which point it is written on disk (this operation is called a *flush*). Retrieving the data requires searching all disk-resident parts of the tree, checking the in-memory table, and merging their contents before returning the result. Figure 5 shows the structure of an LSM-tree: a memory-resident table used for writes, **disk-resident SSTables are used for reads**. Whenever the memory table is large enough, its sorted contents are written on disk. Reads are served, hitting both disk- and memory-resident tables, requiring a merge process to reconcile the data.

Sorted string tables. Many modern LSM-tree implementations (such as RocksDB and Apache Cassandra) implement disk-resident tables as Sorted String Tables (SSTables), because of their simplicity (easy to write, search, and read) and merge properties (during the merge, source SSTable scans and merged result writes are sequential).

An SSTable is a disk-resident ordered immutable data structure. Structurally, an SSTable is split into two parts: data and index blocks, as shown in Figure 6. A data block consists of sequentially written unique key/value pairs, ordered by key. An index block contains keys mapped to data-block pointers, pointing to where the actual record is located. An index is often implemented using a format optimized for quick searches, such as a B-tree, or using a hash table for a point-query. Every value item in an SSTable has a timestamp associated with it. This specifies the write time for inserts and updates (which are often indistinguishable) and removal time for deletes.

SSTables have some nice properties:

- Point-queries (that is, finding a value by key) can be done quickly by looking up the primary index.
- Scans (that is, iterating over all key/value pairs in a specified key range) can be done efficiently simply by reading key/value pairs sequentially from the data block.

An SSTable represents a snapshot of all database operations over a period of time, as the SSTable is created by the *flush* process from the memory-resident table that served as a buffer for operations against the database state for this period.

Lookups. Retrieving data requires searching all SSTables on disk, checking the memory-resident tables, and merging their contents before returning the result. The merge step during the read is required since the searched data can reside in multiple SSTables.

The merge step is also necessary to ensure the deletes and updates work. Deletes in an LSM-tree insert placeholders (often called *tombstones*), specifying which key was marked for deletion. Similarly, an update is just a record with a later timestamp. During the read, the records that get shadowed by deletes are skipped and not returned to the client. A similar thing happens with the updates: out of two records with the same key, only the one with the later timestamp is returned. Figure 7 shows a merge step reconciling the data stored in separate tables for the same key: as shown here, the record for Alex was written with timestamp 100 and updated with a new phone and timestamp 200; the record for John was deleted. The other two entries are taken as is, as they are not shadowed.

To reduce the number of searched SSTables and to avoid checking every SSTable for the searched key, many storage systems employ a data structure known as a Bloom Filter.[2] This is a probabilistic data structure that can be used to test whether an element is a member of the set. It can produce false-positive matches (that is, state that the element is a member of set, while it is not, in fact, present there) but cannot produce false negatives (that is, if a negative match is returned, the element is guaranteed not to be a member of the set). In other words, a Bloom Filter is used to tell if the key "might be in an SSTable" or "is definitely not in an SSTable." SSTables for which a Bloom Filter has returned a negative match are skipped during the query.

Figure 8. Compaction.

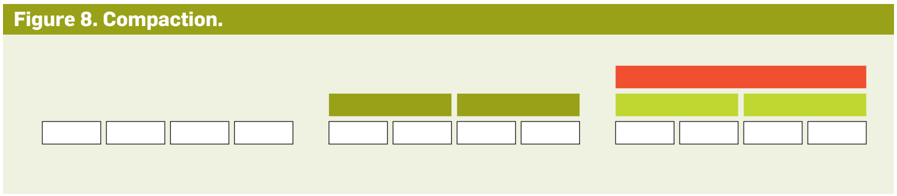
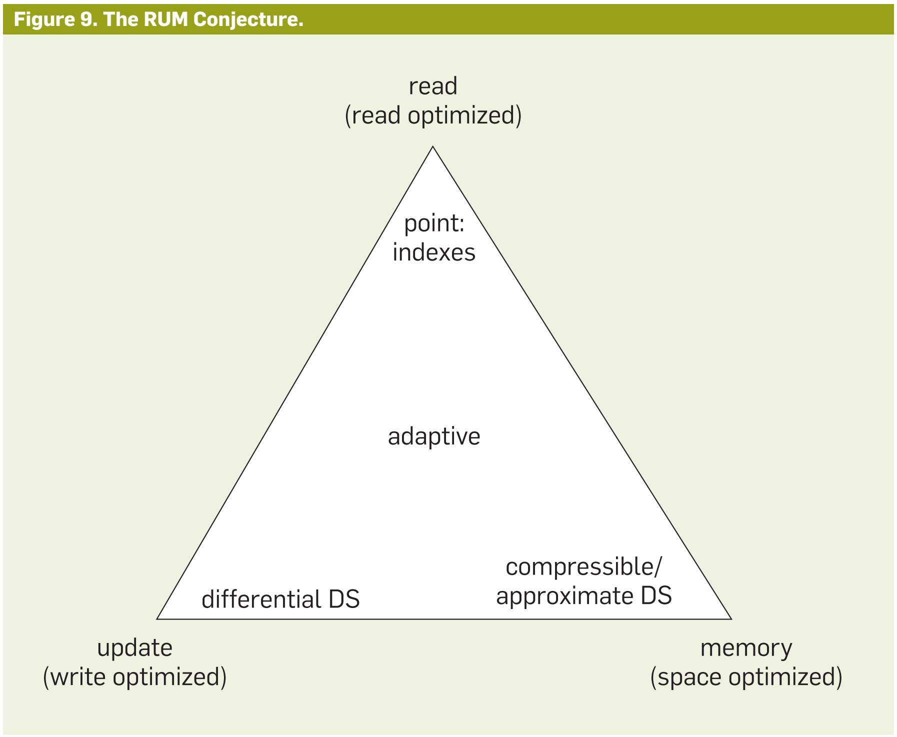


Figure 9. The RUM Conjecture.



LSM maintenance. Since SSTables are *immutable*, they are written sequentially and hold no reserved empty space for in-place modifications. This means insert, update, or delete operations would require rewriting the whole file. All operations modifying the database state are "batched" in the memory-resident table. Over time, the number of disk-resident tables will grow (data for the same key located in several files, multiple versions of the same record, redundant records that got shadowed by deletes), and the reads will continue getting more expensive.

To reduce the cost of reads, reconcile space occupied by shadowed records, and reduce the number of disk-resident tables, LSM-trees require a *compaction* process that reads complete SSTables from disk and merges them. Because SSTables are sorted by key and compaction works like merge-sort, this operation is very efficient: records are read from several sources sequentially, and merged output can be appended to the results file right away, also sequentially. One of the advantages of merge-sort is that it can work efficiently even for merging large files that don not fit in memory. The resulting table preserves the order of the original SSTables.

During this process, merged SSTables are discarded and replaced with their "compacted" versions, as shown in Figure 8. Compaction takes multiple SSTables and merges them into one. Some database systems logically group the tables of the same size to the same "level" and start the merge process whenever enough tables are on a particular level. After compaction, the number of SSTables that have to be addressed is reduced, making queries more efficient.

Atomicity and Durability

To reduce the number of I/O operations and make them sequential, both B-trees and LSM-trees batch operations in memory before making an actual update. This means data integrity is not guaranteed during failure scenarios and *atomicity* (applying a series of changes atomically, as if they were a single operation, or not applying them at all) and *durability* (ensuring that in the face of a process crash or power loss, data has reached persistent storage) properties are not ensured.

To solve that problem, most modern storage systems employ WAL (write-ahead logging). The key idea behind WAL is that all the database state modifications are first durably persisted in the append-only log on disk. If the process crashes in the middle of an operation, the log is replayed, ensuring no data is lost and all changes appear atomically.

In B-trees, using WAL can be understood as writing changes to data files only after they have been logged. Usually log sizes for B-tree storage systems are relatively small: as soon as changes are applied to the persisted storage, they can be discarded. WAL serves as a backup for the in-flight operations: any changes that were not applied to data pages can be redone from the log records.

In LSM-trees, WAL is used to persist changes that have reached the memtables but have not yet been fully flushed on disk. As soon as a memtable is fully flushed and switched so that read operations can be served from the newly created SSTable, the WAL segment holding the data for the flushed memtable can be discarded.

Summarizing

One of the biggest differences between the B-tree and LSM-tree data structures is what they optimize for and what implications these optimizations have.

One of the biggest differences between B-tree and LSM-tree data structures is what they optimize for and what implications these optimizations have.

Let's compare the properties of B-trees with LSM-trees. In summary, B-trees have the following properties:

- They are mutable, which allows for in-place updates by introducing some space overhead and a more involved write path, although it does not require complete file rewrites or multisource merges.
- They are read-optimized, meaning they do not require reading from (and subsequently merging) multiple sources, thus simplifying the read path.
- Writes might trigger a cascade of node splits, making some write operations more expensive.
- They are optimized for paged environments (block storage), where byte addressing is not possible.
- Fragmentation, caused by frequent updates, might require additional maintenance and block rewrites. B-trees, however, usually require less maintenance than LSM-tree storage.
- Concurrent access requires reader/writer isolation and involves chains of locks and latches.

LSM-trees have these properties:

- They are immutable. SSTables are written on disk once and never updated. Compaction is used to reconcile space occupied by removed items and merge same-key data from multiple data files. Merged SSTables are discarded and removed after a successful merge as part of the compaction process. Another useful property coming from immutability is that flushed tables can be accessed concurrently.
- They are write optimized, meaning that writes are buffered and flushed on disk sequentially, potentially allowing for spatial locality on the disk.
- Reads might require accessing data from multiple sources, since data for the same key, written during different times, might land in different data files. Records have to go through the merge process before being returned to the client.
- Maintenance/compaction is required, as buffered writes are flushed on disk.

Evaluating Storage Systems

Developing storage systems always presents the same challenges and factors to consider. Deciding what to optimize for has a substantial influence on the result. You can spend more time during write in order to lay out structures for more efficient reads, reserve extra space for in-place updates, facilitate faster writes, and buffer data in memory to ensure sequential write operations. It is impossible, however, to do this all at once. An ideal storage system would have the lowest read cost, lowest write cost, and no overhead. In practice, data structures compromise among multiple factors. Understanding these compromises is important.

Researchers from Harvard's DASlab (Data System Laboratory) summarized the three key parameters database systems are optimized for: read overhead, update overhead, and memory overhead, or RUM. Understanding which of these parameters are most important for your use-case influences the choice of data structures, access methods, and even suitability for certain workloads, as the algorithms are tailored having a specific use-case in mind.

The RUM Conjecture[4] states that setting an upper bound for two of the mentioned overheads also sets a lower bound for the third one. For example, B-trees are read-optimized at the cost of write overhead as well as having to reserve empty space for the (thereby resulting in memory overhead). LSM-trees have less space overhead at a cost of read overhead brought on by having to access multiple disk-resident tables during the read. These three parameters form a competing triangle, and improvement on one side may imply compromise on the other. Figure 9 illustrates the RUM Conjecture.

B-trees optimize for read performance: the index is laid out in a way that minimizes the disk accesses required to traverse the tree. Only a single index file has to be accessed to locate the data. This is achieved by keeping this index file mutable, which also increases write amplification resulting from node splits and merges, relocation, and fragmentation/imbalance-related maintenance. To amortize update costs and reduce the number of splits, B-trees reserve extra free space in nodes on all levels. This helps to postpone write amplification until the node is full. In short, B-trees trade update and memory overhead for better read performance.

LSM-trees optimize for write performance. Neither updates nor deletes require locating data on disk (which B-trees do), and they guarantee sequential writes by buffering all insert, update, and delete operations in memory-resident tables. This comes at the price of higher maintenance costs and a need for compaction (which is just a way of mitigating the ever-growing price of reads and reducing the number of disk-resident tables) and more expensive reads (as the data has to be read from multiple sources and merged). At the same time, LSM-trees eliminate memory overhead by not reserving any empty space (unlike B-tree nodes, which have an average occupancy of 70%, an overhead required for in-place updates) and allowing block compression because of the better occupancy and immutability of the end file. In short, LSM-trees trade read performance and maintenance for better write performance and lower memory overhead.

There are data structures that optimize for each desired property. Using adaptive data structures allows for better read performance at the price of higher maintenance costs. Adding metadata facilitating traversals (such as fractional cascading) will have an impact on write time and take space, but can improve the read time. Optimizing for memory efficiency using compression (for example, algorithms such as Gorilla compression,[9] delta encoding, and many others) will add some overhead for packing the data on writes and unpacking it on reads. Sometimes, you can trade functionality for efficiency. For example, heap files and hash indexes can provide great performance guarantees and smaller space overhead because of the file format simplicity, for the price of not being able to perform anything but point queries. You can also trade precision for space and efficiency by using approximate data structures, such as the Bloom Filter, HyperLogLog, Count-Min sketch, and many others.

The three tunables—read, update, and memory overheads—can help you evaluate the database and gain a deeper understanding of the workloads for which it is best suited. All of them are quite intuitive, and it is often easy to sort the storage system into one of the buckets and guess how it is going to perform, then validate your hypothesis through extensive testing.

Of course, there are other important factors to consider when evaluating a storage system, such as maintenance overhead, operational simplicity, system requirements, suitability for frequent updates and deletes, access patterns, and so on. The RUM Conjecture is just a rule of thumb that helps to develop an intuition and provide an initial direction. Understanding your workload is the first step on the way to building a scalable back end.

Some factors may vary from implementation to implementation, and even two databases that use similar storage-design principles may end up performing differently. Databases are complex systems with many moving parts and are an important and integral part of many applications. This information will help you peek under the hood of a database and, knowing the difference between the underlying data structures and their inner doings, decide what is best for you.

Related articles on queue.acm.org

The Five-Minute Rule: 20 Years Later and How Flash Memory Changes the Rules
Goetz Graefe
https://queue.acm.org/detail.cfm?id=1413264

Disambiguating Databases
Rick Richardson
https://queue.acm.org/detail.cfm?id=2696453

You're Doing It Wrong
Poul-Henning Kamp
https://queue.acm.org/detail.cfm?id=1814327

References

1. Apple HFS Plus Volume Format; https://developer.apple.com/legacy/library/technotes/tn/tn1150.html#BTrees
2. Bloom, B.H. Space/time trade-offs in hash coding with allowable errors, *Commun. ACM 13*, 7 (1970), 422–426
3. Comer, D. The ubiquitous B-tree. *Computing Surveys 11*, 2 (1979); 121–137; https://bit.ly/2w2Ms01
4. Data Systems Laboratory at Harvard. The RUM Conjecture; http://daslab.seas.harvard.edu/rum-conjecture/.
5. Graefe, G. Modern B-tree techniques. *Foundations and Trends in Databases 3*, 4 (2011), 203–402; https://bit.ly/2IbTB36
6. Mathur, A., Cao, M., Bhattacharya, S., Dilger, A., Tomas, A. and Vivier, L. The new ext4 filesystem: Current status and future plans. In *Proceedings of the Linux Symposium*. Ottawa, Canada, 2007; https://bit.ly/2HMJiyW
7. MySQL 5.7 Reference Manual. The physical structure of an InnoDB index; https://dev.mysql.com/doc/refman/5.7/en/innodb-physical-structure.html.
8. O'Neil, P., Cheng, E., Gawlick, D. and O'Neil, E. The log-structured merge-tree. *Acta Informatica 33*, 4 (1996), 351–385; https://bit.ly/2HJ2UYO
9. Pelkonen, T., Franklin, S., Teller, J., Cavallaro, P., Huang, Q., Meza, J. and Veeraraghavan, K. Gorilla: A fast, scalable, in-memory time series database. In *Proceedings of the VLDB Endowment 8*, 12 (2015): 1816–1827; http://www.vldb.org/pvldb/vol8/p1816-teller.pdf.
10. Suzuki, H. The internals of PostgreSQL, 2015–2018; http://www.interdb.jp/pg/pgsql01.html.

Alex Petrov (http://coffeenco.de/, @ifesdjeen) is an Apache Cassandra committer and storage systems enthusiast. He has worked on databases, building distributed systems and data-processing pipelines for various companies.

DOI:10.1145/3190574

Article development led by acmqueue
queue.acm.org

What happens when we wish to actually deploy a machine learning model to production?

BY DAN CRANKSHAW AND JOSEPH GONZALEZ

Research for Practice: Prediction-Serving Systems

This installment of Research for Practice features a curated selection from Dan Crankshaw and Joey Gonzalez, who provide an overview of machine learning serving systems. What happens when we wish to actually deploy a machine learning model to production, and how do we serve predictions with high accuracy and high computational efficiency? Dan and Joey's picks provide a thoughtful selection of cutting-edge techniques spanning database-level integration, video processing, and prediction middleware. Given the explosion of interest in machine learning and its increasing impact on seemingly every application vertical, it is possible that systems such as these will become as commonplace as relational databases are today. Enjoy your read!

—*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.

Machine learning is an enabling technology that transforms data into solutions by extracting patterns that generalize to new data. Much of machine learning can be reduced to learning a model—a function that maps an input (for example, a photo) to a prediction (for example, objects in the photo). Once trained, these models can be used to make predictions on new inputs (for example, new photos) and as part of more complex decisions (for example, whether to promote a photo). While thousands of papers are published each year on how to design and train models, there is surprisingly less research on how to manage and deploy such models once they are trained. It is this later, often overlooked, topic that this article addresses.

Before examining the recent work on how to manage and deploy machine-learning models, let's first briefly review the three phases of machine-learning application development: model development, training, and inference.

Model development typically begins with collecting and preparing training data. This data is then used to design new feature transformations and choose from a wide range of model designs (for example, logistic regression, random forest, or convolutional neural network) and their corresponding training algorithms. Even after a model and training algorithm are selected, there are often additional hyperparameters (for example, smoothing parameters) that must be tuned by repeatedly training and evaluating the model.

The result of model development is typically a training pipeline that can be run at scale. The training phase executes the training pipeline repeatedly as new data arrives to produce new trained models that can be used to render predictions as part of some application or service.

The final phase of rendering predictions is often referred to as prediction serving, model scoring, or inference. Prediction serving requires integrating machine-learning software with other systems including user-facing application code, live databases, and high-volume data streams. As such, it comes with its own set of challenges and trade-offs and is the domain of the emerging class of prediction-serving systems.

While prediction serving has been studied extensively in domains such as ad targeting and content recommendation, because of the domain-specific requirements these systems have developed highly specialized solutions without addressing the full set of systems challenges critical to developing high-value machine-learning applications. Here we have selected four complementary papers, each of which provides practical lessons for developing machine-learning applications, whether you are developing your own prediction-serving system or using off-the-shelf software.

Putting Models in the Database

A. Deshpande and S. Madden
MauveDB: Supporting model-based user views in database systems In *Proceedings of SIGMOD '06*; https://dl.acm.org/citation.cfm?id=1142483.

MauveDB is an ambitious effort to incorporate machine-learning models into a traditional relational database while preserving the declarativity of SQL-based query languages. MauveDB (for *m*odel-b*a*sed *u*ser *v*i*e*ws, in an homage to a well-known Dilbert cartoon; http://dilbert.com/strip/1995-11-17) starts with the observation that the modeling process is fundamentally rooted in data, yet traditional database management systems provide little value for those seeking to create and manage models. The extent of database support for models at the time the paper was written was the ability to use a trained model as a UDF (user-defined function). This allows users to bring the model to the data but is insufficient for integrating the model into a query optimizer or enabling the database to maintain the model automatically.

MauveDB observes that a model is just a way of specifying a complex materialized view over the underlying training data. The SQL view mechanism was extended to support declaratively specifying models as views that the database engine can understand and optimize. As a result, the database can automatically train and maintain models over time as the underlying data evolves. Furthermore, by integrating the models as views instead of user-defined functions, the query optimizer can use existing cost-based optimization techniques to choose the most efficient method for querying each trained model.

This deep integration of models into the database, however, has some significant limitations. In particular, MauveDB is focused on modeling sensor data and thus considers only two types of models—regression and interpolation—that are widely used in that context. Even for these two relatively simple models, the view definitions become complex to account for all of the available modeling choices. Declaratively specifying models also restricts the user to using only existing database functionality. Any custom preprocessing operations or model specialization must be written as UDFs, defeating the purpose of the tight integration between model and database. Finally, the various access methods and materialization strategies for the optimizer to choose from must be studied and developed separately for each training algorithm. As a result, the addition of new types of model-based views requires developing new access methods and incremental maintenance strategies, as well as modification to the database engine itself—tasks that ordinary users are typically neither willing nor able to do without significant effort.

The key insight in this paper is that by finding and exposing the semantics of your model to the applications in which they are embedded, you can make your end-to-end machine-learning applications both faster and easier to maintain. But this tight integration comes at the cost of generality and extensibility by making it much harder to change the modeling process or apply these techniques to new domains.

Prediction Serving at Scale

D. Agarwal, B. Long, J. Traupman, D. Xin, and L. Zhang
LASER: A scalable response prediction platform for online advertising. In *Proceedings of WSDM '14*; https://dl.acm.org/citation.cfm?id=2556195.2556252

The LASER system developed at LinkedIn explores a holistic approach to building a general platform for both training and serving machine-learning models. LASER was designed to power the company's social-network-based advertising system but found wide use within the company. The LASER team deliberately restricted the scope of models that it supports—generalized linear models with logistic regression—but took an end-to-end approach to building a system to support these models throughout the entire machine-learning life cycle. As a consequence, this paper has many insights that can be applied broadly when developing new machine-learning applications. By restricting the classes of models supported, the authors are able to build all of the techniques they discuss directly into the platform itself. But these same ideas (for example, those around caching or lazy evaluation) could be applied on a per-application basis on top of a more general-purpose serving system as well. The paper describes ideas for improving training speed, serving performance, and usability.

LASER uses a variety of techniques for intelligent caching and materialization in order to provide real-time inference (these are similar to the view-maintenance strategies discussed in §3.3.2 of the MauveDB paper). The models described in LASER predict a score for displaying a particular ad to a particular user. As such, the model includes linear terms that depend only on the ad or user, as well as a quadratic term that depends on both the user and the ad. LASER exploits this model structure to partially prematerialize and cache results in ways that maximize cache reuse and minimize wasted computation and storage. The quadratic term is expensive to compute in real time but precomputing the full cross product matching users to ads (a technique described in the literature as full prematerialization) would be wasteful and expensive, especially in a setting such as online advertising when user preferences can change quickly, and ad campaigns frequently start and stop.

Instead, the paper describes how LASER leverages the specific structure of its generalized linear models to prematerialize part of the cross product to accelerate inference without incurring the waste of precomputing the entire product. LASER also maintains a partial results cache for each user and ad campaign. This factorized cache design is particularly well suited for advertising settings in which many ad campaigns are run on each user. Caching the user-specific terms amortizes the computation cost across the many ad predictions, resulting in an overall speedup for inference with minimal storage overhead. The partial prematerialization and caching strategies deployed in LASER could be applied to a much broader class of models (for example, neural features or word embeddings).

LASER also uses two techniques that trade off short-term prediction accuracy for long-term benefits. First, LASER does online exploration using Thompson sampling to explore ads with high variance in their expected values because of small sample sizes. Thompson sampling is one of a family of exploration techniques that systematically trade off exploiting current knowledge (for example, serving a known good ad) and exploring unknown parts of the decision space (serving a high-variance ad) to maximize long-term utility.

Second, the LASER team adopted a philosophy it calls "*Better wrong than late*." If a term in the model takes too long to be computed (for example, because it is fetching data from a remote data store), the model will simply fill in the unbiased estimate for the value and return a prediction with degraded accuracy rather than blocking until the term can be computed. In the case of a user-facing application, any revenue gained by a slightly more accurate prediction is likely to be outweighed by the loss in engagement caused by a Web page taking too long to load.

There are two key takeaways from the LASER paper: First, trained models often perform computation whose structure can be analyzed and exploited to improve inference performance or reduce cost; second, it is critical to evaluate deployment decisions for machine-learning models in the context of how the predictions will be used rather than blindly trying to maximize performance on a validation dataset.

Applying Cost-Based Query Optimization to Deep Learning

D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia

NoScope: Optimizing neural network queries over video at scale. In *Proceedings of the VLDB Endowment 10*, 11 (2017); https://dl.acm.org/citation.cfm?id=3137664.

This paper from Kang et al. at Stanford presents a set of techniques for significantly reducing the *cost* of prediction serving for object detection in video streams. The work is motivated by current hardware trends—in particular, that the cost of video data acquisition is dropping as cameras get cheaper, while state-of-the-art computer vision models require expensive hardware accelerators such as GPUs

The LASER team deliberately restricted the scope of models that it supports—generalized linear models with logistic regression—but took an end-to-end approach to building a system to support these models throughout the entire machine-learning life cycle.

To support the uniform prediction interface, Clipper adopts a modular, layered architecture, running each model in a separate Docker container and interposing an intermediate layer between the models and the querying applications.

to compute predictions in realtime for a single video stream.

To reduce this cost imbalance, the authors developed a system called NoScope (https://github.com/stanford-futuredata/noscope) to reduce the monetary cost of processing videos by improving model-inference performance. The authors developed a set of techniques to reduce the number of frames on which a costly deep-learning model must be evaluated when querying a video stream, and then developed a cost-based query optimizer that selects which of these techniques to use on a per-query basis. (Note that in the NoScope work, the use of the term *query* refers to a streaming query to identify the periods of time in which a particular object is visible in the video.) As a result, while NoScope is restricted to the domain of binary classification on fixed-location cameras, it can automatically select a cost-optimal query plan for many models and applications within that domain.

The paper presents two techniques used in combination to reduce the number of frames that require a state-of-the-art model for accurate classification. First, the authors use historical video data for the *specific camera feed* being queried to train a much smaller, specialized model for the query. While this model forgoes the generality of the more expensive model, it can often classify frames accurately with high confidence. The authors use the more expensive model only if the specialized model returns a prediction below a specific confidence threshold. This approach is similar to prior work on model cascades, first introduced by Viola and Jones (https://bit.ly/2KteogS). It also bears some similarities to work on model distillation by Hinton, Vinyals, and Dean (https://arxiv.org/abs/1503.02531), although in the case of distillation the goal is to train a cheaper model to *replace* the more expensive one rather than supplement it.

NoScope combines these specialized models with a technique called *difference detectors*, which exploit the temporal locality present in fixed-angle video streaming to skip frames altogether. If the difference detectors find that the current frame is similar enough to an existing frame that has already been labeled, NoScope skips inference completely and simply uses the label from the previously classified frame. NoScope uses a cost-based optimizer to select the optimal deployment for a particular video stream, query, and model from the set of possible specialized model architectures and difference detectors.

NoScope's key insight is the identification of a domain-specific structure that can be exploited to accelerate inference in a range of settings within that domain. While the specific structure NoScope leverages is limited to fixed-location object detection, identifying temporal and spatial redundancy to reduce the load on expensive state-of-the-art models has the potential to be exploited in many different prediction-serving settings.

A General-Purpose Prediction-Serving System

D. Crankshaw, X. Wang, G. Zhou, M.J. Franklin, J.E. Gonzalez, and I. Stoica

Clipper: A low-latency online prediction-serving system. In *Proceedings of NSDI'17*; https://dl.acm.org/citation.cfm?id=3154630.3154681.

The last paper included here describes the Clipper (http://clipper.ai/) prediction-serving system. Rather than making any assumptions or restrictions on the types of models that can be served as the previous papers did, Clipper starts with the design goal of easily serving any trained model at interactive latencies. From this starting point the paper explores techniques for optimizing both inference performance and accuracy, while encapsulating the models in a uniform, black-box prediction interface.

To support the uniform prediction interface, Clipper adopts a modular, layered architecture, running each model in a separate Docker container and interposing an intermediate layer between the models and the querying applications. This distributed architecture enables the system to serve models with conflicting software and hardware requirements at the same time (for example, serving models written in different programming languages running on a mix of CPUs and GPUs). Furthermore, the architecture provides process isolation between different models and ensures a single model failure does not affect the availability of the rest of the system. Finally, this disaggregated design provides a convenient mechanism for horizontally and independently scaling each model via replication to increase throughput.

Clipper also introduces *latency-aware batching* to leverage hardware-accelerated inference. Batching prediction requests can significantly improve performance. Batching helps amortize the cost of system overheads (for example, remote procedure call and feature method invocation) and improves throughput by enabling models to leverage internal parallelism. For example, many machine-learning frameworks are optimized for batch-oriented model training and therefore capable of using SIMD (single instruction, multiple data) instructions and GPU accelerators to improve computation on large input batches. While batching increases throughput, however, it also increases inference latency because the entire batch must be completed before a single prediction is returned. Clipper employs a latency-aware batching mechanism that automatically sets the optimal batch size on a per-model basis in order to maximize throughput, while still meeting latency constraints in the form of user-specified service-level objectives.

To improve prediction accuracy, Clipper introduces a set of *selection policies* that enable the prediction-serving system to adapt to feedback and perform online learning on top of black-box models. The selection policy uses reward feedback to choose between and even combine multiple candidate models for a given prediction request. By selecting the optimal model or set of models to use on a per-query basis, Clipper makes machine-learning applications more robust to dynamic environments and allows applications to react in real time to degrading or failing models. The selection policy interface is designed to support ensemble methods (https://bit.ly/2a7aB8N) and explore/exploit techniques that can express a wide range of such methods, including multiarmed bandit techniques and the Thompson sampling algorithm used by LASER.

There are two key takeaways from this paper: the first is the introduction of a modular prediction-serving architecture capable of serving models trained in any machine-learning framework and providing the ability to scale each model independently; the second is the exploitation of the *computational structure* of inference (as opposed to the mathematical structure that several of the previous papers exploit) to improve performance. Clipper exploits this structure through batching, but there is potential for exploiting other kinds of structures, particularly in approaches that take more of a gray- or white-box approach to model serving and thus have more fine-grained performance information.

Emerging Systems and Technologies

Machine learning in general, and prediction serving in particular, are exciting and fast-moving fields. Along with the research described in this article, commercial systems are actively being developed for low-latency prediction serving. TensorFlow Serving (https://www.tensorflow.org/serving/) is a prediction-serving system developed by Google to serve models trained in TensorFlow. The Microsoft Custom Decision Service (https://bit.ly/2JHp1v2), with accompanying paper (https://arxiv.org/abs/1606.03966), provides a cloud-based service for optimizing decisions using multiarmed bandit algorithms and reinforcement learning, with the same kinds of explore/exploit algorithms as the Thompson sampling used in LASER or the selection policies of Clipper. Finally, Nvidia's TensorRT (https://developer.nvidia.com/tensorrt) is a deep-learning optimizer and runtime for accelerating deep-learning inference on Nvidia GPUs.

While the focus of this article is on systems for prediction serving, there have also been exciting developments around new hardware for machine learning. Google has now created two versions of its TPU (Tensor Processing Unit) custom ASIC. The first version, announced in 2016, was developed specifically to increase the speed and decrease the power consumption of its deep-learning inference workloads. The TPUv2, announced in 2017, supports both training and inference workloads and is available as part of Google's cloud offering. Project Brainwave (https://bit.ly/2iotXMQ) from Microsoft Research is exploring the use of FPGAs (field-programmable gate arrays) to perform hardware-based prediction serving and has already achieved some exciting results demonstrating simultaneously high-throughput and low-latency deep-learning inference on a variety of model architectures. Finally, both Intel's Nervana chips and and Nvidia's Volta GPUs are new, machine learning-focused architectures for improving the performance and efficiency of machine-learning workloads at both training and inference time.

As machine learning matures from an academic discipline to a widely deployed engineering discipline, we anticipate that the focus will shift from model development to prediction serving. As a consequence, we are anxious to see how the next generation of machine-learning systems can build on the ideas pioneered in these papers to drive further advances in prediction-serving systems.

Dan Crankshaw is a Ph.D. student in the UC Berkeley CS department working in the RISELab. His current research interests include systems and techniques for serving and deploying machine learning, with a particular emphasis on low-latency and interactive applications.

Joseph Gonzalez is an assistant professor at UC Berkeley and co-director of the UC Berkeley RISELab where he studies the design of algorithms, abstractions, and systems for scalable machine learning. Before joining UC Berkeley, he co-founded Turi Inc. (formerly GraphLab) to develop AI tools for data scientists and later sold Turi to Apple. He also developed the GraphX framework in Apache Spark.

DOI:10.1145/3209277

Article development led by acmqueue
queue.acm.org

For many data items, the work never settles on a value.

BY PAT HELLAND

Consistently Eventual

IN RECENT YEARS, there has been a lot of excitement over eventual consistency.[6] Heck, I get pretty excited about it! Eventual consistency is an aspect of some data that says its underlying value is unknown until work on that item settles down. It turns out that, in many cases, there are data items for which the work never settles down. In addition to being eventually consistent, many data items remain consistently eventual!

Eventual consistency occurs when the value for something is replicated in more than one place, and there is a protocol for these replicas converging. Changes to one or more of the replicas can be done independently, and they will propagate and converge.

Eventual consistency:

When we all know the same stuff, we will have the same result.

In this article, I do not want to talk about how eventual consistency can be accomplished but more about what it looks like when it's used. Many fun papers have been written about eventual consistency. One of my favorites is "Eventual Consistency Today: Limitations, Extensions, and Beyond," by Peter Bailis and Ali Ghodsi.[1]

As already mentioned, eventual consistency is typically used to describe the behavior of a data item that is replicated over decoupled systems. When updates happen to disconnected replicas separately, how do they behave when they reconnect and share their state?

Still, eventual consistency typically refers to *the behavior of a single replicated object*. It does not usually speak to transactions and what they mean with eventually consistent objects.

Data Floating Loose in the Mean, Cruel World

Data in a relational database behaves differently from data kept outside of one. When nonrelational data is unlocked, it gets captured as a message, file, key value, or something else grouped as a lump. These lumps (or objects, values, or entities) have an identity and version.[2]

Eventual consistency arises when an object with multiple replicas, *each with the same identity*, somehow coalesces to a common value—*even when the different replicas are updated independently*. This inherently means the version(s) of the object are not linearly assigned. It no longer makes sense to talk about a strict ordering of the changes. You must be prepared to capture the version of the object in a fashion that represents independent changes coming together. An excellent versioning mechanism is the vector clock.

It is important to recognize that the entire eventual consistency discussion must necessarily work in a world with objects, identities, and versions. It's not really a classic database thing.

Wait ... What Does a Transaction Mean?

"Last Writer Wins" is a form of eventual consistency. Consider a system that captures the wall-clock time from the local system whenever it updates a replica. When everyone has heard all the updates, the one written

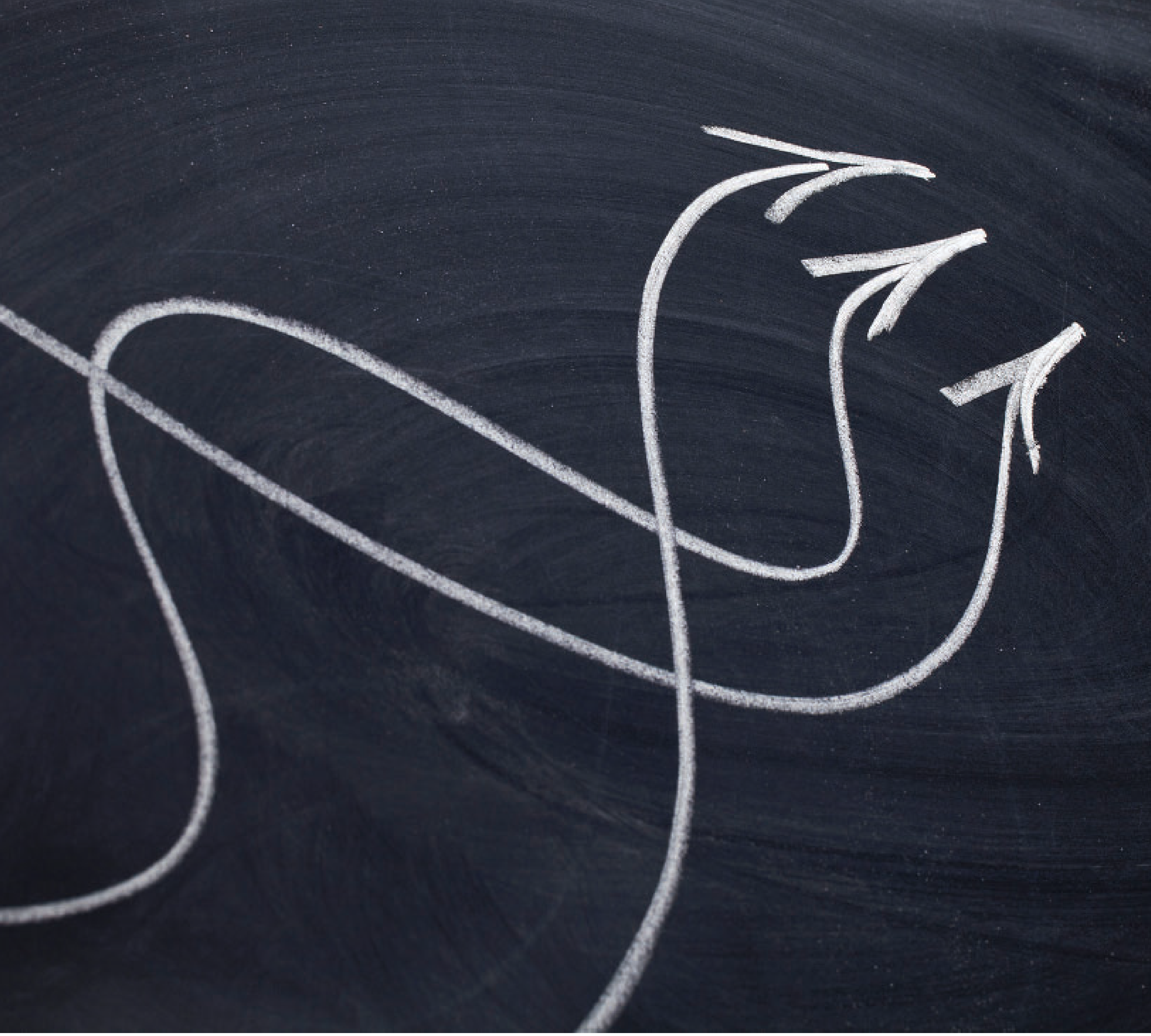


IMAGE BY MAREK ULIASZ

with the latest wall-clock time is kept everywhere. This is challenging for transactional updates. Sometimes, a change within a transaction has the latest time and is kept. Sometimes, the transactional change is stomped out by a later update. This makes atomicity a challenge.

A change can also be captured as a commutative operation.[5] This is evident in banking when a debit or credit is applied to your account and these operations can be reordered or commuted.

When you write a check on your joint checking account and your spouse writes a check at the same time, hopefully they will both clear. Given enough money in the bank, it doesn't matter whose check clears first. A transaction can deposit money into one account via a check drawn on another. The check will usually clear, making a valid transaction. Sometimes, a bounced check will form another transaction to compensate for the first transaction with the bum check.

When dealing with all of these issues, the best you can hope for is a probabilistic success combined with eventual compensation.[3] While we strive for perfect transactional work in banking, we end up compensating when stuff goes wrong. Unlike some other areas of human endeavor, for the most part we can compensate for financial errors.

When Is Eventual? Is It Now?

One problem with having replicas is that you really never know when one of your evil twins will pop back into existence. Sometimes, algorithms codify that a replica is *persona non grata* after a certain period of time. Sometimes, you overlook that and a zombie replica will come back when least expected.

My wife and I have a checking account that is perennially in a state of debits and credits. When we both use it, no one really knows how much money is flying and floating.

Our personal checking account is consistently eventual.

The only way to figure out the balance is to stop using the account for a while.

There is also the pesky problem of the check written to someone who doesn't deposit it in a reasonable amount of time. Perhaps it was left in a wallet and deposited a year later. If the check is not deposited for months, do you put a stop order on it or just wait and assume it's not coming through? The balance in your checking account is annoyingly eventual!

Snapping Uncertainty into "What We Know So Far"

Our bank sends us monthly statements. They represent the debits and credits to our account that cleared a strongly consistent location as of a deadline. That strongly consistent location is the bank's centralized computer system. The debits and credits that have arrived at the clearinghouse by the monthly deadline get scooped up into the account's statement.

Quarterly reports for public companies take a similar but different approach. At midnight of the last day of the quarter, new business and new expenses start being allotted to the next quarter. The company begins gathering and organizing all the income and expenses from the now-closed quarter. Then all records of what was spent and earned are swept into a big mélange that results in a public quarterly profit and loss report. This usually happens within 40 days or so after the quarter closes.

Some transactions during the quarter, however, may not be sent to the accountants in a timely fashion. One contributing factor may be the employment of software engineers, who are notoriously bad at the punctual submission of expense reports. So, the results published for the quarter are approximately correct but not perfect.

After the quarterly report, corrections will dribble in to the accountants. They will either categorize them as minor and issue a slight correction to the numbers for the previous quarter or issue a restatement. For a public company, a restatement showing a noticeable difference from the published report is embarrassing and rarely happens. Minor corrections are common.

You can't really know what happened until you have heard everything.

The longer you wait, the more you hear and the more accurate your opinion. Eventually, you give up waiting for new information and declare your opinion of what happened a few months ago is accurate enough.

In the bank account statement, the definition of certainty is provided at the bank's centralized computer system at end-of-day when the month closes. Uncertainty *from the perspective of the bank* is eliminated. In the corporate quarterly reports, uncertainty is gauged by how much of the underlying truth of the business filters its way back to the accountants. The quarterly report is not definitive, just pretty darned close—at least usually.

Trust, Timeouts, and Escalation

Working across trust boundaries is always eventual. Because you may not trust another entity, you are not going to do a distributed two-phase commit[4] and lock up your database waiting for that other company. Instead, you have a workflow in which partial trust is used to get your cooperative business done. Throughout this process, there are long windows in which you are just waiting and waiting … still more examples of being consistently eventual.

This eventual nature of uncertainty continues through the steps of the workflow. You agree to reserve 200 widgets from your inventory on the receipt of a deposit from your purchaser. If you hear nothing back to consummate the purchase of the widgets, you are stuck. While your disappointment is somewhat tempered by the deposit you keep, it's not enough to pay for the widgets. Darn! You lost out on selling them to another customer!

Working across trust boundaries, cooperative work functions using timeouts and escalation. If you do not call to cancel your room reservation at a hotel with 72 hours advance notice, you are stuck with the first night's room charge. The hotel, however, is likely stuck with the remaining six days of an empty room from your one-week reservation.

Indeed, the hotel has an ongoing parade of eventually resolved room sales. By the time it knows what is happening on Tuesday, the confusion of Wednesday's occupancy is about to be resolved. Again, payments to the hotel are consistently eventual.

Conclusion

Applications are no longer islands. Not only do they frequently run distributed and replicated over many cloud-based computers, but they also run over many handheld computers. This makes it challenging to talk about a single truth at a single place or time. In addition, most modern applications interact with other applications. These interactions settle out to impact understanding. Over time, a shared opinion emerges just as new interactions add increasing uncertainty. Many business, personal, and computational "facts" are, in fact, uncertain. As some changes settle, others meander from place to place.

With all the regular, irregular, and uncleared checks, my understanding of our personal joint checking account is a bit hazy. While I try to convince myself I will someday understand it, I have reconciled myself that it's really consistently eventual. C

Related articles on queue.acm.org

Don't Settle for Eventual Consistency
Wyatt Lloyd et al.
https://queue.acm.org/detail.cfm?id=2610533

Eventually Consistent: Not What You Were Expecting?
Wojciech Golab et al.
https://queue.acm.org/detail.cfm?id=2582994

Scalable SQL
Michael Rys
https://queue.acm.org/detail.cfm?id=1971597

References

1. Bailis, P. and Ghodsi, A. Eventual consistency today: limitations, extensions, and beyond. *acmqueue 11*, 3 (2013); https://queue.acm.org/detail.cfm?id=2462076.
2. Helland, P. Data on the outside versus data on the inside. In *Proceedings of the Conference on Innovative Database Research*, 2005.
3. Helland, P. and Campbell, D. Building on quicksand. In *Proceedings of the Conference on Innovative Database Research*, 2009.
4. Mohan, C. and Lindsay, B. Efficient commit protocols for the tree of processes model of distributed transactions. *ACM SIGOPS Operating Systems Review 19*, 2 (1985), 40–52; https://dl.acm.org/citation.cfm?id=850772.
5. Shapiro, M., Preguiça, N., Baquero, C. and Zawirski, M. Conflict-free replicated data types. In *Proceedings of the 13th International Conference on Stabilization*, Safety, and the Security of Distributed Systems, 2011, Springer-Verlag, Berlin, 386–400.
6. Vogels, W. Eventually consistent. *Commun. ACM 52*, 1 (Jan. 2009), 40–44; https://dl.acm.org/citation.cfm?id=1435432.

Pat Helland has been implementing transaction systems, databases, application platforms, distributed systems, fault-tolerant systems, and messaging systems since since 1978. He currently works at Salesforce.

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
Insup Lee, SIG Governing Board Awards Committee Liaison
John R. White, ACM Awards Committee Co-Chair
Rosemary McGuinness, ACM Awards Committee Liaison

DOI:10.1145/3154485

Science fiction in particular offers students a way to cultivate their capacity for moral imagination.

BY EMANUELLE BURTON, JUDY GOLDSMITH, AND NICHOLAS MATTEI

How to Teach Computer Ethics through Science Fiction

COMPUTER SCIENCE FACULTY have a responsibility to teach students to recognize both the larger ethical issues and particular responsibilities that are part and parcel of their work as technologists. This is, however, a kind of teaching for which most of us have not been trained, and that faculty and students approach with some trepidation. In this article, we explore the use of science fiction as a tool to enable those teaching artificial intelligence to engage students and practitioners about the scope and implications of current and future work in computer science. We have spent several years developing a creative approach to teaching computer ethics, through a course we call "Science Fiction and Computer Ethics."[7–9,18,28] The course has been taught five times at the University of Kentucky and two times at the University of Illinois at Chicago and has been successful with students, as evidenced by increasing and full enrollments; high teaching-evaluation numbers; positive anonymous comments from students; nominations and awards for good teaching; and invitations to speak about the course on conference panels and in talks.

Computer science, as a field, already recognizes that some ethics education is essential; the Accreditation Board for Engineering and Technology (http://www.abet.org/), one of the largest U.S.-based accreditors of engineering and technology programs, requires instruction on professional ethics. Indeed, some in computer science have gone so far as to require students in undergraduate courses to perform ethics consultations for local industry.[24] However, educating students to engage ethical challenges is often left to the cross-disciplinary portions of university curricula, especially in the U.S.[12] We, as well as others, argue that spending time focused on how these issues apply to both our own research and our students' future work is important and necessary within computer science.[30,36]

In fields with a strong practical component and established body of

» key insights

- It is important to teach students to understand the difference between normative ethics—or what is the right answer or normal mode of thought—and descriptive ethics—or using the language of ethical theory to understand and describe a situation.
- Using fiction to teach ethics allows students to safely discuss and reason about difficult and emotionally charged issues without making the discussion personal.
- A good technology ethics course teaches students how to think, not what to think, about their role in the development and deployment of technology, as no one can foresee the problems that will be faced in a future career.

knowledge (such as medicine, engineering, and the undergraduate levels of many sciences) there is a temptation to teach solely through the transmitting of facts, rather than encouraging discussion and dissent.[11] This approach, which many undergraduates have seen, can condition them to interpret what they learn in terms of an authority-based view of "truth" that in turn leaves them unequipped to reason about situations involving no single correct answer or think cogently about ethical trade-offs.[23,34] We want to teach our students to move past this authority-based view and find the best, most efficient solution to technical problems; we argue that the same skills must be developed to engage with ethical challenges that arise from the substance of their work as well.

Many courses focused on both research and ethical considerations taught through fiction have been offered worldwide, including at Humboldt University in Berlin, Germany,[a] and a version focused on legal issues at Stanford University.[b,1–3,19,20] Courses in other fields use literature (including science fiction) in non-majors courses as both a "hook" and a platform for exploring core ethical issues.[3,13] Scholars in other humanistic disciplines (such as history and philosophy) have also argued that literature is an invaluable teaching tool for ethics and other topics; see Copp,[16] Davis,[17] and Pease.[35] The common observation is that a fiction-based approach makes it much easier to push beyond a review of best practices toward a more in-depth education in ethical reasoning; Nevala-Lee[33] said: "[…] fiction often removes the intellectual and emotional resistance some students might at first feel towards the subject of ethics."

A key part of ethics education is helping students see beyond their own reflexive assumptions about what is true or right.

Ethics and Values in Computer Science

Researchers in computing, as in all professions, hold multiple and often conflicting sets of values, as well as different ways to approach living up to one's values. It is important to be clear that the purpose in teaching ethics is not to unify the field around a particular value system but to encourage reflection and precision of thought among all computer professionals. Teaching this way will, we hope, lead to an openness and exchange of ideas about both core values and best practices.

The very idea of a universally applicable ethical doctrine has serious problems. As anthropologist Melville Herskovits wrote in protest of the United Nation's Universal Declaration of Human Rights, the declaration—although intended "to be applicable to all human beings … [is] conceived only in terms of the values prevalent in countries of Western Europe and America."[15] That is, any attempt to codify a universal definition of the "right" way to be human cannot, by definition, take account of the particular social and ethical context of individual cultures. Cultures that have historically been most oppressed would thus be the most likely to be ignored or delegitimized by a "universal" declaration.

Although the precise status and possibilities of human rights discourse continue to be debated, scholars in both ethics and anthropology agree there is no way to formulate universal precepts of this kind that do not, on some level, reinforce the very kinds of social inequality they are designed to combat. The idea that a single code of laws or duties would solve all problems, and that our responsibility as teachers is to transmit those laws to students, is appealing but ultimately false. As Callahan[10] says, "No teacher of ethics can assume that he or she has such a solid grasp on the nature of morality as to pretend to know what finally counts as good moral conduct. No society can assume it has any better grasp of what so counts as to empower teachers to propagate it in colleges and universities. The premise of higher education is that students are at an age where they have to begin coming to their own conclusions and shaping their own view of the world. It is the time and place to teach them intellectual independence, and instill in them a spirit of critical inquiry."[10]

The responsibility of an ethics instructor is to train students to engage in understanding and reasoning. The students are thus prepared to navigate situations that offer no clean solutions and engage other computer science practitioners in discussion about what and

a http://waste.informatik.hu-berlin.de/Lehre/ws0910/dystopien/

b http://web.stanford.edu/class/cs122/

how to choose. Callahan[10] also endorses the idea of helping " ... students develop a means and a process for achieving their own moral judgments" when confronted with challenging situations.

It is essential that open ethical debates between well-informed practitioners take place. Computer science does not take place in a vacuum; to an ever-increasing degree, the IT systems and platforms, from search engines to smartphones, that are built by computer scientists and engineers are creating and redefining the social, political, and individual contexts in which human beings understand themselves.[21] Whatever principles and norms are adopted by computer scientists, and reinforced through the design and deployment of their systems, will have profound ethical and societal implications. Teachers and leaders in the field have a responsibility to drive the discussion about the effects of their own work and the work of their students. Indeed, Boyer[6] argued that academics have a responsibility to engage students and the public with their research.

We have started to see this engagement through a number of initiatives in the computer science community, including the International Joint Conference on Artificial Intelligence 2015 letter on autonomous weapons research[c] and the 2017 follow-on letter signed by CEOs of tech companies around the world;[d] ACM statement on algorithmic accountability;[e] development of the IEEE standard for algorithmic bias considerations;[f] and new conferences and research groups focused on fairness, accountability, and transparency,[g] as well as conferences focusing on the effect of artificial intelligence on society.[h] These debates are important for shaping the direction of the field, even though they rarely result in consensus. The utility of the debates is not that they result in standardized practices but rather that individual practitioners become more thoughtful and better informed about their work and its long-term effects.

As in other areas of thought, this viewpoint diversity is a strength when it can be harnessed toward a productive exchange of ideas and perspectives. An example of such an exchange is the ongoing debate within the artificial intelligence research community about the appropriate value systems on which to build artificial intelligence systems. The goal of teaching ethics is to foster the debates and equip practitioners to participate productively. It does so, not by imposing a value system on students, but by informing them about the range of ethical descriptive and evaluative tools available to them.

At the same time, educators should make students and professionals aware of the social ramifications of their work, that research, development, and implementation can be carried out in a variety of ways and for a variety of ends. Computer science educators should dedicate significant time to ethics education, helping enable students to make informed, thoughtful, ethical choices about technology and its applications and implications.

What is ethics? Ethics can be understood as the task of answering "What should I do?" which is never a simple matter. Ethics includes both thought and practice, an organized and intentional reflection on morality and the effort to live in ways that are good, just, and/or right. Although many people use the words morality and ethics interchangeably, many ethicists understand them to be different. One common way of drawing the distinction is to define "morality" as a set of values or a worldview and "ethics" as the practice of reflecting on those values and their foundations and applications.[4,22]

There are many different, often conflicting, ways to understand how to be moral. The clashes are sometimes between people who share the same fundamental premises and method of inquiry into how to be moral but disagree about conclusions. Other times, the clashes are between people whose basic ideas of how to answer the question of how to be moral conflict with one another. Most approaches to morality can be understood in terms of the three major traditions of ethical thought—deontological ethics, virtue ethics, and utilitarianism—with each growing out of different core questions and ways of seeing the world.

Ethics is typically understood to be normative; that is, it is aimed at establishing norms of thought, values, or conduct. This assumption is especially prevalent in professional ethics courses that are typically used as a means to steer students' future behavior toward a set of professionally agreed-upon values (such as professionalism and honesty).[26] But ethics is also a tool for description, furnishing decision makers with a critical framework that enables them to understand what is happening in a given situation and what is at stake in any action they might take. The boundary between normative and descriptive functions is sometimes fuzzy; for example, it is often the case that different details of a situation will appear salient depending on which ethical approach one adopts. This malleability of relevant details can make ethics itself seem murky or imprecise. However, teaching students to appreciate this difference, understand the modes of reasoning that they or others might employ in making an ethical decision, and move between these reasoning structures themselves is the goal of a good ethics course.

Educating students in the descriptive functions of ethics is as important as communicating to them the professional norms of computer science. Computer science is a field in which everyday practice and problem solving takes place in a context that could barely be imagined the decade before. Educators cannot predict the ethical quandaries their students will face. With an education in ethical description, the students will be better able to engage in subtle and substantive ethical reasoning when new and challenging problems confront them.

Practical challenges of teaching ethics. Ethics education is a notable challenge for two reasons. First, in the absence of any ideal universal ethics program, students must be taught how to approach problems as distinct from being led to particular pre-ordained conclusions that might narrow their vision and exclude important elements of a given problem. Second is how to achieve this goal while overcoming the biases students often bring to the classroom.

---

c http://futureoflife.org/AI/open_letter_autonomous_weapons

d https://futureoflife.org/autonomous-weapons-open-letter-2017

e https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf

f http://sites.ieee.org/sagroups-7003/

g http://www.fatml.org/

h http://www.aies-conference.com/

Teaching how, not what, to think. It is important to consider what it means for us as educators to say we want to inform our students how to think instead of what to think. It is tempting to assume we can formulate a set of rules in natural language, refine them until we agree on them, and then proceed as if these rules can be applied without further reflection. However, the real world is messy, and rules that may seem reliable under one set of conditions can falter under others. Furthermore, language is not always identical to the world it is intended to describe. Different people often describe the same experience in different ways or understand the same phrase to refer to different phenomena. At minimum, such universal rules would require everyone who relied on them to engage in ongoing reflection about their own understanding and application of the rules to the world.

Both the appeal of this rule-based approach and its limits can be seen with respect to the question of programming robots against concrete actions (such as law enforcement robots never to shoot a human). While this operating principle seems at first like a straightforward way to ensure the preservation of human life, it is not difficult to imagine scenarios in which shooting a person, perhaps even lethally, can be expected to save the lives of others. But how should a robot calculate the risks and values at stake in such a scenario? What sorts of input should it use when ascertaining if it should shoot a human? What sorts of input should it ignore? And what are the social costs or benefits of using robots that will shoot a human under certain circumstances? Another example is the ongoing recent discussion about the classic trolley problem in light of the rapid advance of self-driving cars.[5]

Ethics education often requires a different kind of education from understanding and applying an established body of knowledge. In computer science, knowledge usually leads to action; if one chooses to create or program a system to solve a problem, and know how to do it, there is little reason not to solve the problem in the most direct and efficient way possible. Ethical understanding, however, requires an additional layer of commitment. One must overcome both the temptation to adopt an easier or more self-serving course and the distractions that might prevent someone recognizing an ethical problem in the first place. It is not difficult to imagine a student who can get 100% on an exam correctly identify terms and offer cogent and sensible solutions to hypothetical scenarios but then enter the work world and act in ways that ignore ethical consequences or even violate their[i] own values. This student might not stop to think they have acted wrongly; or such a student might notice but consider practical or professional pressures to be more important. An ethics course is successful only if it goes beyond equipping such a student with information and knowledge they can use but also prepares students to scrutinize their use of that knowledge, even when doing so is not convenient or comfortable.

In order to avoid causing great harm in the world, any field that involves practice requires not only technical proficiency of its practitioners but also ethical proficiency, as manifest not only in a command of the relevant knowledge but also the inclination and ability to let that knowledge take precedence over laziness or self-interest. That is, a successful professional ethics education does not just offer resources to indicate how problems can be identified and addressed; it also trains students to avail themselves of those resources, even when it is possible and easier not to. Teaching such skills and habits to students is a challenging task that cannot be successfully realized through cross-disciplinary requirements alone but must be integrated into their computer science education.[38] The number of recent professional society calls to deal with algorithmic bias and the disparate effects of information technology systems makes clear that computer science departments must engage directly with this responsibility.

Negotiating student biases. A key part of ethics education is helping students see beyond their own reflexive assumptions about what is true or right. Our classroom experience shows us that introducing students to three of the major schools of ethical theory—deontology, virtue ethics, and utilitarianism—helps broaden students' ability to recognize and reflect on those assumptions. While all three schools have proponents among philosophers, theologians, and other scholars who work in ethics, broader cultural discourse about ethics tends to adopt a utilitarian approach, often without being aware that there are other ways to frame ethical inquiry. This larger cultural reliance on utilitarianism may help explain why it consistently seems, to students, to be the most crisply defined and "usable" of the ethical theories. But there are significant critical shortcomings to this popular version of utilitarianism. The concept of "the greatest good" is notoriously ill-defined in utilitarianism, and while trained philosophers struggle to identify or formulate a suitable definition, the gap typically goes unnoticed in less-philosophical circles, enabling agents to plug in their own definition of "the good" without submitting it to scrutiny. Furthermore, it is very easy to try to apply the basic formula of utilitarianism—the greatest good for the greatest possible number—to a decision without thorough consideration of all those who will be affected. This move enables agents to declare they have pursued a morally reasoned course when, in fact, they have calculated the benefits only to themselves and those in their immediate sphere. This difficulty attaining a sufficiently broad understanding of the effects of actions, and thus in appropriately computing the utility of those actions, can curtail the ability to have a substantive ethical discussion, even insofar as everyone assents to utilitarianism.

In our experience teaching ethics courses under the auspices of computer science departments, we find that students are often drawn first to utilitarianism, perhaps because it seems more computational than the alternatives. One of the most important aspects of the course is to broaden their experience to help them see past the non-rigorous version of utilitarianism to which they were previously exposed. The aim is not to demonstrate the superiority of one approach over another but rather to help them understand the uses and limits of each approach. This limitation can be exemplified by the question of whether to replace factory

i In the spirit of inclusion, and in deference to shifting usage norms, we use "they" as a singular, non-gendered pronoun, as well as for the more traditional third-person plural; we trust context disambiguates.

workers with robots. They may focus on the happiness of the factory owners, shareholders, and those who will pay less for manufactured goods, without considering the utility of the human factory workers and those whose jobs depend on factory workers having money to spend; or even the more high-level question about whether or not it is reasonable to consider human beings and machines as interchangeable. Indeed, the three approaches can be complementary, or even mutually informative; for example, recent theorists have argued that virtue ethics is best seen as part of successful deontology.[27]

Why fiction to teach ethics? Stories—literature, plays, poetry, and other narrative forms—have always been a way to talk about the world as it is, telling us what it is like and what effect our choices will have. Whether they are transmitted in print or through other media, stories play a potent role in shaping the thoughts and ideas of individuals, as well as the cultural norms of the societies in which they live.

Scholars of ethics have, in the past several decades, embraced fiction as an ideal way to think about and teach ethics, because, as philosopher Martha Nussbaum[32] writes, fiction " ... frequently places us in a position that is both like and unlike the position we occupy in life; like, in that we are emotionally involved with the characters, active with them, and aware of our incompleteness; unlike, in that we are free of the sources of distortion that frequently impede our real-life deliberations." By offering the reader both immersion and distance, an ethics course based in fiction helps students perceive the degree to which ethical quandaries are tangled up in other aspects of life while furnishing a context that keeps them connected to abstract principles and questions. As such, fiction-based ethics education helps them cultivate the capacity to recognize ethically complex situations as they arise or extract an ethical dilemma from a larger context. This combination of qualities also helps students develop the moral imagination that is a key component of successful ethics education.[10] The common alternative is to provide them with a prepackaged case studies in which the particular ethical dilemma under study is cleanly identified for the student.

Science fiction is particularly well suited to teaching computer ethics. As Alec Nevala-Lee[31] says, "Science fiction has been closely entwined with military and technological development from the very beginning. The first true science fiction pulp magazine, *Amazing Stories*, was founded by editor Hugo Gernsback expressly as a vehicle for educating its readers about future technology." Our project builds on this long-recognized insight—that science fiction is, in key respects, better able than "realistic" fiction to reflect the near future (or possible futures) in which computer professionals work. Science fiction thus permits a curricular design that hews more closely to the concerns and quandaries of computer-related fields of study and work. A successful ethics course will reframe the task of ethical engagement so students understand the ongoing responsibility to ask ethical questions of themselves and their work; and further, that they are equipped to perceive, describe, and understand the challenges as they arise. We find that science fiction makes the key ethical questions of technology development and use more vivid and engaging and the critical resources for addressing ethical questions more intelligible.

We take science fiction in its broadest sense, as the fantastical worlds or even the futuristic technology gives us a starting platform for discussion. The category of science fiction was first described by Hugo Gernsback, for whom the prestigious Hugo Prize is named, in the editorial of the first issue of *Amazing Stories* in 1926 as, " ... I mean the Jules Verne, H.G. Wells, and Edgar Allan Poe type of story—a charming romance intermin-

Spring 2018 Syllabus.

Week 1. Using ethical language; in-class discussion
Read Isaac Asimov's short story "The Dead Past"

Week 2. Professional ethics
Read the ACM's and the IEEE's codes of ethics

Week 3. Utilitarianism
Read Harlan Ellison's short story "'Repent, Harlequin!' Said the Ticktockman"
Ethical description exercise #1 due in class

Week 4. Deontology
Read Elizabeth Bear's short story "Dolly"
Ethical description exercise #2 due in class

Week 5. Virtue ethics
Read E.M. Forster's short story "The Machine Stops"
Ethical description exercise #3 due in class

. . .

Week 6. Selfhood and technological mediation
Read James Patrick Kelly's short story "Itsy Bitsy Spider" and
Tom Sorell's and Heather Draper's paper "Robot Carers, Ethics, and Older People"
Ethical description exercise #4 due in class

. . .

Week 9. Privacy
Read Ken Liu's short story "Here-and-Now" and
Articles: Helen Nissenbaum's "Privacy as Contextual Integrity,"
Manan Kakkar's "A Case Against Online Privacy," and Adam D. Moore's "Privacy, Speech and Values: What We Have No Business Knowing"
. . .

Week 14. What is ethical warfare?
Read Linda Nagata's short story "Codename: Delphi," Ronald C. Arkin's essay "Ethical Robots in Warfare," Jean Elshtain's paper "The Problem of Dirty Hands," and Emerson T. Brooking's and Peter Singer's "War Goes Viral"
Ethical argument assignment #3 due in class

Last week. Professional ethics, the importance of integrity
Read E. Saxey's short story "Not Smart, Not Clever"
Ethical argument assignment #4 due in class
Reread Asimov's "The Dead Past"

gled with scientific fact and prophetic vision." Using this broad definition, almost any fiction dealing with sufficiently advanced technology is science fiction. Though the majority of the literary and philosophical establishment has not, until recently, seen science fiction as a venue for serious ethical thinking, this fact reflects longstanding biases in the field rather than the merits or possibilities of science fiction itself.

Fiction allows educators to reframe recognizable human situations and problems in terms of unfamiliar settings and technology. Hence, any fiction, and especially science fiction in the case of technology, can be an ideal medium for raising and exploring ethical concerns. By presenting a familiar problem (such as conflicts between different social groups or the invasion of privacy in unfamiliar terms and settings), a work of science fiction can mitigate a reader's tendency to defend, reflexively, their own previously held views. As Nussbaum[32] writes, "Since the story is not ours, we do not get caught up in the vulgar heat of our personal jealousies or angers or the sometimes blinding violence of our loves." In this way, science fiction creates an opportunity for students to gain fresh insight into, and even empathy for, ethical positions and people whose real-world analogues are not embraced by their values or politics.

We thus advocate science fiction for several reasons in addition to the ones outlined here. First, the use of futuristic or alien settings allows students to detach from political preconceptions and experience the dilemmas of plot and characters as something fresh. Second, it has so far proven popular and effective with students. One student wrote the following on a Spring 2017 anonymous course evaluation: "Going into this course, there were several times that I could acknowledge an ethical situation and had my own ideas as to whether it was 'right' or 'wrong,' but I couldn't necessarily articulate why. This course gave me the tools to be able to have a meaningful discussion about these topics. It was also a productive way to get out of the coding mindset, take a step back, and consider what other results might come from the technologies that we will be making. Phenomenal course, and phenomenal instructor." Finally, some of the science fiction we chose also posits new science infrastructure and allows students to think about doing research and development outside the fairly rigid industrial and academic boxes, driven by something other than current funding paradigms. This creative thinking about practical problems, according to some philosophers[29,37] and educators,[14] is a crucial component in developing the ethical reasoning abilities of students. All these reasons, along with the distance from the material that can be created through fiction, have led to a very successful course taught more than eight times as of August 2018 and that has won us multiple teaching awards.

The Course

The aim of the course is to prepare our students to recognize ethical problems in their present and future work as technologists, focusing on methods of applied ethical reasoning (for the future), as well as on particular current problems. During class discussion and in homework assignments, they analyze both science fiction stories and brief articles, using the major ethical theories not only as evaluative tools but as a descriptive apparatus to enable them to recognize problems and consider possible solutions from multiple perspectives. As we have seen, this focus on ethical theory as a descriptive tool, combined with the use of science fiction stories as an arena for ethical description and analysis, sharpens the students' ability to perceive and describe ethical challenges and expands their capacity to address them with creativity and nuance. An abbreviated example syllabus is outlined in the figure here.

The class opens with a crash course on ethical theories and a review of the IEEE and the ACM codes of ethics. Students consider the different modes of ethical engagement invited by each code and discuss whether, and in what ways, either one is likely to affect their decision making. Although this discussion typically evinces varying opinions on the usefulness or relevance of either code, there is near-universal consensus that the codes are not, by themselves, sufficient to help an IT professional address the challenging problems that may arise. We, the instructors, stress this is a problem common to all codes of ethics and the solution is not a more-perfect code but rather IT professionals better prepared to engage in ethical reasoning, and thus to make use of professional codes.

The course then spends several weeks on in-depth study of each of the three major ethical theories—utilitarianism, deontology, and virtue ethics—with one day for each on a critical reading assignment that introduces the theory in detail and another day analyzing and discussing a short story from within the perspective of that theory. To prepare for these discussions, students write "ethical description exercises," answering guided questions about how the story world can be understood through that week's ethical lens. Some of these stories, particularly Elizabeth Bear's "Dolly," which is used to teach deontology, and E.M. Forster's "The Machine Stops," which is used to teach virtue ethics, end up as touchstones for the course, resurfacing in student discussions about later subjects.

After helping build the students' analytic competency in ethical theory, the course moves to a consideration of major ethical concerns in IT, including surveillance, the interrelationship between news and social media, and self-driving cars. On the strength of the assigned science fiction stories, students consider both immediate practical problems and deep underlying issues that recur in IT ethics past, present, and possibly future.

Each story touches on multiple core issues, enabling the students to appreciate, and grapple with, the interconnectedness of the various challenges they will confront. Stories like James Patrick Kelly's "Itsy Bitsy Spider" and Paul Shoemaker's "Today I Am Paul," both focusing on carebots looking after aging parents with dementia, serve as the basis for a discussion of carebots in particular but also inspire broader discussions on how technological interventions can change the conditions in human relationships. Paolo Bacigalupi's "The Gambler" helps frame a discussion of new media and the attention economy, highlighting the particular hurdles this new information environment creates for minority experience and positions.

Ken Liu's "Here-and-Now" offers a potent view of the personal and social stakes of the post-privacy era, particularly in the context of the mostly unregulated gig economy that depends so heavily on IT innovations. And Michael Burstein's "Teleabsence" explores how technological innovations designed to address social inequality can in fact exacerbate it while raising probing questions about the powers and limits of how one might redefine oneself on the Internet. Although the reading list has changed with each iteration, these stories and others like them have formed the backbone of each version of the course.

In each such iteration, our students have emerged from the semester's reading inspired, troubled, and invigorated by the new perspectives they have gained on their future work.

The assignments in the course help develop their capacity for attention and critical thought in a manner intended to serve them well throughout their professional lives. By working descriptively with three different ethical theories, they develop a rich critical vocabulary for recognizing ethically fraught situations as they arise. The questions given to the students for a particular story are deliberately open-ended, requiring them to identify and formulate the problem from the ground up, an approach that addresses a practical gap created when they are taught using only case studies. This open-endedness also fosters a wider range of responses than a more closely tailored set of questions, thus creating a more varied class discussion.

Through the multiple writing assignments, the students not only become aware of a range of potential ethical challenges in their work in computer science but also alert to the variety of ways these problems might initially emerge. They are thus able to identify potential ethical risks in a given technology or model or in a company's and the public's use of the technology or model. They are better prepared to articulate their arguments for why a given approach is the most (or least) ethical choice and see past incomplete or specious defenses of potentially unethical projects.

Science fiction makes the key ethical questions of technology development and use more vivid and engaging and the critical resources for addressing ethical questions more intelligible.

Example Story Materials

Here, we include an example of the pedagogical materials we have developed to capitalize on the lively accessibility of the fiction-reading experience while also helping the students come to grips with the complexity of considering a problem in the context of the wider world where it takes place. These materials include both a story frame to introduce the stories to students and a pedagogy guide to help instructors. The stories we have collected for the course (and, no doubt, many others) are engaging enough to spark energetic debate about ethical questions on their own and reward sustained scrutiny along ethical lines with several layers of productive challenge beyond an initial encounter. Once the problems illustrated by the narrative are described and conceptualized, the full ethical implications and challenges can be understood by "re-embedding" the problem back into its narrative context. The students should then consider how the world of the story created the conditions for both the external problems and the internal struggles addressed by the related characters.

The story frame furnishes the students with light guidelines, preparing them to pay attention to particular issues without instructing them how to answer, or even ask, ethical questions. The story frame thus leaves room for the students to discover the questions for themselves and grapple with the challenge of identifying and naming the problems at hand. This choice not only helps preserve the excitement of discovery that comes with reading good fiction but also requires the students to undertake these tasks on their own. While their own initial attempts to frame, define, and address ethical problems are likely inadequate, their attempts to do so both individually and collectively are an essential part of the learning in an ethics course, as the real-world problems they encounter will not come with a set of pre-formulated guidelines to steer them toward the nominal best answer.

The pedagogy guide, in addition to offering generalized tips for stimulating and sustaining productive discussion about fiction and ethics, also points the instructor toward relevant themes, details, and patterns in the text. These details and patterns do not, by themselves, constitute an "answer" to any of the core ethical questions raised by the stories. As a list of facts, they are not especially helpful for students grappling with the core ethical challenges of a given story. In the context of an ongoing discussion, the instructor can introduce these details to raise new questions or challenge provisional explanations about how the world of the story works or why characters make the choices they do. In story discussions—and, indeed, in discussions concerning the real world—students often begin the course by wanting to find tidy answers for challenging ethical problems. To counter this impulse, future instructors will find it useful to interject into the discussion details that complicate the students' explanations. In this way, discussion of the story worlds can help train students to perceive complexity in the real world.

Story frame for "Here-and-Now." The story under study here is Liu's "Here-and-Now," a short story that has sparked lively and productive discussion among students in previous versions of the course. Liu, a trained computer scientist, has written several excellent stories in recent years that directly address issues in computer ethics, material circulated to students, along with the story text itself; "Here-and-Now" is available for free at http://www.kasmamagazine.com/here-and-now.cfm

The story itself[25] begins with this sentence: "It's amazing what you can get, just by asking."

How much is information worth? That is the question Aaron, the protagonist of "Here-and-Now," is forced to confront over the course of one complicated afternoon and evening. Aaron is one of thousands (if not millions) of people using a new app called *Tilly Here-and-Now* that allows users to pose anonymous requests for "information" of any kind. The story asks deceptively simple questions as to why information matters. It also points out that some kinds of information are much more meaningful or valuable to some people than to others, asking readers to consider whether that difference should matter, and how.

The world of the story is not quite the same as ours but is similar in many ways. It appears that Centillion, the app's parent company, has achieved data-management capabilities that are not yet available in the real world, though we recognize the possibility is certainly on the horizon. Likewise, nothing exactly like the *Here-and-Now* app exists yet, but it is a plausible amalgam of many apps and services that do exist, including *TaskRabbit*, *Pokémon Go*, and *YikYak*. Indeed, the app in the story was based on one described in a 2013 academic paper.[39] Still, we are fast approaching a world like the one in the story, and it is not difficult to imagine an app like *Here-and-Now* existing here, and now.

Discussing ethics in the context of fiction can make it easier for instructors to adopt an open-ended approach required for a good ethics course.

Study questions. Among the many essential ingredients of *Tilly Here-and-Now*'s economy are money and information but also interest on the part of users, information requesters and information gatherers alike. What are the sorts of interest that might lead someone to use the app in either of these roles? Are any of these interests in tension with the others?

Does it matter that Tilly's request function is anonymous? Why or why not?

Early in the story, Aaron decides "*Tilly Here-and-Now* made you more aware of the world around you … more connected to your community." How do the events of the story itself confirm or challenge that conclusion? Characters in the story you can use to think about this question include Aaron's acquaintance, Lucas, Aaron's parents, the unnamed people whose bounties are being fulfilled, the girls in the video Lucas has purchased, and Aaron himself.

The reward for fulfilling an information request is called a "bounty," rather than, say, a "fee," "one-time payment," or other possible term; you can probably think of others. How does that choice of word affect the way the reader think about the relationship between the information-requester and the information gatherer? Does it affect how the reader thinks about the relationship between either of these individuals and the information that is gathered? Do you think the choice of the word "bounty" has an effect on the characters in the story, as well? If so, in what way?

Moreover, who has access to the requests, and how is it controlled?

Instructor's guide. This material is available to instructors to help them guide in-class discussion of the story.

The *Tilly Here-and-Now* app exists in a world that is just different enough from our own to be provocative but similar enough to feel intuitive. Students may be tempted to jump straightaway to talking about the app itself, independent of the story. But this particular narrative provides an exceptionally effective window onto Liu's slightly reimagined world, and the discussion will likely be more focused and productive if you dedicate at least 20 minutes to 30 minutes to discussing Aaron's experiences and reflections before moving onto the more general implications of the Tilly app.

As always, the best approach is a Socratic one, in which you guide your students toward discovering things for themselves. Here are some observations and details about the story. You can use them to ask "fishing" questions if you think your students are missing important details or to prompt them to reassess their view of the story if they have settled on a version that ignores such details.

Aaron. Aaron is interested in information about others. He likes claiming bounties and furnishing others with the information they want but also likes trying to figure out why it is that people want them. When Lucas baits him, saying, "I got something cool," Aaron cannot help asking about it.

On the other hand, Aaron hates giving up information about himself to the people he knows. He does not want his mother to know about his part in the school play and will not tell Lucas how much money he earned. In the entire story, we learn of only one instance in which Aaron willingly shares information with another character, when he teaches his mother (before the beginning of the story) about *Tilly Here-and-Now*. By the end of the story, Aaron regrets having shared that information, since his mother is now using the app "against" him to learn things about him and about his father.

The individuality of knowledge. At several points in the story, both major and minor, the reader's attention is directed to the ways information matters more to the people it touches directly. The story thus adds a new layer to frequently expressed concerns about privacy, focusing on the damage done to the character(s) whose information is known or made available. As the story explains, the person who knows can be just as affected or damaged by that knowledge as the subjects about whom it is known.

Lucas is happy with his video of two girls kissing (which strikes Aaron as invasive of the girls' privacy) but "Would have been even better if they're people I know," as Lucas says. "Next time I'm going to raise the bounty and limit the range more. It's amazing what you can get, just by asking."[25] Lucas anticipates that knowing the girls involved would make the video more satisfying. The invasion of private space is part of the pleasure.

Aaron and Lucas respond very differently to the license-plate request, not only because Aaron recognizes the plate number but because he has something personal at stake in the fulfillment of the request, and in its asking. Previously, the reader has seen Aaron wonder why he is being asked to fulfill this or that request, but never whether he should. Only when the request touches him personally does he realize the damage that might be done if it is fulfilled.

Aaron himself is later undermined (in a small way) by another *Tilly Here-and-Now* user, fulfilling another of his mother's requests, when she discovers he has been cast in the school play. On the surface, this plot point lines up primarily with more typical concerns about privacy. Aaron, who had hoped to conceal the information about his being cast in the play, is the one who has been injured but, insofar as Aaron trusts his mother less, she is also damaged.

Information control and performance. The story repeatedly touches on the theme of people pretending to be who they are not, as signaled at the opening of the story, when the reader learns Aaron has been cast in a play. A play is a performance, but the "deception" is a matter of mutual consent; the audience knows it is watching actors, and in this sense the play does not represent a miscarriage of knowledge.

This non-deceptive deception differs from the way Aaron's parents talk to each other over dinner toward the end of the story. Aaron knows by then that his mother suspects his father of cheating and he halfway suspects his father as well, but they treat each other normally, as if nothing is wrong. "He couldn't hear anything different in their tones. His mother acted like she had never asked the question. His father acted like he had nothing to hide."[25] Aaron's mother, and possibly his father as well, perform with an intent to mislead. But whom are they misleading—Aaron, each other, or both? And when did the deception begin?

It is also worth raising the question of whether, and how, Aaron's own actions qualify as misrepresentations, as in his desire to keep his role in the play a secret, and his own Tilly request, which is designed to distract Lucas from fulfilling his mother's request.

Additional topics. Liu's "Here-and-Now" also raises issues of access control and information integration, or combining different possibly innocuous sources to complete more complex, thorough, and possibly invasive records. At one point, the reader's perception shifts when Aaron recognizes his father's license plate. How do the different ethical theories frame the possibility of deanonymization in the story, either deliberately or accidental? Discussing deanonymization can lead to further discussion of hacking and Wikileaks, trust and distrust in data scrubbing, as well as other directions.

Ethical description writing assignment. The purpose of this assignment is description. Addressing the points cited in the following paragraphs, describe Liu's story in terms of one of the three major theories of ethics. (You will receive separate instructions telling you which theory to use.) Be sure to title your assignment "Here-and-Now: [name of ethical theory]."

Assigned theory. Using the concepts and worldview of your assigned theory, give a two-to-four sentence summary of the central ethical problem(s) in the story.

Ethical problems. What is at stake in the ethical problem(s) so described? That is, what possible goods could be gained or lost or what kinds of harm could occur or be prevented? Using the language of your theory, explain why these costs or benefits are significant.

Characters. What character(s) is/are in a position to take meaningful action with respect to the problem? What about their character or circumstances positions them to take such action?

Course of action. Choose one such character from your answer. Using the language and concepts of your assigned theory, describe the course of action this character takes in the story. Are there other possible courses of action the story suggests the character might have taken? Describe them, again using the language of your assigned theory. According to that theory, what might be a better course of action, and why?

Argument. What argument do you think the ending of the story intends to make? You are still describing, rather than arguing. Use the language of your assigned theory to describe Liu's argument.

Students will bring these assignments to class on the day they are due. They are welcome to make notes on them, over the course of discussion, for their own edification and turn them in to the professor at the end of class.

Conclusion

Teaching ethics to computer science students is a pressing responsibility for computer science faculty but also a challenge. Using fiction as the basis for an ethics course offers several advantages beyond its immediate appeal to many students and some faculty. First, fiction offers students a way to engage with ethical questions that helps them cultivate their capacity for moral imagination; science fiction in particular can make the ethical stakes of blue-sky projects vivid, pressing, and immediate. Second, stories offer students the chance to develop their writing and verbal skills in ethical description. And finally, discussing ethics in the context of fiction can make it easier for instructors to adopt an open-ended approach required for a good ethics course. A course built around fiction enables instructors to incorporate the best and most useful aspects of a humanistic approach to ethics education while remaining close to the central technological concerns within computer science.

Acknowledgments

We would like to thank John Fike, Cory Siler, and Sara-Jo Swiatek for proofreading and discussion to improve this article. The ideas here are based on work supported by the National Science Foundation under Grant No. 1646887. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. C

References

1. Bates, R., Goldsmith, J., Berne, R., Summet, V., and Veilleux, N. Science fiction in computer science education. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (Raleigh, NC, Feb. 29–Mar. 3). ACM Press, New York, 2012, 161–162.
2. Bates, R., Goldsmith, J., Summet, V., and Veilleux, N. Using science fiction in CS courses. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, GA, Mar. 5–8), ACM Press, New York, 2014, 736–737.
3. Bates, R.A. AI & SciFi: Teaching writing, history, technology, literature and ethics. In *Proceedings of the ASEE Annual Conference & Exposition* (Vancouver, B.C., Canada, June 26–29). American Society for Engineering Education, Washington D.C., 2011, 1–12.
4. Beard, M. Ethics, morality, law: What's the difference? The Ethics Centre, Sydney, NSW, Australia, Aug. 27, 2012; http://www.ethics.org.au/on-ethics/blog/september-2016/ethics-morality-law-whats-the-difference
5. Bonnefon, J.-F., Shariff, A., and Rahwan, I. The social dilemma of autonomous vehicles. *Science 352*, 6293 (June 24, 2016), 1573–1576.
6. Boyer, E.L. *Scholarship Reconsidered: Priorities of the Professoriate.* Jossey-Bass, San Francisco, CA, 1997.
7. Burton, E., Goldsmith, J., Koenig, S., Kuipers, B., Mattei, N., and Walsh, T. Ethical considerations in artificial intelligence courses. *AI Magazine 38*, 2 (Summer 2017), 22–35.
8. Burton, E., Goldsmith, J., and Mattei, N. Teaching AI ethics using science fiction. In *Proceedings of the First International Workshop on AI, Ethics and Society at the 29th AAAI Conference on Artificial Intelligence* (Austin, TX, Jan. 25–30). Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 2015, 33–37.
9. Burton, E., Goldsmith, J., and Mattei, N. Using 'The Machine Stops' for teaching ethics in artificial intelligence and computer science. In *Proceedings of the Second International Workshop on AI, Ethics and Society at the 30th AAAI Conference on Artificial Intelligence* (Phoenix, AZ). Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 2016, 80–88.
10. Callahan, D. Goals in the teaching of ethics. Chapter 2 in *Ethics Teaching in Higher Education.* Plenum Press, New York, 1980, 61–80.
11. Colbeck, C.L. Merging in a seamless blend: How faculty integrate teaching and research. *The Journal of Higher Education 69*, 6 (Nov./Dec. 1998), 647–671.
12. Davis, B.G. *Tools for Teaching, Second Edition.* Jossey-Bass, San Francisco, CA, 2009.
13. Dils, L.S. Science fiction and the future. Curriculum Unit, Yale-New Haven Teachers Institute, New Haven, CT, 1987; http://teachersinstitute.yale.edu/curriculum/units/1987/2/87.02.04.x.html
14. Edmiston, B. Ethical imagination: Choosing an ethical self in drama. Chapter 3 in *Imagining to Learn: Inquiry, Ethics, and Integration Through Drama.* B. Edmiston and J.D. Wilhelm, Eds. Heinemann Drama, Portsmouth, NH, 1998.
15. Executive Board of the American Anthropological Association. Statement on Human Rights. *American Anthropologist 49*, 4 (Oct.-Dec. 1947), 539–543.
16. Garcia Iommi, L. Let's watch a movie!: Using film and film theory to teach theories of international politics from a critical perspective. In *Proceedings of the American Political Science Association 2011 Annual Meeting* (Seattle, WA, Sept. 1–4). American Political Science Association, Washington, D.C., 2011; https://ssrn.com/abstract=1903282
17. Goering, S. Using children's literature as a spark for philosophical discussion: Stories that deal with death. Chapter 15 in *Ethics and Children's Literature*, C. Mills, Ed. Ashgate, Farnham, U.K., 2014, 233–247.
18. Goldsmith, J. and Burton, E. Why teaching ethics to AI practitioners is important. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence* (San Francisco, CA, Feb. 4–9). Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 2017, 4836–4840.
19. Goldsmith, J. and Mattei, N. Science fiction as an introduction to AI research. In *Proceedings of the Second AAAI Symposium on Educational Advances in Artificial Intelligence* (San Francisco, CA, Aug. 7–11). Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 2011, 1717–1722.
20. Goldsmith, J. and Mattei, N. Fiction as an introduction to computer science research. *ACM Transactions on Computer Science Education 14*, 1 (Mar. 2014), 1–14.
21. Greenfield, A. *Radical Technologies: The Design of Everyday Life.* Verso, London, U.K., and New York, 2017.
22. Haidt, J. *The Righteous Mind: Why Good People Are Divided by Politics and Religion.* Vintage, New York, 2012.
23. Haworth, J.G. and Conrad, C.F. Curricular transformations: Traditional and emerging voices in the academy. In *Revisioning Curriculum in Higher Education*, J.G. Haworth and C.F. Conrad, Eds. Simon & Schuster Custom Publishing, New York, 1995, 191–202.
24. Huff, C. and Furchert, A. Toward a pedagogy of ethical practice. *Commun. ACM 57*, 7 (July 2014), 25–27.
25. Liu, K. Here-and-Now. *Kasma Online* (Nov. 1, 2013); http://www.kasmamagazine.com/here-and-now.cfm
26. Martin, C.D. and Weltz, E.Y. From awareness to action: Integrating ethics and social responsibility into the computer science curriculum. *ACM SIGCAS Computers and Society 29*, 2 (June 1999), 6–14.
27. McNaughton, D. and Rawling, P. Deontology. Chapter 15 in *The Oxford Handbook of Ethical Theory*, D. Copp, Ed. Oxford University Press, Oxford, U.K., 2006.
28. Mihail, R., Rubin, B., and Goldsmith, J. Online discussions: Improving education in CS? In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, GA, Mar. 5–8). ACM Press, New York, 2014.
29. Murdoch, I. The idea of perfection. Chapter 1 in *The Sovereignty of Good*, Routledge & Kegan Paul, Abingdon, U.K., and New York, 1970.
30. Narayanan, A. and Vallor, S. Why software engineering courses should include ethics coverage. *Commun. ACM 57*, 3 (Mar. 2014), 23–25.
31. Nevala-Lee, A. Deadline: John W. Campbell in World War II. *WorldCon 2016 Academic Track* (Kansas City, MO, Aug. 18, 2016) and personal communication.
32. Nussbaum, M. *Love's Knowledge: Essays on Philosophy and Literature.* Oxford University Press, Oxford, U.K., 1990.
33. Pease, A. Teaching ethics with science fiction: A case study syllabus. *Teaching Ethics: The Journal of the Society for Ethics Across the Curriculum 9*, 2 (Spring 2009), 75–82.
34. Perry, W.G. Cognitive and ethical growth: The making of meaning. Chapter 3 in *The Modern American College*, A.W. Chickering and Associates, Eds. Jossey Bass, San Francisco, CA, 1980, 76–109.
35. Rodwell, G. *Whose History?: Engaging History Students Through Historical Fiction.* University of Adelaide Press, Adelaide, Australia, 2013.
36. Rogaway, P. *The Moral Character of Cryptographic Work. Cryptology ePrint Archive*, Report 2015/1162, 2015; http://eprint.iacr.org/
37. Smith, B. Analogy in moral deliberation: the role of imagination and theory in ethics. *Journal of Medical Ethics 28*, 3 (Aug. 2002), 244–248.
38. Spradling, C.L. *A Study of Social and Professional Ethics in Undergraduate Computer Science Programs: Faculty Perspectives.* Ph.D. thesis, University of Nebraska-Lincoln, 2007; https://digitalcommons.unl.edu/dissertations/AAI3255458/
39. Von Der Weth, C. and Hauswirth, M. Finding information through integrated ad hoc socializing in the virtual and physical world. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies* (Atlanta, GA, Nov. 17–20). IEEE Press, Washington, D.C., 2013, 37–44.

Emanuelle Burton (enburton@uic.edu) is a faculty member in the Computer Science Department at the University of Illinois at Chicago, Chicago, IL, USA.

Judy Goldsmith (goldsmit@cs.uky.edu) is a professor of computer science in the Computer Science Department at the University of Kentucky, Lexington, KY, USA.

Nicholas Mattei (n.mattei@ibm.com) is a research staff member with the IBM Research AI-Reasoning Lab at the T.J. Watson Research Center, Yorktown Heights, NY, USA.

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/how-to-teach-computer-ethics-with-science-fiction

DOI:10.1145/3232559

Queueing theoretic models can guide design trade-offs in systems targeting tail latency, not just average performance.

BY CHRISTINA DELIMITROU AND CHRISTOS KOZYRAKIS

Amdahl's Law for Tail Latency

TRANSLATING THE IMPACT of Amdahl's Law on tail latency provides new insights on what future generations of data-center hardware and software architectures should look like. The emphasis on latency, instead of just throughput, puts increased pressure on system designs that improve both parallelism and single-thread performance.

Computer architecture is at an inflection point. The emergence of warehouse-scale computers has brought large online services to the forefront in the form of Web search, social networks, software-as-a-service, and more. These applications service millions of user queries daily, run distributed over thousands of machines, and are concerned with tail latency (such as the 99th percentile) of user requests in addition to high throughput.[6] These characteristics represent a significant departure from previous systems, where the performance metric of interest was only throughput, or, at most, average latency. Optimizing for tail latency is already changing the way we build operating systems, cluster managers, and data services.[7,8] This article investigates how the focus on tail latency affects hardware designs, including what types of processor cores to build, how much chip area to invest in caching structures, how much resource interference between services matters, how to schedule different user requests in multicore chips, and how these decisions interact with the desire to minimize energy consumption at the chip or data-center level.[2]

While the precise answers will come from detailed experiments with both simulated and real systems, there is great value in having an analytical framework that identifies the major trade-offs and challenges in latency-sensitive cloud systems. We aim here to complement the previous analyses on Amdahl's Law for parallel and multicore systems[1,11] by designing a model that draws from basic queueing theory

» key insights

- **Optimizing for tail latency makes Amdahl's Law more consequential than when optimizing for average performance.**
- **Queueing theory can provide accurate first-order insights into how hardware for future interactive services should be designed.**
- **As service responsiveness and predictability become more critical, finding a balance between compute and memory resources likewise becomes more critical.**

Analytical Framework

Amdahl's Law describes the speedup of a program when a fraction f of the computation is accelerated by a factor S. Speedup is then defined as

$$Speedup(f,S) = \frac{1}{(1-f)+\frac{f}{S}}$$

In a multi-core machine, Amdahl's Law captures the benefit from multiple cores in average performance. While this interpretation is still relevant, it is, by itself, insufficient for describing tail latency requirements. To bridge the gap we build upon ideas from queueing theory, which provides a framework to reason about task-arrival rates, service times, and end-to-end response times. Simple models (such as M/M/1 and M/M/k) are particularly attractive for first-order performance calculations because they can concisely describe performance in closed-form expressions.

M/M/1 model. We start with one of the simplest queueing models: the M/M/1 queue, modeling a system in which a single server processes incoming tasks. Tasks arrive under a Poisson process with rate λ. The service times also follow an exponential distribution, with rate parameter μ and mean service time $T_s = 1/\mu$ ($\mu = perf(r)$ in the main text of the article. A larger μ means a more powerful server and results in lower latency. Tasks are processed in a simple first-in-first-out order. This simple queueing system is stable when $\mu > \lambda$. In contrast, when $\mu > \lambda$, queued tasks keep increasing, leading to instability. The load of the system is defined as $\rho = \lambda/\mu$. Given these definitions, the mean number of tasks in the system is

$$E[N] = \frac{\rho}{1-\rho}$$

where N is a random variable for the number of tasks. Likewise, the mean of task response time (using random variable R) is

$$E[R] = E[N]\lambda^{-1} = \frac{1}{\mu-\lambda}$$

and the ρ-th percentile of response time is

$$m_p = -\frac{ln(1-p/100)}{\mu-\lambda}$$

Figure 1a outlines the 99th percentile of request latency as a function of the service rate μ. As μ increases, tail latency drops both at low and high load.

M/M/k model. We now extend the M/M/1 model to a more realistic system with k equivalent servers in order to model a multicore machine. Tasks are now added to a single, shared queue, where servers draw them from for processing. As with the M/M/1 model, tasks arrive under a Poisson process with arrival rate λ and each server processes tasks with service rate μ. Closed-form solutions for the mean response time and response-time percentiles exist but are more complicated than in the M/M/1 model. Specifically, system load is $\rho = \lambda/(k\mu)$. The probability that a new task must be enqueued is given by Erlang's C formula

$$C(k,\lambda/\mu) = \frac{\left(\frac{(k\rho)^k}{k!}\right)\left(\frac{1}{1-\rho}\right)}{\sum_{c=0}^{k-1}\frac{(k\rho)^c}{c!}+\left(\frac{(k\rho)^k}{k!}\right)\left(\frac{1}{1-\rho}\right)}$$

and the mean number of tasks in the system

$$E[N] = \frac{\rho}{1-\rho}C(k,\lambda/\mu)+k\rho$$

The average response time is

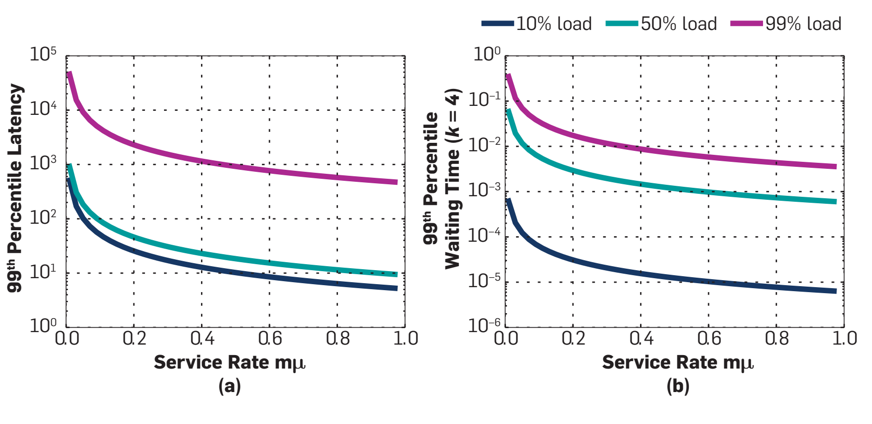
$$E[R] = \frac{C(k,\lambda/\mu)}{k\mu-\lambda}+\frac{1}{\mu}$$

Finally, the p-th percentile of queueing time is

$$w_p = -\frac{ln(1-p/(100C(k,\lambda/\mu)))}{k\mu-\lambda}$$

Figure 1b outlines how the 99th percentile of queueing time correlates to the service rate μ for one and four servers. Higher service rates correspond to less time spent by requests in the queue. We use the M/M/k model for analysis of system trade-offs unless otherwise specified. In the article's section on validation, we verify that this model closely reflects real system behavior. For applications with non-Poisson arrival- and service-time distributions, more general queueing models may be needed (such as the G/G/k model).[10,24] For more complex applications (such as multi-tier services), system architects would need a more sophisticated analytical model (such as a queueing network).

Figure 1. Building system insights from queueing theory: (a) 99th percentile response time in an M/M/1 model; and (b) 99th percentile queueing time in an M/M/4 model as a function of μ.



(see Figure 1 in the sidebar "Analytical Framework") and can provide first-order insights on how design decisions interact with tail latency. As was the case with the previous analyses based on Amdahl's Law, our model has significant implications for processor designs for cloud servers.

While analytical models help draw first-order insights, they run the risk of not accurately reflecting the complex operation of a real system. In Figure 2, we show a brief validation study of the queueing model, as discussed in the sidebar, with {1, 4, 8, 16} compute cores against a real instantiation of memcached, a popular in-memory, key-value store, with the same number of cores. We set the mean interarrival rate and service time of the queueing model based on the measured times with memcached. In both cases, when providing memcached with exponentially distributed input load, the memcached request latency is close to the one estimated by the queueing model across load levels.

Cost Model

Since hardware resources are not infinite, this analysis requires a cost model that relates resource usage to performance. We use a model similar to the one used by Hill and Marty[11] to extend Amdahl's Law to multicore chips. That is, we assume a given multicore chip is limited to R base core equivalents (BCE) units. This limitation represents area or power-consumption constraints in the chip design. The BCE is an abstract cost unit that captures processor resources and caches but does not share resources (such as interconnection networks and memory controllers). As in Hill and Marty,[11] we assume these resources are fairly constant in the system variations we examine. A baseline core that consumes 1BCE unit achieves performance of $perf(1)=1$. Chip architects can build more powerful cores by dedicating $r \in [1,R]$ resource units to each core to achieve performance $perf(r)$, where $perf(r)$ is the rate parameter μ in our performance model. Larger cores have higher service rate μ, which is inversely related to tail latency, as discussed in the sidebar. If performance increases superlinearly with resources, then more cores are always better. In practice $perf(r) < r$, creating trade-offs between opting for few brawny or many wimpy cores. By default, we follow Shekhar Borkar[3] and use $perf(r) = sqrt(r)$ but have also investigated how higher roots affect the corresponding insights.

Brawny Versus Wimpy Cores

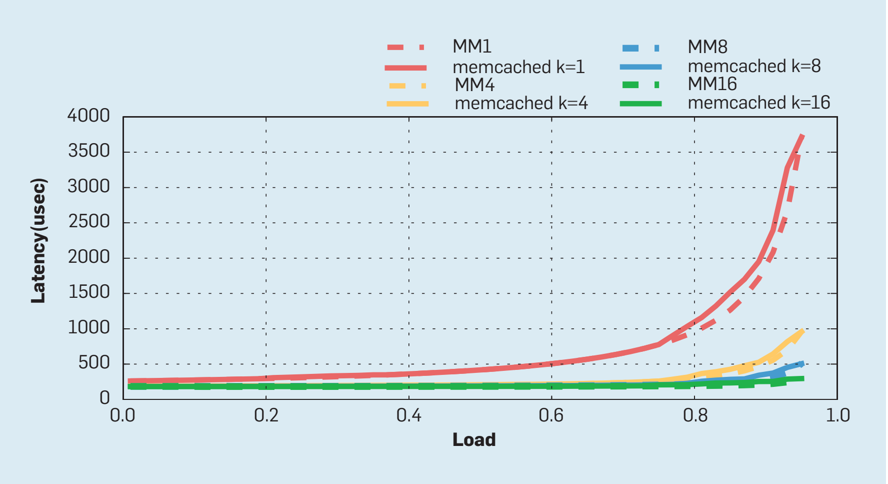
We first examine a system where all cores are homogeneous and have identical cost. An important question the designer must answer is: Given a constrained aggregate power or area budget, should architects build a few large cores or many small cores? The answer has been heavily debated in recent years in both academia and industry,[4,12,14,17,19,22] as it relates to the introduction of new designs (such as the ARM server chips and throughput processors like Xeon Phi).

Assuming the total budget is R = 100BCEs, an architect can build 100 basic cores of 1BCE each, 25 cores of 4BCEs each, one large core of 100BCEs, or in general R/U cores of U units each, as shown in Figure 3. We consider an online service workload with tail latency quality-of-service (QoS) constraints. QoS is defined as a function of the mean service time T_s of the 100BCE machine. For example, a very strict QoS target would require the 99th percentile of request latency to be T_s. This means the time between arrival and completion of 99% of requests must be less or equal to the machine's mean service time, allowing no tolerance for queueing or service-time variability. More relaxed QoS targets are defined as multiples of T_s: $QoS = \alpha T_s$, $\alpha \in [5, 10, 50, 100]$.

Figure 4a shows how throughput in queries per second (QPS) changes for different latency QoS targets, under the M/M/N queueing model described in the sidebar. Throughput of 100QPS for QoS=10Ts means the system achieved 100QPS for which the 99th latency percentile is $10T_s$. The x-axis captures the size of selected cores, moving from many small cores on the left side to a single core of 100BCEs on the right side. We examine all core sizes from 1BCE up to 100BCEs in increments of a single resource unit. In configurations with multiple cores, throughput is aggregated across all cores. The discontinuities in the graph are an artifact of the limited resource budget and homogeneous design; for example, for U = 51, an architect can build a single 51BCE core, while 49 resource units remain unused. Throughput for $10T_s$ for cores greater than 7BCE overlaps with 100Ts, as does throughput for $5T_s$ for cores of more than 12BCEs.

Finding 1. Very strict QoS targets put a lot of pressure on single-thread performance. When QoS = T_s or 5 T_s, cores smaller than 22BCEs or 12BCEs, respectively, achieve zero QPS for which the tail latency satisfies the QoS target. This happens because the cores are too weak to handle variability in service time even in the absence of queueing, and the queueing naturally occurs when cores operate close to saturation. This result means that, for services with extremely low-latency requirements (such as in-memory caching and in-memory distributed storage),[21] architects must focus on improving

Figure 2. Validation of the queueing model against a real instantiation of an in-memory key-value store (memcached) for {1,4,8,16} cores.

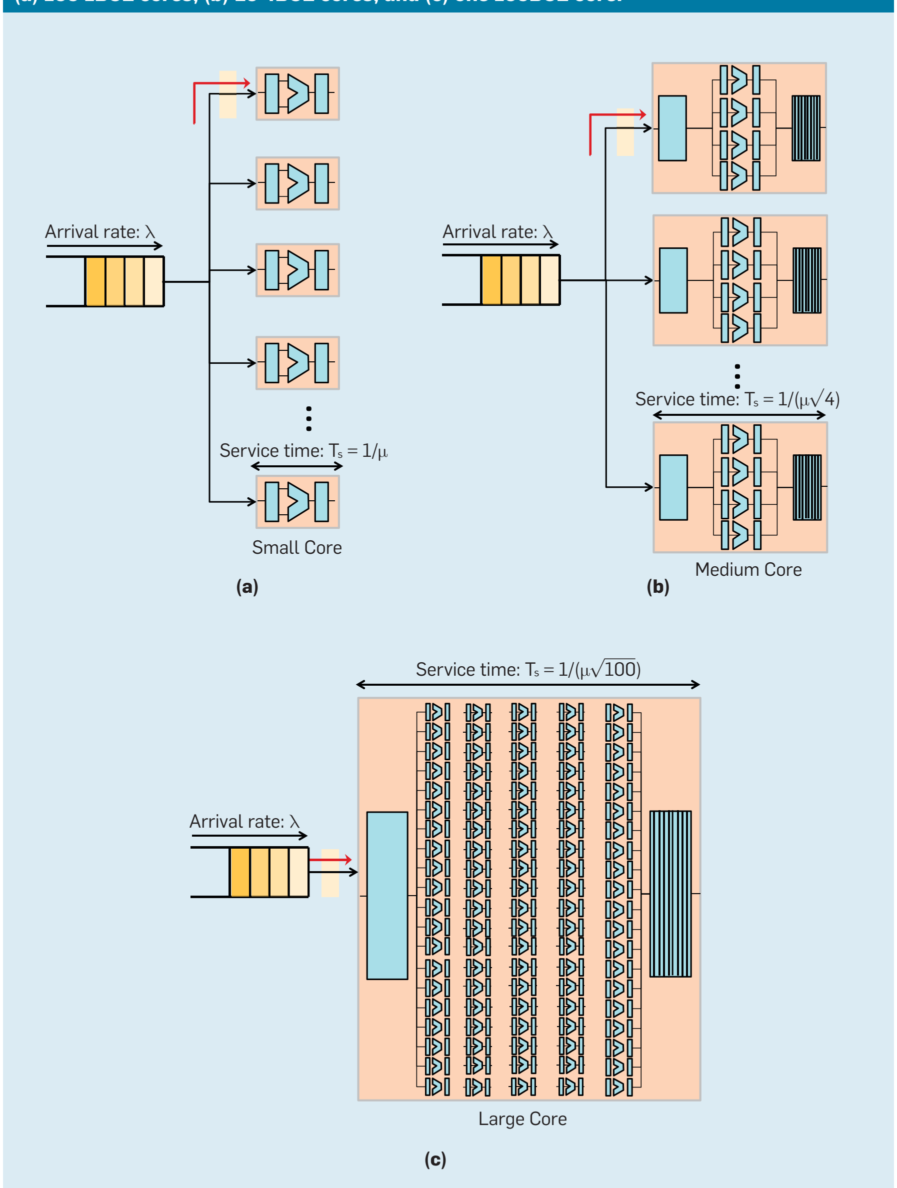


single-thread performance even at high cost. At the same time, some core parallelism is needed. A single 100BCE core performs significantly worse than four 25BCE cores. This finding is in agreement with industry concerns about the performance of small cores with warehouse-scale services.[12] The need for high single-thread performance also motivates application- or domain-specific accelerators as a more economical way of improving performance than incremental out-of-order core optimizations.

Finding 2. At lower latency constraints, architects should look for ways to balance optimizations for single-thread performance and request-level parallelism. At lower QoS targets, a larger set of medium-size cores achieves the best performance. For example, 7BCE cores are optimal for QoS = $10T_s$. For applications with moderate latency requirements (such as Web search and Web servers), architects should seek to balance improvements in single-thread performance (instruction-level parallelism) and multi-core performance (request-level parallelism). Increasing single-thread performance at high cost yields diminishing returns in this case. Nevertheless, a large pool of wimpy cores—1BCE—is optimal only when applications have no latency constraints, as with long data mining queries or log-processing requests. With QoS = $100T_s$, applications are essentially throughput-limited and perform best with many wimpy cores.

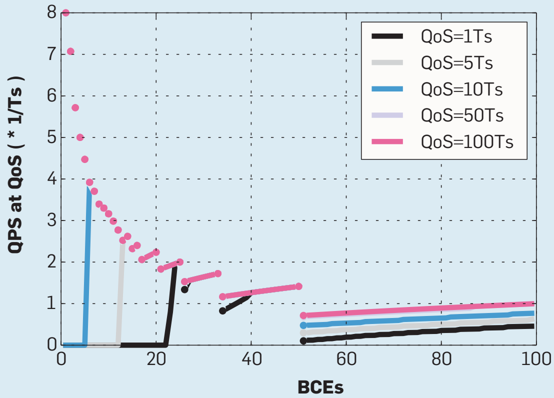
These findings highlight a disparity between optimal system design when optimizing for throughput versus when optimizing for tail latency. For example, in a homogeneous system where throughput is the only performance metric of interest and parallelism is plentiful, the smallest cores achieve the best performance; see the 1BCE cores in Figure 4a. In comparison, when optimizing for throughput under a tail latency constraint, the optimal design point shifts toward larger cores, unless the latency constraint relaxes significantly.

Figure 3. Homogeneous server configurations for a budget of $R = 100$ resource units: (a) 100 1BCE cores; (b) 25 4BCE cores; and (c) one 100BCE core.

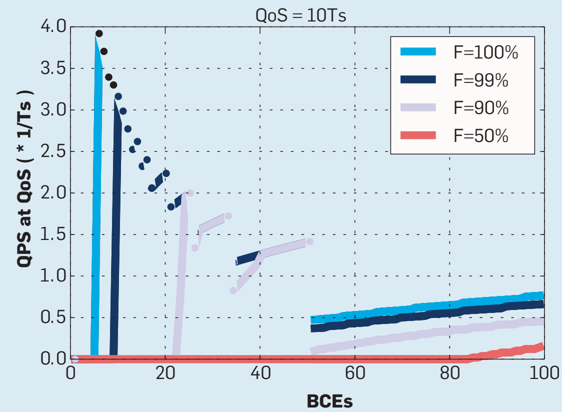


Finding 3. Limited parallelism also calls for more powerful cores. So far we have assumed all user requests are independent and perfectly parallelizable, though it is rarely the case in practice. Requests are often dependent on each other and on system issues like connection ordering and locks for writes causing serialization. The growing trend of breaking complex services down to smaller components (microservices) will only make the problem of request dependencies more common. This brings up the caveat of Amdahl's Law. To what extent are the previous findings accurate when parallelism is limited? Figure 4b shows the case of a reasonable QoS ($10T_s$) with $f \in \{50\%, 90\%, 99\%, 100\%\}$. When, for example, the parallel fraction of the computation f is 90%, 10% of requests are serialized. As a result, while optimal performance was previously achieved with seven BCE cores, the optimal core size now shifts to 25 BCEs. Limited parallelism also affects throughput-centric systems,[11] with more powerful cores outperforming wimpy cores in applications with serial regions. Using Hill's and Marty's model[11] with a 100BCE budget and 10% serialization, an architect would determine that 10BCE cores are optimal for throughput, a less aggressive increase in core size than when optimizing for latency. As parallelism decreases further, more performant cores are needed to drive down tail latency. When 50% of execution is serial, a single 100BCE core is optimal,

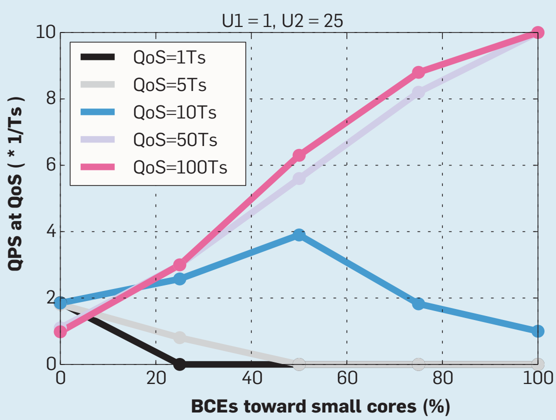
Figure 4. Studies on big versus small cores, core heterogeneity, and caching using the queueing model.



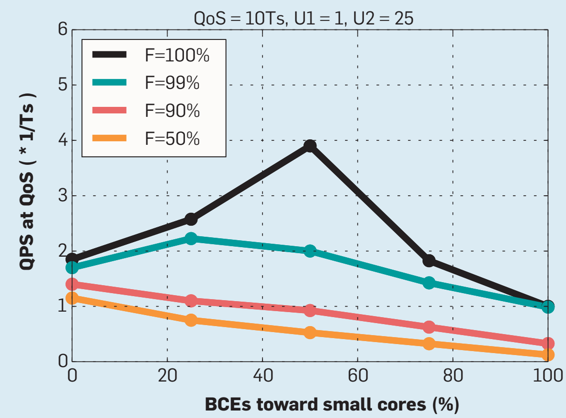
(a) Throughput (QPS) under a tail latency constraint as a system architect increases the resources per core when parallelism is unlimited;



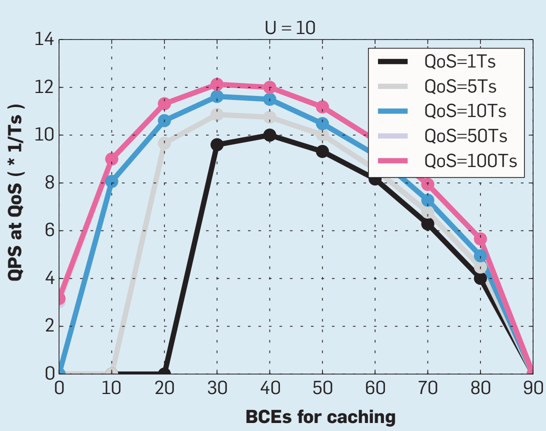
(b) Throughput under a tail latency constraint when parallelism is not plentiful;



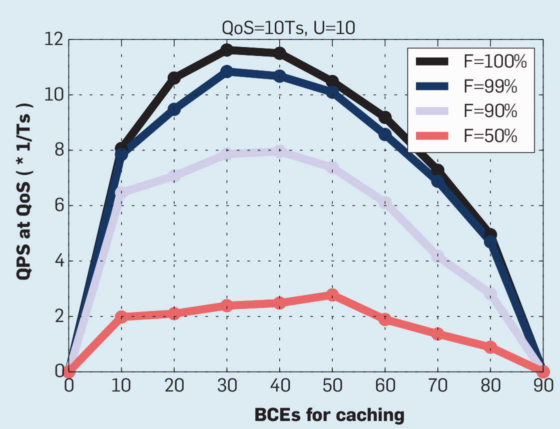
(c) Throughput (QPS) under a tail latency constraint as a system architect increases the resources for small cores (U1=1) under the assumption of unlimited parallelism;



(d) Throughput under a tail latency constraint when parallelism is limited;



(e) Throughput (QPS) under a tail latency constraint as a system architect increases resources for caching, as opposed to compute when parallelism is unlimited;



(f) Throughput under a tail latency constraint when parallelism is not plentiful.

a dramatic shift from the unlimited-parallelism case; overall throughput is also an order of magnitude lower. Quantifying the degree of parallelism in latency-critical services is essential when deciding how to build the underlying hardware. At the same time, computer scientists should strive to remove serialization across the system stack—at the application level by developing tracing and monitoring systems that detect and minimize cross-service dependencies, at the operating system by minimizing the need for lock serialization, and at the architecture level by investing in methods that increase single-thread performance and intra-request parallelism.[9]

These findings remain consistent for *perf* (r) scaling with the square, cubic,

Figure 5. Heterogeneous server configuration with 25BCE large cores and 1BCE small cores.

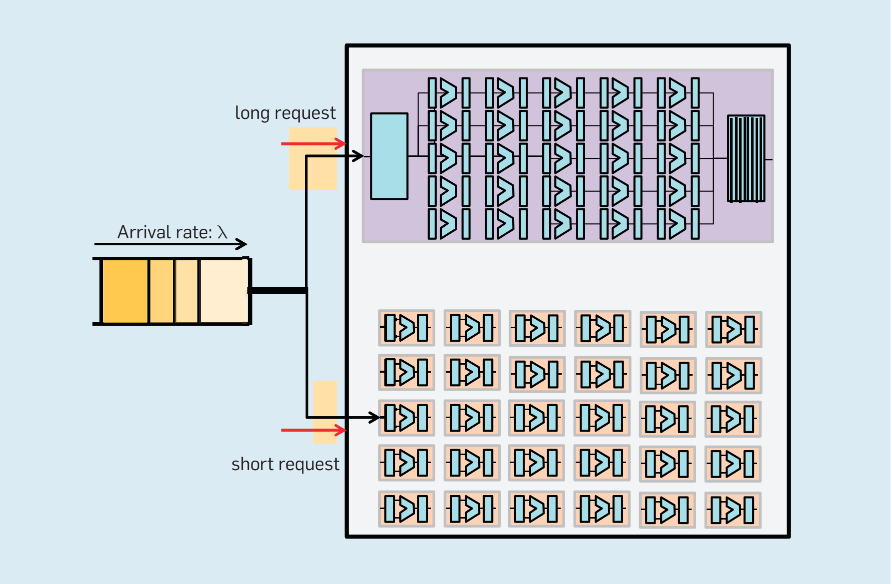
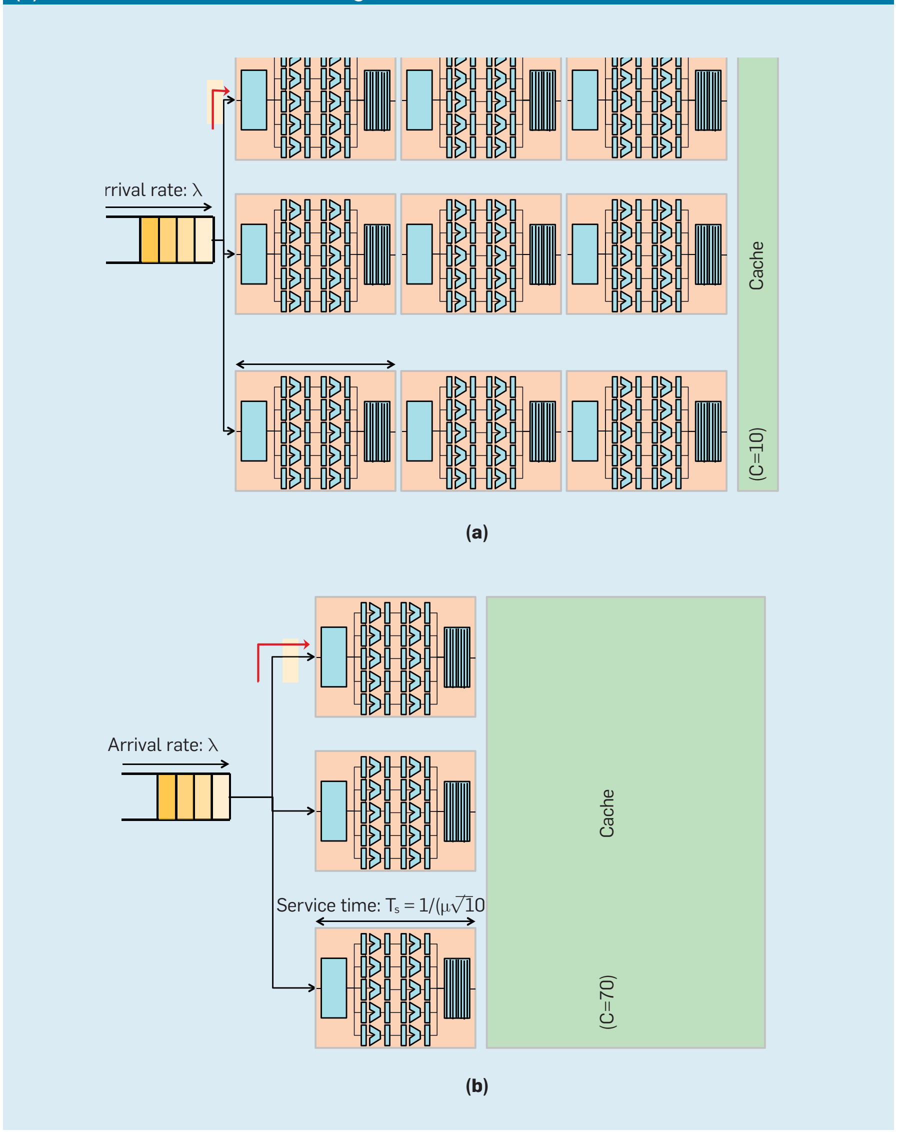


Figure 6. Server configurations with 10BCE cores when dedicating (a) 10 resource units and (b) 70 resource units toward caching.



and fourth root of r. Beyond that point, optimal design favors smaller cores.

Core Heterogeneity

The previous section explored the trade-offs between powerful, brawny cores and power-efficient, wimpy cores. Neither type of core provides high efficiency across a wide range of QoS targets, raising several obvious questions, including: Should an architect combine multiple core types in the same system, as is already the norm in multicore chips for mobile systems? How should architects determine the size of these cores? And at what ratio should they use them? Determining the right mix of large-versus-little cores, as well as devising schedulers that take advantage of heterogeneous cores, especially in the presence of heterogeneous load, has been a notably active topic of research in computer architecture in recent years.[5,9,15] Figure 4c shows the QPS under various QoS targets for a set of heterogeneous designs. In all cases, the system has two core configurations: small cores with $U = 1$, benefiting applications with relaxed QoS, and big cores with $U = 25$, benefiting applications with strict QoS. The system also receives two exponentially distributed input request streams, one with short and the other with long mean-service-time requests, and design a simple heterogeneity-aware scheduler that routes long requests to big cores and short requests to small cores. Requests are admitted to a single queue, as in Figure 5, and the ratio of long-to-short requests is, for now, 1:1. Figure 5 starts with all big cores at the leftmost point of the x-axis, explores the heterogeneous space, and ends with all small cores at the rightmost point.

Finding 4. Figure 4c captures a surprising trend. For strict QoS targets, like $1 \cdot T_s$, homogeneous systems with all big cores achieve optimal performance. In contrast, for very relaxed QoS targets, like $100T_s$, using all small cores achieves the best performance. However, for QoS targets in the middle (such as $10T_s$), heterogeneous systems, coupled with heterogeneity-aware schedulers, outperform their homogeneous counterparts. This result is especially true when the ratio of big to small cores matches the ratio of long-to-short requests. Varying the request ratio affects these findings significantly. The further away the ratio of long-to-short requests is from the ratio of big-to-small cores the more homogeneous systems outperform their heterogeneous counterparts. This result means that for heterogeneous architectures to make sense the system must closely track the input load and adjust to its changes, a common phenomenon in large-scale online services.[18]

Finding 5. We have again assumed unlimited request parallelism. Once serialization between requests is introduced, the optimal operation point shifts. Figure 4d shows QPS under various tail-latency QoS targets for increasing values of $f \in \{50\%, 90\%, 99\%, 100\%\}$. Where previously homogeneity outperformed heterogeneous designs for extreme QoS requirements—very strict and very relaxed—now takes the lead heterogeneity. For example, for a moderate QoS target of $10T_s$ and $f = 0.9$ a single big core achieves optimal performance, compared to the 50:50 mix in Figure 4c. In general, the more parallelism is limited the more the optimal operation point shifts left, with more big and fewer smaller cores. This is in agreement with Hill's and Marty's observations,[11] with the added implication that latency considerations cause a more rapid shift toward larger cores than when throughput is the only performance metric of interest. For example, when $f = 0.9$ and the system optimizes only for throughput, two 50BCE cores achieve the best performance under Hill's and Marty's model. As before, this result highlights the importance of quantifying the degree of parallelism in interactive applications. It also establishes that, even with limited parallelism, scheduling that takes into account the different capabilities of available hardware is essential for harnessing the potential of hardware heterogeneity.

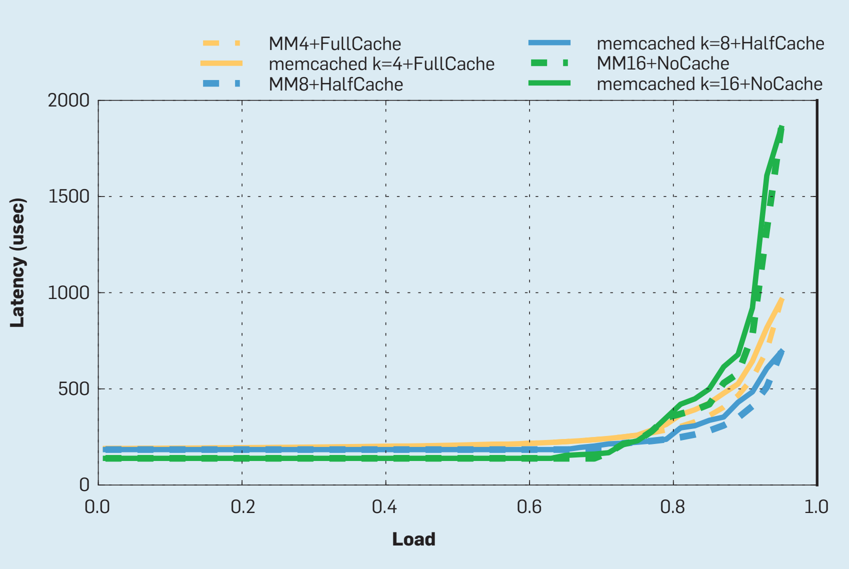
Caching

Architects constantly deal with the trade-off of using the limited resources for compute or caching. Larger caches help avoid the long latencies of main memory but draw significant static power and reduce the amount of resources available for compute cores; see Figure 6 for two characteristic configurations. Using the same total budget as before—$R = 100$—we explore how QPS under a tail-latency constraint changes as a fraction $C \in [0, 90]$ of resources goes toward building caches, as opposed to cores. We use 10BCE cores, benefitting applications with moderately strict QoS targets; Figure 4e shows this trade-off. On the leftmost point of the x-axis all resources are dedicated to building cores. On the rightmost point, 90% of resources go toward building caches and the remaining 10% toward building cores, one 10BCE core in this case. Increasing caching by 10BCE results in one fewer core in the system. We assume caches improve service time under a $sqrt(C)$ function, meaning $T_s0 = T_s = sqrt(C)$.[23] We validate the selection of the scaling factor against a real installation of memcached where the allocated last-level cache partition is adjusted using Intel's Cache Allocation Technology. As the number of used cores increases, the allocated cache capacity decreases. Figure 7 outlines that the difference between the analytical model and the real system is, in general, marginal. The findings reported in Figure 4e remain consistent for scaling functions until the seventh root of C, which corresponds to progressively lower benefits from caching, causing the optimal point to shift increasingly to the left.

Finding 6. For services with strict tail-latency requirements that exhibit locality, the benefit from caching is critical to achieving QoS. For strict QoS constraints (such as QoS = T_s), at least $C = 20$ units are needed to lower the core's service time in a way that achieves QPS under the tail-latency constraint.[16,20] Moderately increasing caching resources beyond $C = 20$ units further improves performance, as larger fractions of the working set fit in the last-level cache;[16] that is, more requests enjoy the shorter processing time of caches for the purpose of the queueing model. However, the benefits diminish beyond $C = 40$, and performance degrades rapidly as compute resources become insufficient.[16] Existing server chips dedicate one-third to one-half of their area budget to caches. Our analysis indicates this trend will continue.

Finding 7. For relaxed QoS targets, caching is less critical. Since smaller cores are sufficient for achieving the QoS constraints in this case, and although caching is still beneficial, moderate cache provisioning (such as $C = 10$ units to 30 units) yields most of its potential performance benefits. Increasing caching units to $C = 40$ has no effect on performance, and further increase degrades performance. Architects should focus instead on exploiting request parallelism in a way that keeps the large number of smaller cores busy.[12,16]

Figure 7. Validation of the queueing model against a real instantiation of an in-memory key-value store (memcached) with increasing caching and reduced compute resources.



Finding 8. Limited parallelism highlights the importance of increased caching. Figure 4f reports the performance for a moderate QoS target of $10T_s$ and increasing values of $f \in [50\%, 90\%, 99\%, 100\%]$. When 10% of the requests need to be serialized, the optimal point for caching is $C = 40$ units compared to $C = 30$ units with unlimited parallelism. Serialized execution requires higher single-thread performance, and larger on-chip caches is one way to achieve such performance.

Discussion

The models we offer here aim to provide first-order insight into how system design decisions affect tail latency and throughput in QoS-constrained services. These models do not capture every aspect of a data-center machine or application.[13] For example, while we can arbitrarily scale service times using the presented queueing model, system call and RPC overheads in real systems have hard lower limits. Likewise, software, especially in cloud applications, is not static. These frequent changes in cloud environments affect the degree of dependencies across requests, in terms of both the request fanout and the dependencies across components of a service (such as in microservices-based cloud applications). A more sophisticated model that captures such dependencies, potentially through a queueing network, can provide more accurate performance estimations at the cost of greater complexity. Finally, in hardware, architects cannot build cores with arbitrarily higher performance by simply adding more resources. They must also account for such factors as locality, coherence, and memory scheduling absent from our current model.

We see queueing theoretic models as a starting point for using queueing theory principles to draw insights into system design. We hope this analysis motivates researchers to develop more sophisticated models that address the limitations we have identified and, more important, the hardware and software that can achieve the performance requirements we highlighted.

Conclusion

Amdahl's Law is as pervasive when it comes to tail latency as it has been for traditional systems. Our goal here has been to offer a simple, intuitive, practical model that can lend first-order insights into which optimizations make sense when an application cares about tail performance. Using it, we have shown the overarching trade-offs in large-versus-small-core systems, heterogeneity, and caching. We encourage computer systems researchers to expand this model to express more sophisticated systems and studies.

Acknowledgments

We thank Mark Hill, Partha Ranganathan, Daniel Sanchez, and the anonymous reviewers for their helpful feedback on earlier drafts of this article.

References

1. Amdahl, G.M. Validity of the single-processor approach to achieving large-scale computing capabilities. In *Proceedings of the Spring Joint Computer Conference* (Atlantic City, NJ, Apr. 18–20). AFIPS ACM Press, New York, 1967, 483–485.
2. Barroso, L. and Hölzle, U. The case for energy-proportional computing. *Computer 40*, 12 (Dec. 2007), 33–37.
3. Borkar, S. Thousand-core chips: A technology perspective. In *Proceedings of the 44th Annual Design Automation Conference* (San Diego, CA, June 4–8). ACM Press, New York, 2007, 746–749.
4. Chen, S., Galon, S., Delimitrou, C., Manne, S., and Martinez, J.F. Workload characterization of interactive cloud services on big and small server platforms. In *Proceedings of the IEEE International Symposium on Workload Characterization* (Seattle, WA, Oct. 1–3). IEEE Press, 2017, 125–134.
5. Craeynest, K., Jaleel, J. et al. Scheduling heterogeneous multi-cores through performance impact estimation (pie). In *Proceedings of the 27th International Conference of the International Society for Computers and Their Applications* (Las Vegas, NV, Mar. 12–14). International Society for Computers and Their Applications, Winona, MN, 2012, 213–224.
6. Dean, J. and Barroso, L.A. The tail at scale. *Commun. ACM 56*, 2 (Feb. 2013), 74–80.
7. Delimitrou, C. and Kozyrakis, C. Paragon: QoS-aware scheduling for heterogeneous datacenters. In *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems* (Houston, TX, Mar. 16–20). ACM Press, New York, 2013.
8. Delimitrou, C. and Kozyrakis, C. Quasar: Resource-efficient and QoS-aware cluster management. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems* (Salt Lake City, UT, Mar. 1–5). ACM Press, New York, 2014.
9. Haque, Md. E., Eom, Y.h., He, Y., Elnikety, S., Bianchini, R., and McKinley, K.S. Few-to-many: Incremental parallelism for reducing tail latency in interactive services. In *Proceedings of the 20th International Conference on Architectural Support for Programming Languages and Operating Systems* (Istanbul, Turkey, Mar. 14–18). ACM Press, New York, 2015, 161–175.
10. Harchol-Balter, M. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action.* Cambridge University Press, Cambridge, U.K., 2013.
11. Hill, M. and Marty, M.R. Amdahl's Law in the multicore era. *IEEE Computer 41*, 7 (July 2008), 33–38.
12. Hölzle, U. Brawny cores still beat wimpy cores, most of the time. *IEEE Micro 30*, 4 (July-Aug. 2010), 20–24.
13. Kanev, S., Darago, J.P., Hazelwood, K., Ranganathan, P., Moseley, T., Wei, G.-Y., and Brooks, D. Profiling a warehouse-scale computer. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture* (Portland, OR, June 13–17). 2015, 158–169.
14. Khubaib, M., Suleman, A., Hashemi, M., Wilkerson, C., and Patt, Y.N. Morphcore: An energy-efficient microarchitecture for high-performance ILP and high-throughput TLP. In *Proceedings of the 45th Annual IEEE/ACM International Symposium on Microarchitecture* (Vancouver, B.C., Dec. 1–5). IEEE Computer Society, Washington, D.C., 2012, 305–316.
15. Li, J., Agrawal, K., Elnikety, S., He, Y., Lee, I-T.A., Lu, C., and McKinley, K.S. Work stealing for interactive services to meet target latency. In *Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (Barcelona, Spain, Mar. 12–16). ACM Press, New York, 2016, 1–13.
16. Li, S., Lim, H., Lee, V.W., Ahn, J.H., Kalia, A., Kaminsky, M., Andersen, D.G., Seongil, O., Lee, S., and Dubey, P. Architecting to achieve a billion requests per second throughput on a single key-value store server platform. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture* (Portland, OR, June 13–17). ACM Press, New York, 2015, 476–488.
17. Liang, X., Nguyen, M., and Che, H. Wimpy or brawny cores: A throughput perspective. *Journal of Parallel and Distributed Computing 73*, 10 (Oct. 2013), 1351–1361.
18. Meisner, D., Sadler, C.M., Barroso, L.A., Weber, W.D., and Wenisch, T.F. Power management of online data-intensive services. In *Proceedings of the 38th Annual International Symposium on Computer Architecture* (San Jose, CA, June 4–8). ACM Press, New York, 2011, 319–330.
19. Meisner, D. and Wenisch, T.F. Does low-power design imply energy efficiency for data centers? In *Proceedings of the 17th IEEE/ACM International Symposium on Low-Power Electronics and Design* (Fukuoka, Japan, Aug. 1–3). IEEE Press, Piscataway, NJ, 2011, 109–114.
20. Novakovic, S., Daglis, A., Bugnion, E., Falsafi, B., and Grot, B. Scale-Out NUMA. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems* (Salt Lake City, UT, Mar. 1–5). ACM Press, New York, 2014, 3–18.
21. Ousterhout, J., Agrawal, P. et al. The case for RAMClouds: Scalable high-performance storage entirely in DRAM. *SIGOPS Operating Systems Review 43*, 4 (Jan. 2010), 92–105.
22. Reddi, V.J., Lee, B.C., Chilimbi, T., and Vaid, K. Web search using mobile cores: Quantifying and mitigating the price of efficiency. In *Proceedings of the 37th IEEE/ACM International Symposium on Computer Architecture* (Saint-Malo, France, June 19–23). ACM Press, New York, 2010, 314–325.
23. Sprangle, E. and Carmean, D. Increasing processor performance by implementing deeper pipelines. In *Proceedings of the International Symposium on Computer Architecture* (Anchorage, AK, May 25–29). IEEE Press, 2002, 25–34.
24. Trivedi, K. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications, Second Edition.* John Wiley & Sons, Inc., New York, 2002.

Christina Delimitrou (delimitrou@cornell.edu) is an assistant professor and the John and Norma Balen Sesquicentennial Faculty Fellow in the Department of Electrical and Computer Engineering at Cornell University, Ithaca, NY, USA.

Christos Kozyrakis (kozyraki@stanford.edu) is a professor in the Departments of Electrical Engineering and Computer Science at Stanford University, Stanford, CA, USA.

DOI:10.1145/3208039

Online privacy is not just about what you disclose about yourself, it is also about what others disclose about you.

BY JOSE M. SUCH AND NATALIA CRIADO

Multiparty Privacy in Social Media

OVER TWO BILLION users consume social media to build and participate in online social networks (OSNs), uploading and sharing hundreds of billions of data items.[15] OSNs are not only huge in scale, they are predicted to keep growing in the coming years both in the number of users and in the amount of data users upload and share. The vast amount of data in social media is user-generated and personal most of the time, which clearly calls for appropriate privacy preservation mechanisms that allow users to benefit from social media while adequately protecting their personal information. Protecting users' privacy is not only essential to respect the Universal Declaration of Human Rights but also to serve as a first line of defense to mitigate cybercrime and other illegal activities that leverage the data obtained due to privacy breaches in social media, such as social phishing, identity theft, cyberstalking, and cyberbullying.

There have been many efforts devoted to study privacy in social media and how to protect users' personal

» key insights

- Multiparty privacy is an important problem in social media that also expands into other areas of social computing like cloud-based file sharing, collective intelligence, and wiki pages.
- Mainstream social media does not provide sufficiently adequate support for multiparty privacy and, as a result, users are forced to use different coping strategies that are far from optimal.
- There is an ongoing and growing body of multiparty privacy research, which we summarize and explore the limitations of in this article.
- We outline a research roadmap and a set of requirements for multiparty privacy tools.

CGI BY PETER CROWTHER ASSOCIATES; GROUP SHOT BY RAWPIXEL.COM

information since the very early days of social media, such as explored by Gross and Acquisti.[10] However, most of these efforts have focused on privacy from an individual point of view. For instance, advances include research[9] and industry[16] efforts on helping individual users better target their audience by modeling different relationships and social circles beyond the binary friendship model that is prevalent in most social media. While this has indeed helped to advance the state of the art on the topic, the problem of content affecting the privacy of more than one user at the same time has received little attention.

Privacy is not just about what you say or disclose about yourself. It is also about what others say or disclose about you. Evidence shows there are privacy boundaries collectively held and managed by individuals within relationships, families, groups, and organizations.[22] With the massive growth of social media, however, collectively held privacy boundaries have become extremely challenging to maintain, as many of the hundreds of billions of items uploaded are co-owned by multiple users,[14,15] yet mainstream social media only allow the user uploading a co-owned data item to set its privacy settings, which often leads to conflicts and severe privacy violations.[33,35] Multiparty privacy (MP) aims to facilitate the coordination of collectively held privacy boundaries by all individuals that co-own a data item online, as the privacy of all of them may be at stake depending on with whom the co-owned data item is shared.[a] MP particularly focuses on supporting the detection and resolution of multiparty privacy conflicts (MPCs), when individuals whose privacy may be affected by the same co-owned data item have conflicting privacy preferences. Take a simplified but illustrative example of MPC: Alice takes a photo of her and Bob. Mainstream social media would only allow Alice (assuming she uploads the photo) to set the privacy settings for the photo, but what if Bob would not like to share it with some of the friends Alice would like to share the

a MP is different from other collective approaches that focus on protecting just *one* individual.[4,6,21]

photo with? MP is concerned with not only photos but also other social media content such as posts, videos, comments, or events. Beyond social media, MP could also be useful in other social computing domains, in which information is co-created and co-owned by multiple users, so all these users should have a say on with whom this information is shared, such as collaborative software (for example, cloud-based collaborative documents), internal/external wiki pages, blogs, collective intelligence, crowdsourcing, among others.

Designing MP tools is a complex and difficult task, as users have different privacy attitudes and preferences; they socialize online with multiple types of relationships; and they share varying amounts of different types of content. In this article, we discuss the limited MP support users current have, the coping strategies users are forced to resort to in the absence of adequate MP support, and the latest developments in MP mechanisms and tools. Based on this, we outline a roadmap for future research with a set of requirements for developing MP tools.

Social Media Support for MP

Mainstream social media sites support some sort for MP, which mainly comes in the form of two mechanisms: tagging/untagging and reporting inappropriate content.

Tags are normally used to name people that appear in a photo with a link to their profile. People tagged in a photo can, however, untag themselves from the photo. There are some social media sites (like Facebook) in which you can opt-in to receive notifications about the photos you have been tagged in to approve tags before they become effective. Tagging/untagging represents some sort of MP, but it has three main limitations. The first limitation is that even if you untag yourself from a photo before anyone seeing it, this does not mean that your friends will not end up seeing the photo anyway. For instance, Alice and Bob are in a photo that Alice uploads to Facebook tagging Bob in it. Bob receives a notification, he revises the photo and decides not to approve the tag because he feels, for example, embarrassed about the photo. The point is that the photo, even without Bob being explicitly tagged, will be shared according to what Alice decides. That is, if Alice decides to share with her friends, and Alice and Bob share some friends, all these friends will be able to see the photo in Alice's wall anyway. The second limitation is that tagging/untagging is supported for photos but not for other items such as posts, comments, and events. Posts and comments do usually have the option to include *mentions* (using special symbols such as '@'), but these mentions are only controllable by the post/comment creator—though users can remove comments to their posts/photos. Finally, many users state that they feel very uncomfortable untagging themselves from photos because it may offend (from a social angle) the person who tagged them in the photo.[2]

Regarding reporting, most social media sites allow users to report when content published by others is not appropriate. This mechanism is mainly used to deal with *highly* inappropriate (or even illegal) content such as nudity, hate speech, violence, and other very serious offenses. After being reported, the provider decides unilaterally what to do with the content (delete it or not). Although this mechanism is of utmost importance to fight against these very serious offenses, it is not appropriate for all MP scenarios, as there are many cases in which privacy violations can happen without necessarily being related to these offenses. For instance, it may just be the case that you are not comfortable sharing some information with some other people, or you want to conceal information from your work colleagues. Also, it is important to highlight that reporting is only a *reactive* mechanism, which only activates after content has already been published and someone flags it as inappropriate. However, when the content is flagged, it may well be too late, the privacy violation may have already happened and the derived consequences may be unrecoverable, or other users may have been able to download the content and distribute it using other channels.

The problem of MP is starting to be recognized by mainstream social media as demonstrated by a recent revamp of Facebook's privacy controls.[b] In particular, Facebook's Privacy Basics now explains the newly introduced option to contact users about photos you do not like. The mechanism works as follows: if a user is tagged in a photo and she does not like the photo, she can now flag the photo as not liking it, which then opens up a message window containing a form with the recipient field set to the one who uploaded the photo, so that the user who does not like the photo can ask the user who uploaded it to remove it and include an optional reason for the removal. Although this is a step forward that very much recognizes the issue of MP, it still falls short because of multiple reasons, some of them also shared with tagging/untagging and reporting inappropriate content: a) the process happens once the photo has already been published, so any potential privacy breaches may have already occurred; b) it takes time to take down a photo that has already been published—for example, Lian et al.[19] calculated the time it takes for a photo URL to become unavailable after having deleted the photo from the social media site, which turned out to be three days on Instagram, seven days on Facebook, 14 days on Flickr and over 30 days in MySpace and Tumblr; c) it does not enable *collective* negotiation, as the photo may involve other people and not only the one who uploaded it and the one who complains about it; d) everything needs to be done manually, which introduces an unbearable burden on the users considering the large amount of friends users have online; and e) this mechanism has only been implemented for photos but not for other types of content such as posts, comments, and events.

User Coping Strategies for MP

As noted previously, there is a distinct lack of built-in capabilities in current social media infrastructures to help users compromise by actively negotiating with others.[40] Users are forced to communicate outside social media and apply a number of *coping strategies* to try to overcome or work around that lack of technical support. Basically, most of these coping strategies consist of actions or behaviors in the offline world that aim to prevent MPCs from happening online. Research uncovered several examples of these coping strategies, which very

b https://www.facebook.com/about/basics/how-others-interact-with-you/

much stress the need for MP tools. We discuss some examples of coping strategies and their shortcomings next (summarized in Table 1).

One of the offline strategies people employ before posting an item to avoid MPCs is trying to anticipate whether the item could be sensitive to anyone potentially affected by it.[18] For instance, if Alice and Bob appear together in a photo but Bob appears clearly inebriated, then it is likely that Alice may consider this by either not posting the photo or sharing it only with a restricted number of friends. However, this does not always work, as sometimes the person posting an item cannot anticipate the consequences this may have for others beforehand. An example is given in Lampinen et al.[18] where a person was congratulated by a friend about being accepted for a master's program via a comment, but the person had to quickly remove the comment as he had not yet told his employer about it and his employer was also friend of his online. Note even if the person removed the comment quickly, there was still the risk his employer may have already noticed the comment before it was removed.

Users sometimes ask the other co-owners of an item for approval before sharing it.[18] The problem with this strategy is that it is done offline without any technical means that could facilitate this. That is, one would need to ask permission offline to all people that may be affected by each and every item they upload. Also, when someone did not approve, they would need to negotiate a solution (for example, reduce the initial audience or decide not to upload). This would quickly become an unbearable burden on users.

It has also been observed that teens cloak their messages and share photos with inside jokes.[3] For instance, Boyd and Marwick[3] report an example of a girl writing a post on Facebook about something she knew only her close friends would understand, as she wanted to prevent other friends from knowing what she actually meant. The downside of this strategy is that it clearly does not scale and may not be feasible for all photos or other types of items that people would like to share. For example, a photo about your travel to Mauritius cannot be easily cloaked in case you want to share it with some people but not with others.

As social media proves inadequate to manage disclosures in MP scenarios, some users switch media to share content using other technologies such as cloud-based file sharing, instant messaging, or email attachments.[2] This has the advantage of protecting not only their own content but also limiting the privacy risks for others. There are, however, three main disadvantages as well. Firstly, this may be possible for photos, videos, and so on, but not for other types of content such as events or comments. Secondly, users cannot control which technologies their friends use; that is, their friends could still upload photos using social media without users being able to do anything about it. Thirdly, these technologies might also lead to MPCs. For instance, one user may share a video in a Whatsapp group in which there are people with whom other users in the video would not like to share it.

Users also confirmed that, in the absence of better ways to manage MP situations, they actually change and tightly control their offline behavior. For example, people behave in a different way when they see a camera around.[2,18] If you know a friend likes to take photos and posts them very often, you may decide not to hang out with her to avoid any undesired photos being posted. This highlights the extent to which people feel unable to participate in MP decisions. The effectiveness of this strategy is again very limited, mainly due to the pervasiveness of smartphones and wearable devices, being always alert and constantly modifying your offline behavior is infeasible.

One of the most interesting strategies perhaps is that users collectively negotiate and achieve offline agreements and compromises about what gets posted and to whom it gets shared.[2,18,40] For instance, a group of friends could agree the photos they take in a trip can only be shared among them or with close friends of them. Interestingly, it turns out users are always very open to consider and accommodate others' preferences as much as possible.[18,40] In addition, research uncovered that users do not want to cause any deliberate harm to their friends and will normally listen to reasonable objections, which also acts as a way of reaffirming and reciprocating relationships.[40] The main problem with this strategy, as with many of the other strategies seen so far, is that it does not scale. It is impossible for users to be constantly negotiating with hundreds of friends about hundreds of photos without technical aid.

Research on MP Tools

It seems clear considering all the cases noted here that users actively seek to work around the problem of not having adequate technical support for MP. However, the effectiveness of the coping strategies they use for this seems rather limited according to the drawbacks these strategies have. This has inspired researchers to design interfaces and computational methods that empower users to collectively manage MP in more effective and efficient ways than the current coping strategies they are forced to resort to today. Although research in this area is still in its infancy, there have been a number of proposals that we categorize below into

Table 1. Examples of coping strategies.

Strategy	Main Drawbacks
Try to anticipate consequences for others[18]	Impossible to always anticipate privacy consequences.
Seek approval before posting[18]	Too much burden on the user that uploads the item.
Inside jokes and cloaking[3]	It does not scale and it is not feasible for some types of content.
Alternative sharing media[2]	MPCs can happen in other media too. Also, one user cannot control which media others use to share content.
Change offline behavior when cameras around[2,18]	Very difficult due to the pervasiveness of smartphones and wearables.
Negotiation of a shared policy with other users[2,18,40]	It could easily become a burden on the users due to the amount of co-owned content.

five main approaches (summarized in Table 2), highlighting their strengths and limitations. Note that other works in addition to those discussed have also been published but we could not include all of them due to the space and maximum references allowed, and have instead included those we considered the most representative of each approach.

Manual approaches. The first research stream proposed support for MP by helping users to identify where MPCs can or did occur.[2,39] For instance, Wishart et al.[39] present a way to specify strong and weak sharing preferences so that these preferences could be inspected to find conflicts. Also, Besmer et al.[2] introduce a system whereby users tagged in a photo can contact the user who uploaded the photo to ask to remove it or to restrict the audience of the photo, which resembles the functionality Facebook introduced some time later.[7] While these approaches represented a stepping-stone, recognized the problem of MP, and proposed a partial solution to it, they left all the negotiation process to resolve detected conflicts to happen without any particular technical aid. That is, users must resolve every potential MPC in a *manual* way, which may become an unbearable burden considering the massive amount of content uploaded and the number of friends that users have in social media.

Auction-based approaches. Another research stream proposed solving potential MPCs using a bidding mechanism.[30] Users bid for the sharing decision they would prefer the most and the winning bid determines the sharing decision that will be taken for a particular item. These approaches were the first ones to consider a semiautomated method to aid users in collectively defining a sharing decision—for example, the outcome of the auction is computed automatically from the bids users specify. However, users may have difficulties comprehending the mechanism and specifying appropriate bid values in auctions, and users are required to bid for each and every item co-owned with others.

Aggregation-based approaches. These approaches suggest a solution to a MPC by aggregating the individual privacy preferences of all users involved. They can be abstractly conceptualized as voting mechanisms, where the preferences of each user affected by an item count as one vote (sometimes weighted) for sharing/not sharing. Then, a voting rule models how each of these mechanisms aggregates votes together. For instance, in majority voting,[5] the preference of the majority of users is taken as the decision to be applied to the content. Another example would be veto voting,[35] so that if there is one of the users affected by the content who opposes sharing, then the content is not shared. The main problem with these approaches is that they always aggregate preferences in the very same way. For instance, using majority voting always means that even when content can be very sensitive and lead to privacy violations for one user, it will be shared if the majority of users wishes to. In contrast, always using veto voting may be too restrictive and impact the known benefits users get from sharing in social media.[29] Subsequent works[12] recognize this issue and consider more than one way of aggregating user preferences. However, it is up to the one who uploads the item to decide the aggregation method to apply. This requires the user who uploads the item to anticipate the consequences for others, which may be a very difficult task as discussed earlier, and it may not always render the optimal solution.

Adaptive approaches. These approaches automatically *infer* the best way to solve a MPC based on the particular situation.[32] These approaches model a situation considering factors such as the individual preferences of each user, the sensitivity of the content, or the relationships to the potential audience. Then, a particular situation instantiates particular *concessions* that are known to happen when people negotiate offline an agreement about sharing co-owned items.[2,18,40] Thus, these approaches automatically *adapt* to the situation at hand, turning as restrictive as veto voting if the situation requires so (for example, if the item is very sensitive), or suggesting sharing in other situations (for example, someone having special interest in sharing and the others not caring much about it). While these approaches capture the known situations of when concessions happen during offline negotiations, it is difficult to model all possible situations, and they

Table 2. Summary of MP approaches with example references.

Approach	Short Description	Main Drawbacks
Manual[2,39]	Users are provided with a way of detecting MPCs, and they can manually resolve them when detected.	It may easily become a burden on the users, as they do not provide automated support for conflict resolution.
Auction-based[30]	Users gain fictitious money they can invest in auctions bidding for the most desired sharing decision for co-owned items.	Users may have difficulties to understand and manage the process appropriately.
Aggregation-based[5,12,35]	Individual privacy preferences of all users are aggregated using a rule or set of rules to produce a joint sharing decision.	Individual privacy preferences are aggregated in the same way or the uploader chooses the method to aggregate.
Adaptive[32]	Different situations are modeled based on a number of factors and a different sharing decision is suggested depending on the situation.	It is difficult to model all possible factors that determine a situation and the best method to achieve an optimal sharing decision.
Game-theoretic[13,17,25,31]	Users or automated software agents negotiate a solution following an established protocol. Both the protocol and the negotiation strategies are analyzed using game-theoretic solution concepts.	Users' behavior in social media seems not to be perfectly rational as there are many very social idiosyncrasies that play a role in MP.
Fine-grained[14,36]	Users define individualized access control decisions over personally identifying objects within a photo, for example, users deciding whether or not their faces are blurred.	Blurring objects (for example, faces) within a photo may not be the optimal solution in terms of the utility of the information shared and/or protecting users' privacy.

may not capture opportunistic concessions or agreements that may arise in potentially unknown situations.

Game-theoretic approaches. Another approach has been to define negotiation protocols, which are a means of standardizing the communication between participants in the process of negotiating a solution to a MPC by defining how the participants can interact with each other. These protocols are then enacted by users manually[15] or automatically by software agents[17,31] to negotiate an agreed sharing decision for a particular item. Participants can follow different strategies when enacting the negotiation protocols, and these strategies are analyzed using well-known game-theoretic solution concepts such as the Nash equilibrium. This allows, for instance, to determine analytically which are the best strategies that participants can play as well as to find strategies that are stable (strategies in which no participant has anything to gain by changing only her own strategy unilaterally). While these proposals provided elegant frameworks from a formal point of view and build upon well-studied analytic tools, they may not work well when used in practice.[13] This is because users' behavior does not seem perfectly rational in practice (as assumed in these approaches), and even if some are starting to consider other factors like reciprocity[17] and social pressure,[25] they are still far from considering the many very social idiosyncrasies that play a role in MP.[18,40]

Fine-grained approaches. The last research stream focuses on preventing MPCs by allowing each user in a photo to independently decide whether some personally identifying objects within the photo are shown or blurred.[14,36] In particular, one of the first works in this approach allowed users to individually decide whether their face is shown or blurred.[14] The process works as follows: the users in a photo are identified using face recognition algorithms such as Facebook's DeepFace algorithm;[34] the users recognized are notified and they can suggest the list of friends who can have access to the photo; and when a user wants to access a photo, she will only see the faces of the users that have granted access to her and the other faces in the photo will appear blurred. However, blurring faces (or other objects in a photo) may impact the utility of the photo being shared, negatively impacting the benefits people get by sharing in social media,[29] and there is also the risk that a person can be reidentified even if her face (or other objects in a photo) has been blurred.[23] Hence, when a collaboratively agreed solution to a MPC is possible, that solution might be more desirable than enforcing access separately, as the photo will not lose any utility (no object blurred), but the audience of the photo will be negotiated to remove access to any undesired people.

Requirements For MP Tools

Building upon the previous analysis on existing approaches and their limitations, we now outline a set of requirements to develop MP tools that empower users to collectively manage their privacy together with others and overcome these limitations. These tools would aid end users to identify potential MPCs and, when MPCs are identified, provide support for their resolution (for example, in the form of recommendations), allowing an appropriate "*boundary regulation process by actively negotiating one's boundaries with others*."[40] Next, we describe each of the requirements in detail.

Design informed by real-world empirical data. None of the existing approaches are grounded in a deep understanding of MPCs and their optimal solution in practice. This is in part due to not having enough empirical evidence about MPCs yet. Such an empirical base is utterly essential to inform the design of MP tools that overcome the limitations identified in the existing literature. As mentioned, researchers have shed light on how users are forced *online* to resort to coping strategies to work around the lack of appropriate support for MP,[2,3,18,40] and there is evidence of how collectively held privacy boundaries are managed *offline*.[22] While this previous research already provides a very good foundation to build upon, further research is needed to better understand when and how often MPCs actually happen online and, more importantly, when they become a problem or lead to potential privacy violations and hence need a solution. Particular instances of MPCs users faced could be studied to understand whether they happened despite coping strategies being used, how users came up or would come up with the optimal solution for the MPCs studied, and the factors that played a role in the process. Some very recent research goes in this direction,[33] having contributed the first empirical and public dataset of MPCs. Having this empirical base about MPCs would ultimately underpin a thorough understanding of MPCs and the nuanced factors that affect them from the ground up, which could then be used as the basis to design MP tools that offer support to different types of users, social groups, and relationships and can recommend optimal solutions to MPCs. Recent efforts on privacy engineering should be leveraged to easy the challenging task of going from empirical evidence to privacy design.[11]

User-centric MP controls. The main challenge here is how to develop usable MP tools in line with the empirical base mentioned earlier, so users could effectively manage MP with minimal effort. However, MP tools should aim for usability without becoming a *fully* automated solution, as this may not achieve satisfactory results when it comes to privacy in social media. Instead, users may have to provide some input into MP tools, which will then provide a *recommendation*, as very recent research has shown that the optimal solution for an MP conflict could be predicted given some input from the users, like the reason for their preferred privacy policy.[8] However, if users have to intervene to express their individual privacy preferences and/or to accept/decline the solution recommended for each and every co-owned item and potential conflict, would this not easily become a burden on the users? How do we find adequate trade-offs between intervention and automation? There are previous studies on individual privacy in social media that could help: Tools like AudienceView[20] could be used to show and/or modify the suggested solution or express individual preferences; approaches similar to Fang et al.[7] could be used to *learn* the way users respond to MP over time; and, approaches like Watson et al.[38] could be used to create suitable defaults for MP settings.

Scaled-up and comparable evaluations. The existing approaches for MP presented here were either not evaluated empirically with users,[5,17,25,30,31,39] or the user studies conducted were low-scale with at most 50 participants.[2,12–14,32,36] This is in part due to a distinct lack of systematic and repeatable methods and/or protocols to evaluate MP tools and compare them to each other. In order for evaluations to be more conclusive and generalizable, MP tools should be evaluated considering wider and more varied populations. Also, evaluation protocols should be developed with a view to maximize *ecological validity*, which is particularly challenging in this domain. Firstly, participants in user studies would always seem reluctant to share sensitive information with researchers[37] (for example, photos they feel embarrassed about and prefer not sharing online), which would bias any evaluations toward non-sensitive issues only, leaving out the scenarios where the adequate performance of MP tools would be critical. An alternative could be evaluations with fake data/scenarios where participants self-report how they would behave, but the results may not match participants' actual behavior in practice due to the well-known dichotomy between privacy attitudes and behavior.[1] Secondly, conducting MP evaluations *in the wild* is very difficult, as it would require all the users affected by a particular piece of content to be studied together to understand the conflicts and whether the solutions to the conflicts are optimal. A possible way forward could be methodologies based on *living labs*, which would integrate and validate research in evolving real-life contexts.

Privacy-enhanced party recognition. Given a particular item uploaded, MP tools should derive the users who are affected by the item. For instance, if a user uploads a photo and tags in it all the other users that appear in the photo, MP tools can directly use this to know which users are involved. However, users many times either do not tag all people clearly identifiable in a photo or incorrectly tag people who actually do not appear in the photo. Face recognition software could be used for this, such as the one developed by Facebook researchers called DeepFace,[34] which has 97.35% accuracy. The question that arises is whether using face recognition software could be too privacy invasive for individuals, that is, the social media provider would be able to identify individuals in any photo even for photos outside the social media infrastructure, or individuals could be misidentified and wrongly associated with items that are not relevant to them (note even if accuracy of face recognition is high and false positives are low, the number of items and users is huge). Interestingly, this seems to open a completely new and exciting type of privacy-related trade-off compared to the well-known privacy-utility trade-off, which would be multiparty vs. individual privacy. Note, however, that a multiparty-individual privacy trade-off will not be needed if privacy-preserving face-recognition methods[27] are used by MP tools, so that parties would be recognized while preserving their privacy. Beyond photos, party recognition may be easier for some content type such as events (people invited or attending are explicitly mentioned) or even more challenging for some other content such as text posts, in which affected users may not always be explicitly tagged.

MP tools should aim for usability without becoming a *fully* automated solution, as this may not achieve satisfactory results when it comes to privacy in social media.

Support for inferential privacy. Another issue not considered before in a MP context is that of inferential privacy. That is, it may not only be about what your friends say about you online, but also what it may be inferred from what they said regardless of the type of content. For instance, Sarigol et al.[27] have demonstrated the feasibility of constructing shadow profiles of sexual orientation for users and non-users, using data from more than three million accounts of a single OSN. Note that negotiations or agreements for the case of inferential privacy may be more complex, as the reasons not to publish content may not be about the content itself but more about the consequences in terms of the information that may be inferred from it, so solutions to this type of MPC might be more difficult to comprehend by users, which would also challenge the usability and understandability of MP tools. Also, we are unaware of any social media site that provides users with any sort of controls for inferential privacy; let alone any research conducted that considers both MP and inferential privacy together.

Privacy-preservation guarantees. Last but not least, MP tools should provide some sort of individual privacy guarantees. This is particularly important when a multiparty agreement is not possible. For instance, a user may be posting on purpose content that defames another user. In these cases, there may be room for enforcing individual privacy preferences to some extent. For instance, a possible solution for photos is the work by Ilia et al.,[14] which would allow users to control whether their face is shown or blurred in a particular photo. This seems an appropriate solution when a MP conflict arises and no agreement is found by the users affected, so instead of the *winner taking it all*, the outcome is that all users affected are guaranteed their individual privacy to some extent. This, however, does not completely remove the identification risks, as acknowledged by Ilia[14] because there is still the chance the user may be recognized even after her face has been blurred,[26] and approaches that are able to remove the full body of a person and reconstruct the image are still not there, though there are approaches that already recognize user's body/gesture.[28]

Conclusion

Multiparty privacy (MP) is an important problem in social media that also expands into other areas of social computing where there is co-owned information such as blogs, collective intelligence, wiki pages, cloud-based file sharing,[24] and collaborative documents, which have received even less attention when compared to social media for this matter. As highlighted in this article, mainstream social media does not provide adequate support for MP and, as a result, users are forced to use different coping strategies that are far from optimal. Thus, there is a need for the development of novel privacy-enhancing techniques and mechanisms to help users to manage MP. We still have a long way to go to make such mechanisms a reality and embed them in highly usable tools ready to be utilized by end users, partly due to the complex nature of MP and social behaviors, which requires an interdisciplinary approach to MP. In this article, we have introduced the area of MP tools, discussed its current state and advances, and defined a set of requirements to shape the agenda for future research in this area.

Acknowledgments

Thanks to the EPSRC for supporting this research under grant EP/M027805/1, as well as to William Aiello and the anonymous reviewers for their very useful and helpful comments on a previous version of this manuscript.

References

1. Acquisti, A. and Gross, R. Imagined communities: Awareness, information sharing, and privacy on the Facebook. *Privacy enhancing technologies PET* (Jan. 2006), 36–58.
2. Besmer, A. and Richter Lipford, H. Moving beyond untagging: Photo privacy in a tagged world. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (2010), 1563–1572.
3. Boyd, D. and Marwick, A. Social steganography: Privacy in networked publics. International Communication Association, 2011, Boston, MA.
4. Calikli, G. et al. Privacy Dynamics: Learning privacy norms for social software. In *Proceedings of the 2016 IEEE/ACM 11th Intern'l Symposium on Software Engineering for Adaptive and Self-Managing Systems.* ACM, 47–56.
5. Carminati, B. and Ferrari, E. Collaborative access control in online social networks. *IEEE CollaborateCom*, (2011), 231–240.
6. Criado, N. and Such, J.M. (2015). Implicit contextual integrity in online social networks. *Information Sciences 325* (2015), 48–69.
7. Fang, L. and LeFevre, K. Privacy wizards for social networking sites. In *Proceedings of the 19th Intern'l Conference on World Wide Web.* ACM, 2010.
8. Fogues, R.L., Murukannaiah, P.K., Such, J.M. and Singh, M.P. Sharing policies in multiuser privacy scenarios: Incorporating context, preferences, and arguments in decision-making. *ACM Trans. Computer-Human Interaction 24*, 1 (2017), 5.
9. Fong, P.W. Relationship-based access control: protection model and policy language. In *Proceedings of the 1st ACM Conference on Data and Application Security and Privacy* (2011), 191–202.
10. Gross, R. and Acquisti, A. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, 71–80.
11. Gürses, S., and del Alamo, J. M. (2016). Privacy engineering: Shaping an emerging field of research and practice. *IEEE Security & Privacy 14*(2), 40-46.
12. Hu, H., Ahn, G.J., and Jorgensen, J. Multiparty access control for online social networks: Model and mechanisms. *IEEE Transactions on Knowledge and Data Engineering 25*, 7 (2013), 1614–1627.
13. Hu, H., Ahn, G.J., Zhao, Z. and Yang, D. (2014). Game theoretic analysis of multiparty access control in online social networks. In *Proceedings of the 19th ACM Symposium on Access Control Models and Technologies* (2014), 93–102.
14. Ilia, P., Polakis, I., Athanasopoulos, E., Maggi, F., and Ioannidis, S. Face/Off: Preventing privacy leakage from photos in social networks. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security*, (2015)
15. Internet.org. A focus on efficiency, 2014; http://internet.org/efficiencypaper/
16. Kairam, S., Brzozowski, M., Huffaker, D. and Chi, E. (2012). Talking in circles: selective sharing in google+. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), 1065–1074. ACM.
17. Keküllüoğlu, D., Kökciyan, N. and Yolum, P. Strategies for privacy negotiation in online social networks. In *Proceedings of the ACM International Workshop on AI for Privacy and Security* (2016), 2.
18. Lampinen, A., Lehtinen, V., Lehmuskallio, A. and Tamminen, S. We're in it together: Interpersonal management of disclosure in social network services. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (2011), 3217–3226. ACM.
19. Liang, K., Liu, J. K., Lu, R., and Wong, D.S. Privacy concerns for photo sharing in online social networks. *IEEE Internet Computing 19*, 2 (2015), 58–63.
20. Lipford, H. R., Besmer, A. and Watson, J. Understanding privacy settings in Facebook with an audience view. *UPSEC*, (2008), 1–8. USENIX.
21. Nissenbaum, H. Privacy as contextual integrity. *Wash. L. Rev. 79* (2004), 119.
22. Petronio, S. *Boundaries of Privacy: Dialectics of Disclosure.* Suny Press, 2012.
23. Polakis, I. et al. Faces in the distorting mirror: Revisiting photo-based social authentication. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Nov. 2014), 501–512.
24. Ramokapane, K.M., Rashid, A., and Such, J.M. 'I feel stupid I can't delete...:' A study of users' cloud deletion practices and coping strategies. *USENIX Symposium on Usable Privacy and Security*, (2017), 241–256.
25. Rajtmajer, S., Squicciarini, A., Such, J.M., Semonsen, J. and Belmonte, A. An ultimatum game model for the evolution of privacy in jointly managed content. In *Proceedings of the 2017 International Conference on Decision and Game Theory for Security*, 112¬–130.
26. Sadeghi, A.R., Schneider, T., and Wehrenberg, I. Efficient privacy-preserving face recognition. In *Proceedings of Information, Security and Cryptology*. Springer Berlin Heidelberg, 2010, 229–244.
27. Sarigol, E., Garcia, D., and Schweitzer, F. Online privacy as a collective phenomenon. In *Proceedings of the 2nd ACM Conference on Online Social Networks* (Oct. 2014), 95–106.
28. Shotton, J. et al. Real-time human pose recognition in parts from single depth images. *Commun. ACM 56*, 1 (Jan. 2013), 116–124.
29. Spiekermann, S., Krasnova, H., Koroleva, K. and Hildebrand, T. Online social networks: Why we disclose. *J. Information Technology 25*, 2 2010), 109-125.
30. Squicciarini, A.C., Shehab, M., and Paci, F. Collective privacy management in social networks. In *Proceedings of ACM's 18th International Conference on World Wide Web* (Apr. 2009), 521–530.
31. Such, J.M. and Rovatsos, M. Privacy policy negotiation in social media. *ACM Tran. Autonomous and Adaptive Systems 11*, 1 (2016), 1–29.
32. Such, J.M. and Criado, N. Resolving multiparty privacy conflicts in social media. *IEEE Trans. Knowledge and Data Engineering 28*, 7 (2016), 1851–1863.
33. Such, J.M., Porter, J., Preibusch, S. and Joinson, A. Photo privacy conflicts in social media: A large-scale empirical study. In *Proceedings of the 2017 ACM CHI Conference on Human Factors in Computing Systems*, 3821–3832.
34. Taigman, Y., Yang, M., Ranzato, M.A., and Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1701–1708).
35. Thomas, K., Grier, C., and Nicol, D.M. Unfriendly: Multiparty privacy risks in social networks. *Privacy Enhancing Technologies.* Springer Berlin Heidelberg, 2010, 246-252.
36. Vishwamitra, N., Li, Y., Wang, K., Hu, H., Caine, K., and Ahn, G.J. Towards Pi-based multiparty access control for photo sharing in online social networks. In *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies* (2017), 155–166.
37. Wang, Y., Norcie, G., Komanduri, S., Acquisti, A., Leon, P.G. and Cranor, L.F. I regretted the minute I pressed share: A qualitative study of regrets on Facebook. In *Proceedings of the 7th Symposium on Usable Privacy and Security* (2011), 10.
38. Watson, J., Lipford, H.R. and Besmer, A. Mapping user preference to privacy default settings. *ACM Trans. Computer-Human Interaction, 22*, 6 (2015), 32.
39. Wishart, R., Corapi, D., Marinovic, S., and Sloman, M. (2010). Collaborative privacy policy authoring in a social networking context. In IEEE international symposium on Policies for distributed systems and networks (POLICY) (pp. 1-8).
40. Wisniewski, P., Lipford, H., and Wilson, D. (2012). Fighting for my space: Coping mechanisms for SNS boundary regulation. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (pp. 609-618).

Jose M. Such (jose.such@kcl.ac.uk) is an associate professor in the Department of Informatics, Faculty of Natural and Mathematical Sciences, at King's College London, U.K.

Natalia Criado (natalia.criado@kcl.ac.uk) is an assistant professor in the Department of Informatics, Faculty of Natural and Mathematical Sciences, at King's College London, U.K.

Watch the authors discuss their work in this exclusive *Communications* video. https://cacm.acm.org/videos/multiparty-privacy-in-social-media

research highlights

DOI:10.1145/3230483

To view the accompanying paper, visit doi.acm.org/10.1145/3230485

Technical Perspective
Graphs, Betweenness Centrality, and the GPU

By John D. Owens

GRAPHS ARE THE natural data structures to represent relationships, and in our age of big data, graphs are very big indeed. For instance, Facebook's social graph has well over two billion users (vertices in the graph), and their friendships (edges in the graph) may number in the hundreds of billions. How do we make sense of data this large?

If possible, we can gain significant insight into complex problems of interest both to commerce and to science. Through graph data, we may be able to detect anomalies (say, intrusions into a computer network), make recommendations (say, which movie to watch), search a graph for patterns (say, credit card fraud), or detect communities (say, identifying proteins within a cell with similar functionality). Enabling faster graph computation allows us to find answers to these questions more quickly and cheaply.

As the graphics processor (GPU) has become ubiquitous in personal computers, supercomputers, and more recently datacenters, its advantages in raw performance and price-performance have motivated its use in graph computation. A significant body of recent research has demonstrated the performance advantages of GPUs over CPUs on a variety of graph computations. However, the GPU presents several challenges to authors of efficient graph implementations:

- To be effective on any problem, GPUs require large, parallel workloads. Thus GPU application authors must identify and expose significant parallelism in their applications. Fortunately, most graph computations allow parallelization over the graph's vertices, and large graphs exhibit more than enough parallelism to make GPUs a viable choice.
- However, graphs are particularly challenging because of the load imbalance across vertices. Some vertices have few neighbors, while others have many. A straightforward parallelization that assigns vertices to different processing units means units assigned vertices with few neighbors are idle while waiting for heavily loaded units to finish. The resulting *load balance problem* is perhaps the most significant challenge in writing an efficient graph computation.
- GPUs have modest-sized memories, and the largest graphs of interest cannot fit in a single GPU's memory. Distributing work across multiple GPUs faces two problems: efficiently partitioning both the data and computation across the GPUs in a load-balanced way, and structuring the multi-GPU computation so that the resulting communication between GPUs does not become a bottleneck.

The following work by McLaughlin and Bader ably addresses these challenges in the important context of a graph computation called *betweenness centrality* (BC). Centrality metrics on a graph ascertain the most important nodes in that graph. Betweenness centrality—perhaps the most popular centrality metric—does so by counting how many shortest paths in the graph flow through a particular node. For instance, we may wish to know the most important airports in the world. Betweenness centrality would consider every possible pair of airports and compute the fastest route between each pair; airports involved in the fastest routes would then be the most important.

While the straightforward method for computing betweenness centrality (individually compute the shortest paths between all pairs) would be quite expensive for large graphs, the much cheaper formulation of Ulrik Brandes (2001) is the basis for any modern computation of betweenness centrality, including the following paper. Its efficient parallelization on GPUs is a significant challenge and the focus of this work.

In my view, this paper offers three key insights:

1. The authors describe two different GPU parallelizations of betweenness centrality. Their "work-efficient" approach assigns only active vertices to processing units; their "edge-parallel" approach instead assigns edges to processing units.
2. They analyze both methods through the lens of different types of graphs. Large-diameter graphs with a uniform out-degree are well suited for the work-efficient approach, while the edge-parallel approach is a better fit for scale-free (small-world) graphs. The authors show how to choose the right approach at runtime by first sampling the graph to estimate graph diameter and then choosing the better approach to compute BC on the entire graph.
3. They also identify coarser parallelism in the overall computation that allows them to distribute work across multiple GPUs and demonstrate near-linear speedup on a 192-GPU cluster.

These contributions are crucial building blocks for future work on GPU graph computation. For BC, important next steps include incremental computations on mutable graphs and multi-GPU scaling to graphs that do not fit into GPU memory. More broadly, while work in GPU graph analytics today generally focuses on relatively simple graph problems, real-world workloads are more complex. Our community must move toward frameworks that address these more complex problems that deliver both high performance and high-level programmability.

John D. Owens is the Child Family Professor of Engineering and Entrepreneurship in the Department of Electrical and Computer Engineering at the University of California, Davis, CA, USA.

DOI:10.1145/3230485

Accelerating GPU Betweenness Centrality

By Adam McLaughlin[a] and David A. Bader

Abstract

Graphs that model social networks, numerical simulations, and the structure of the Internet are enormous and cannot be manually inspected. A popular metric used to analyze these networks is Betweenness Centrality (BC), which has applications in community detection, power grid contingency analysis, and the study of the human brain. However, these analyses come with a high computational cost that prevents the examination of large graphs of interest.

Recently, the use of Graphics Processing Units (GPUs) has been promising for efficient processing of unstructured data sets. Prior GPU implementations of BC suffer from large local data structures and inefficient graph traversals that limit scalability and performance. Here we present a hybrid GPU implementation that provides good performance on graphs of arbitrary structure rather than just scale-free graphs as was done previously. Our methods achieve up to 13× speedup on high-diameter graphs and an average of 2.71× speedup overall compared to the best existing GPU algorithm. We also observe near linear speedup when running BC on 192 GPUs.

1. INTRODUCTION

Network analysis is a fundamental tool for domains as diverse as compilers,[17] social networks,[14] and computational biology.[5] Real world applications of these analyses involve tremendously large networks that cannot be inspected manually. An example of a graph analytic that has found significant attention in recent literature is BC. Betweenness centrality has been used for finding the best location of stores within cities,[20] studying the spread of AIDS in sexual networks,[13] power grid contingency analysis,[11] and community detection.[23] The variety of fields and applications in which this method of analysis has been employed shows that graph analytics require algorithmic techniques that make them performance portable to as many network structures as possible. Unfortunately, the fastest known algorithm for calculating BC scores has $O(mn)$ complexity for unweighted graphs with n vertices and m edges, making the analysis of large graphs challenging. Hence there is a need for robust, high-performance graph analytics that can be applied to a variety of network structures and sizes.

Graphics Processing Units (GPUs) provide excellent performance for regular, dense, and computationally demanding subroutines such as matrix multiplication. However, there has been recent success in accelerating irregular, memory-bound graph algorithms on GPUs as well.[6, 17, 19] Prior implementations of betweenness centrality on the GPU have outperformed their CPU counterparts, particularly on scale-free networks; however, they are limited in scalability to larger graph instances, use asymptotically inefficient algorithms that mitigate performance on high diameter graphs, and aren't general enough to be applied to the variety of domains that can leverage their results.

This article alleviates these problems by making the following contributions:

- We provide a work-efficient algorithm for betweenness centrality on the GPU that works especially well for networks with a large diameter.
- For generality, we propose an algorithm that chooses between leveraging either the memory bandwidth of the GPU or the asymptotic efficiency of the work being done based on the structure of the graph being processed. We present an online approach that uses a small amount of initial work from the algorithm to suggest which method of parallelism would be best for processing the remaining work.
- We implement our approach on a single GPU system, showing an average speedup of 2.71× across a variety of both real-world and synthetic graphs over the best previous GPU implementation. Additionally, our implementation attains near linear speedup on a cluster of 192 GPUs.

2. BACKGROUND

2.1. Definitions

Let a graph $G = (V, E)$ consist of a set V of $n = |V|$ vertices and a set E of $m = |E|$ edges. A path from a vertex u to a vertex v is any sequence of edges originating from u and terminating at v. Such a path is a *shortest path* if its sequence contains a minimal number of edges. A Breadth-First Search (BFS) explores vertices of a graph by starting a "source" (or "root") vertex and exploring its neighbors. The neighbors of these vertices are then explored and this process repeats until there are no remaining vertices to be explored. Each set of inspected neighbors is referred to as a *vertex frontier* and the set of outgoing edges from a vertex frontier is referred to as an *edge-frontier*. The *diameter* of a graph is the length of the longest shortest path between any pair of vertices. A *scale-free* graph has a degree distribution that follows a power law, where a small number of vertices have a large number of outgoing

The original version of this paper is entitled "Scalable and High Performance Betweenness Centrality on the GPU" and was published in the *Proceedings of the 26th ACM/IEEE International Conference of High Performance Computing, Networking, Storage, and Analysis* (SC '14), 572–583.

[a]D.E. Shaw Research, New York, NY, USA.

edges and a large number of vertices have a small number of outgoing edges.[2] Finally, a *small world* graph has a diameter that is proportional to the logarithm of the number of vertices in the graph.[25] In these networks every vertex can be reached from every other vertex by traversing a small number of edges.

Representation of sparse graphs in memory. The most intuitive way to store a graph in memory is as an *adjacency matrix*. For unweighted graphs, element A_{ij} of the matrix is equal to 1 if an edge exists from i to j and is equal to 0 otherwise. The real-world graphs that we examine in this article, however, are *sparse*, meaning that a vast majority of the elements are zeros in the adjacency matrix representation of these data sets. Rather than using $O(n^2)$ space to store the entire adjacency matrix, we use the *Compressed Sparse Row* (CSR) format, as shown in Figure 1. This representation consists of two arrays: *row offsets* (`R`) and *column indices* (`C`). The column indices array is a concatenation of each vertex's adjacency list into an array of m elements. The row offsets array in an $n + 1$ element array that points at where each vertex's adjacency list beings and ends within the column indices array. For example, the adjacency list of a vertex u starts at `C[R[u]]` and ends at `C[R[u+1]-1]` (inclusively).

2.2. Brandes's algorithm

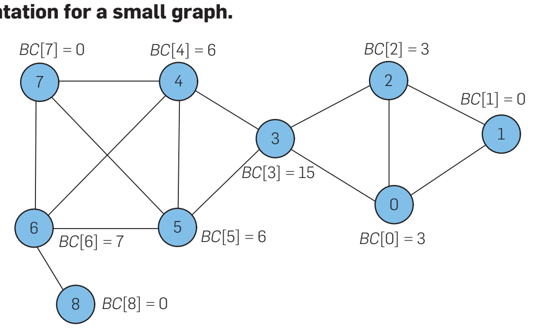
Betweenness centrality was originally developed in the social sciences for classifying people who were central to networks and could thus influence others by withholding information or altering it.[8] The metric attempts to distinguish the most influential vertices in a network by measuring the ratio of shortest paths passing through a particular vertex to the total number of shortest paths between all pairs of vertices. Intuitively, this ratio determines how well a vertex connects pairs of other vertices in the network. Formally, the Betweenness centrality of a vertex v is defined as:

$$BC(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (1)$$

where σ_{st} is the number of shortest paths between vertices s and t and $\sigma_{st}(v)$ is the number of those shortest paths that pass through v.

Consider Figure 1. Vertex 3 is the only vertex that lies on paths from its left (vertices 4 through 8) to its right (vertices 0 through 2). Hence vertex 3 lies on all of the shortest paths

Figure 1. Example betweenness centrality scores and CSR representation for a small graph.



R = [0, 3, 5, 8, 12, 16, 20, 24, 27, 28]

C = [1, 2, 3 | 0, 2 | 0, 1, 3 | 0, 2, 4, 5 | 3, 5, 6, 7 | 3, 4, 6, 7 | 4, 5, 7, 8 | 4, 5, 6 | 6]

between these pairs of vertices and has a high BC score. In contrast, vertex 8 does not belong on a path between any pair of the remaining vertices in the graph and thus has a BC score of zero. Note that the scores reflected in Figure 1 treat a path from vertex u to vertex v as equivalent to a path from vertex v to vertex u since these paths are undirected. In other words, to avoid double counting the number of (undirected) shortest paths we divide the scores by two.

The magnitude of BC values also scales with the size of the network. For a fair comparison of BC values between vertices of two different graphs, a commonly used technique is to normalize the BC scores by their largest possible value[4]: $(n - 1)(n - 2)$. Such a comparison could be useful for comparing discrete slices of a network that changes over time.[15]

Naïve implementations of Betweenness Centrality solve the all-pairs shortest-paths problem using the $O(n^3)$ Floyd-Warshall algorithm and augment this result with path counting. Brandes improved upon this approach with an algorithm that runs in $O(mn)$ time for unweighted graphs.[3] The key concept of Brandes's approach is the *dependency* of a vertex v with respect to a given source vertex s:

$$\delta_s(v) = \sum_{w \in succ(v)} \frac{\sigma_{sv}}{\sigma_{sw}}(1 + \delta_s(w)) \quad (2)$$

The recursive relationship between the dependency of a vertex and the dependency of its successors allows a more asymptotically efficient calculation of the centrality metric. Brandes's algorithm splits the betweenness centrality calculation into two major steps:

1. Find the number of shortest paths between each pair of vertices.
2. Sum the dependencies for each vertex.

We can redefine the calculation of BC scores in terms of dependencies as follows:

$$BC(v) = \sum_{s \neq v} \delta_s(v) \quad (3)$$

2.3. GPU architecture and programming model

The relatively high memory bandwidth of GPUs compared to that of conventional CPUs has resulted in many high-performance GPU graph algorithms.[15, 17, 19] Compared to CPUs, GPUs tend to rely on latency hiding rather than caching and leverage a *Single-Instruction, Multiple-Thread* (SIMT) programming model. The SIMT model allows for transistors to be allocated to additional processor cores rather than structures for control flow management.

GPUs are comprised of a series of *Streaming Multiprocessors* (SMs), each of which manages hundreds of threads. The threads within each SM execute in groups of 32 threads (on current NVIDIA architectures) called *warps*. Although the execution paths of the threads within each warp may diverge, peak performance is attained when all threads within a warp execute the same instructions. Synchronization between the warps of a particular SM is inexpensive but properly synchronizing all of the SMs of the GPU requires the launch of a separate *kernel*, or function that executes on the device. GPU threads have access to many registers (typically 255 or so), a small amount (typically 48KB) of programmer managed *shared memory* unique to each SM, and a larger *global memory* that can be accessed by all SMs.

3. PRIOR GPU IMPLEMENTATIONS

Two well-known GPU implementations of Brandes's algorithm have been published within the last few years. Jia et al.[10] compare two types of fine-grained parallelism, showing that one is preferable over the other because it exhibits better memory bandwidth on the GPU. Shi and Zhang present *GPU-FAN*[22] and report a slight speedup over Jia et al. by using a different distribution of threads to units of work. Both methods focus their optimizations on scale-free networks.

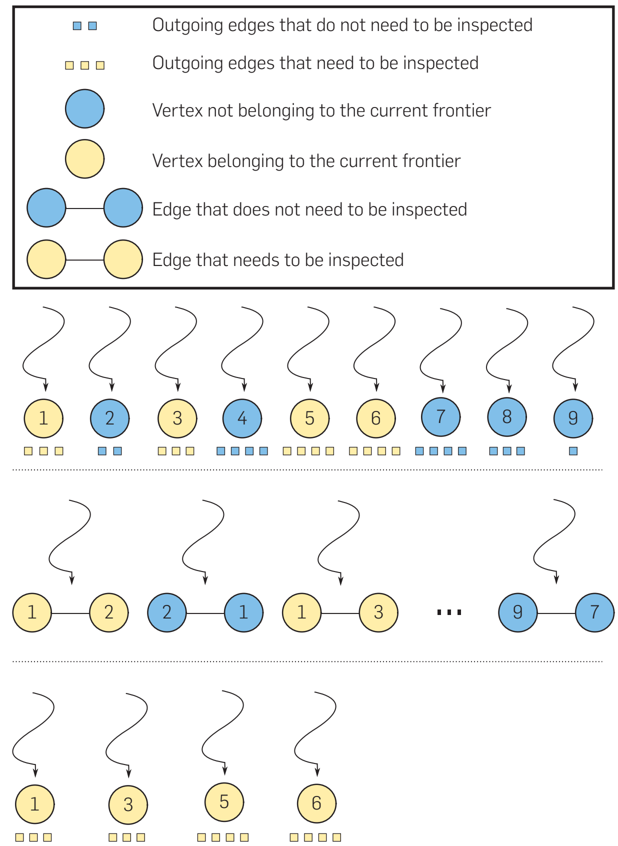
3.1. Vertex and edge parallelism

Jia et al. discussed two distributions of threads to graph entities: *vertex-parallel* and *edge-parallel*.[10] The vertex-parallel approach assigns a thread to each vertex of the graph and that thread traverses all of the outgoing edges from that vertex. In contrast, the edge-parallel approach assigns a thread to each edge of the graph and that thread traverses that edge only. In practice, the number of vertices and edges in a graph tend to be greater than the available number of threads so each thread sequentially processes multiple vertices or edges.

For both the shortest path calculation and the dependency accumulation stages the number of edges traversed per thread by the vertex-parallel approach depends on the out-degree of the vertex assigned to each thread. The difference in out-degrees between vertices causes a load imbalance between threads. For scale-free networks this load imbalance can be a tremendous issue, since the distribution of outdegrees follows a power law where a small number of vertices will have a substantial number of edges to traverse.[2] The edge-parallel approach solves this problem by assigning edges to threads directly. Both the vertex-parallel and edge-parallel approaches from Jia et al. use an inefficient $O(n^2 + m)$ graph traversal that checks if each vertex being processed belongs to the current depth of the search.

Figure 2 illustrates the distribution of threads to work for the vertex-parallel and edge-parallel methods. Using the same graph as shown in Figure 1, consider a Breadth-First Search starting at vertex 4. During the second iteration of the search, vertices 1, 3, 5, and 6 are in the vertex frontier, and hence their edges need to be inspected. The vertex-parallel method, shown in the top portion of Figure 2, distributes one thread to each vertex of the graph even though the edges connecting most of the vertices in the graph do not need to be traversed, resulting in wasted work. Also note that each thread is responsible for traversing a different number of edges (denoted by the small squares beneath each vertex), leading to workload imbalances. The edge-parallel method, shown in the middle portion of Figure 2, does not have the issue of load imbalance because each thread has one edge to traverse. However, this assignment of threads also results in wasted work because the edges that do not originate from vertices in the frontier do not need to be inspected in this particular iteration (but will be unnecessarily inspected during *every* iteration). Finally, the bottom portion of Figure 2 shows a work-efficient traversal iteration where each vertex in the frontier is assigned a thread. In this case only useful work is conducted although a load imbalance may exist among threads.

Figure 2. Illustration of the distribution of threads to units of work. Top: Vertex-parallel. Middle: Edge-parallel. Bottom: Work-efficient.



3.2. GPU-FAN

The *GPU-FAN* package from Shi and Zhang was designed for the analysis of biological networks representing protein communications or genetic interactions.[22] Similar to the implementation from Jia et al., GPU-FAN uses the edge-parallel method for load balancing across threads. The GPU-FAN package, however, focuses only on fine-grained parallelism, using all threads from all thread blocks to traverse edges in parallel for one source vertex of the BC computation at a time. In contrast, the implementation from Jia et al. uses the threads within a block traverse edges in parallel while separate thread blocks each focus on the independent roots of the BC computation.

4. METHODOLOGY

4.1. Work-efficient approach

Taking note of the issues mentioned in the previous section, we now present the basis for our work-efficient implementation of betweenness centrality on the GPU. Our approach leverages optimizations from the literature in addition to our own novel techniques. The most important distinction between our approach and prior work is that we use explicit queues for graph traversal. Since levels of the graph are processed in parallel we use two queues to distinguish vertices that are in the current level of the search (Q_{curr}) from vertices that are to be processed during the next level of the search (Q_{next}). For the dependency accumulation stage we initialize S and its length. In this case, we need to keep track of vertices at all levels of the search and hence we only use one data structure to store these vertices. To distinguish the sections of S that correspond to each level of the search we use the *ends* array, where $ends_{len} = 1 + max_{v \in V} \{d[v]\}$ at the end of the traversal. Vertices corresponding to depth i of the traversal are located from index $ends[i]$ to index $ends[i + 1] - 1$ (inclusively) of S. This usage of the *ends* and S arrays is analogous to the arrays used to store the graph in CSR format.

Algorithm 1: Work-efficient betweenness centrality shortest path calculation.

```
1   Stage 1: Shortest Path Calculation
2   while true do
3       for v ∈ Q_curr do in parallel
4           for w ∈ neighbors(v) do
5               if atomicCAS(d[w], ∞, d[v] + 1) = ∞ then
6                   t ← atomicAdd(Q_next_len, 1)
7                   Q_next[t] ← w
8               if d[w] = d[v] + 1 then
9                   atomicAdd(σ[w], σ[v])
10      barrier()
11      if Q_next_len = 0 then
12          depth ← d[S[S_len − 1]] − 1
13          break
14      else
15          for tid ← 0 ... Q_next_len − 1 do in parallel
16              Q_curr[tid] ← Q_next[tid]
17              S[tid + S_len] ← Q_next[tid]
18          barrier()
19          ends[ends_len] ← ends[ends_len − 1] + Q_next_len
20          ends_len ← ends_len + 1
21          Q_curr_len ← Q_next_len
22          S_len ← S_len + Q_next_len
23          Q_next_len ← 0
24          barrier()
```

A work-efficient shortest path calculation stage is shown in Algorithm 1. The queue Q_{curr} is initialized to contain only the source vertex. Iterations of the while loop correspond to the traversal of depths of the graph. The parallel for loop in Line 3 assigns one thread to each element in the queue such that edges from other portions of the graph aren't unnecessarily traversed. The atomic Compare And Swap (CAS) operation on Line 5 is used to prevent multiple insertions of the same vertex into Q_{next}. This restriction allows us to safely allocate $O(n)$ memory for Q_{next} instead of $O(m)$ in the case that duplicate queue entries are allowed. Since we only require one thread for each element in Q_{curr} rather than one thread for every vertex or edge in the graph, this atomic operation experiences limited contention and thus doesn't significantly reduce performance.

The conditional on Line 11 checks to see if the queue containing vertices for the next depth of the search is empty; if so, the search is complete, so we break from the outermost while loop. Otherwise, we transfer vertices from Q_{next} to Q_{curr}, add these vertices to the end of S for the dependency accumulation, and do the appropriate bookkeeping to set the lengths of these arrays.

Algorithm 2 shows a work-efficient dependency accumulation. We are able to eliminate the use of atomics by checking *successors* rather than the predecessors of each vertex. Rather than having multiple vertices that are currently being processed in parallel update the dependency of their common ancestor atomically, the ancestor can update itself based on its successors without the need for atomic operations.[14]

Algorithm 2: Work-efficient betweenness centrality dependency accumulation.

```
1   Stage 2: Dependency Accumulation
2   while depth > 0 do
3       for tid ← ends[depth] ... ends[depth + 1] − 1 do in
        parallel
4           w ← S[tid]
5           dsw ← 0
6           sw ← σ[w]
7           for v ∈ neighbors(w) do
8               if d[v] = d[w] + 1 then
9                   dsw ← dsw + (sw / σ[v]) (1 + δ[v])
10          δ[w] ← dsw
11      barrier()
12      depth ← depth − 1
```

Note that the parallel for loop in Line 3 of Algorithm 2 assigns threads only to vertices that need to accumulate their dependency values; this is where the bookkeeping done to keep track of separate levels of the graph traversal in the *ends* array comes to fruition. Rather than naïvely assigning a thread to each vertex or edge and checking to see if that vertex or edge belongs to the current depth we instead can instantly extract vertices of that depth since they are a consecutive block of entries within S. This strategy again prevents unnecessary branch overhead and accesses to global memory that are made by previous implementations. For further implementation details we refer the reader to the associated conference paper.[16]

4.2. Rationale for hybrid methods

The major drawback of the approach outlined in the previous section is the potential for significant load imbalance between threads. Although our approach efficiently assigns threads to units of useful work, the distribution of edges to threads is entirely dependent on the structure of the graph. Our approach is significantly faster than other methods on graphs with a large diameter because such graphs tend to have a more uniform distribution of outdegree. On scale-free or small world graphs, however, the algorithm outlined in the previous section does not improve performance. Based on this result we propose a hybrid approach that chooses between the edge-parallel and work-efficient methods based on the structure of the graph. Rather than preprocessing the graph to attempt to determine if it can be classified as a scale-free or small world graph, we implement our hybridization as an online approach.

Figure 3 illustrates our rationale behind the decision to use a hybrid algorithm. Each sub-figure shows how the vertex frontier evolves for three randomly chosen source vertices within a graph. Note that the axes of the sub-figures are on different scales to appropriately show trends in the frontiers. Although the position of the source vertex plays an important role in precisely how the vertex frontier changes with search iteration, we can see that the general sizes and changes in size of the vertex frontier across iterations of the search are more dependent on the overall structure of the graph. For high-diameter graphs such as *rgg_n_2_20* and *delaunay_n20* (Figures 3a and 3b), the vertex frontier grows gradually and is always a small portion of the total number of vertices in the graph. For graphs with a smaller diameter such as *kron_g500-logn20* (Figure 3c), the vertex frontier grows large after just a few iterations and contains over half of the total number of vertices in the graph at its peak.

Intuitively, for large vertex frontiers, the edge-parallel approach is favorable because of its memory throughput whereas for small vertex frontiers the work-efficient approach is favorable because the number of edges that will be traversed is significantly smaller than the total number of edges in the graph.

4.3. Sampling

The exact computation of betweenness centrality computes a BFS for each vertex in the graph. Since all of these searches are independent, they can be executed in parallel. For graphs whose vertices mostly belong to one large connected component, the amount of time to process each source vertex is roughly equivalent, as the same number of edges need to be traversed for each source vertex. Therefore the amount of time required to process k source vertices is roughly k times the time required to process one source vertex.[21]

Algorithm 3: Sampling method for selecting parallelization strategy.

Input: Set of n_{samps} connected component sizes (*keys*)
sort(*keys*)
barrier()
if $keys[n_{samps}/2] < \gamma * \log_2(n)$ **then**
  //Switch to the edge-parallel method

Using the above analysis, an estimate of the average size of the connected components within the graph (and thus the preferred method of parallelism) is obtained by processing a small subset of its vertices. Algorithm 3 shows how this method is implemented. We initially use the work-efficient method to process a small subset of source vertices, recording the maximum depth of each of their BFS traversals. We then use the median of this set to be our estimate of the graph diameter. If this median is smaller than a threshold (determined by the parameter γ) then it is likely that our graph is a small-world or scale-free graph and that we should switch to using the edge-parallel approach.

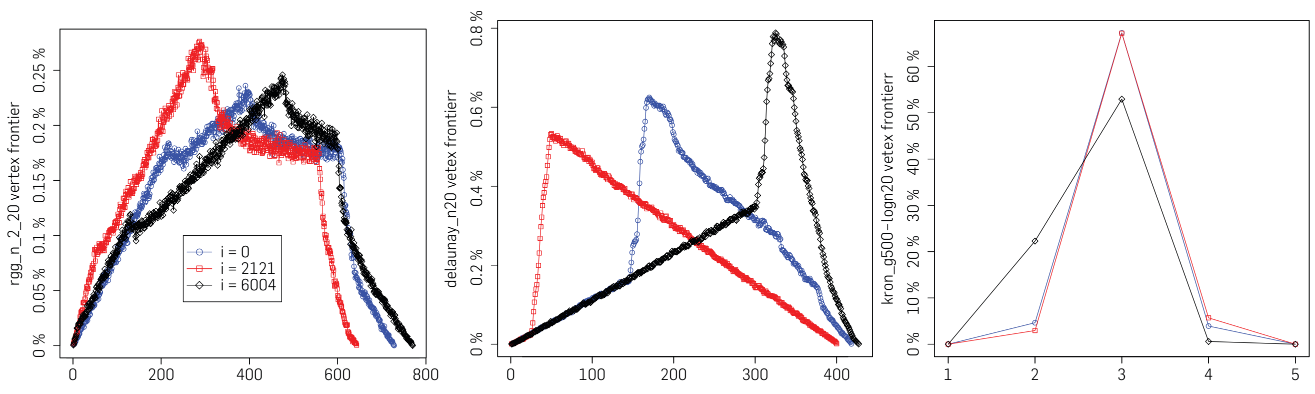
5. RESULTS

5.1. Experimental setup

Single-node GPU experiments were implemented using the Compute Unified Device Architecture (CUDA) 6.0 Toolkit. The CPU is an Intel Core i7-2600K processor running at 3.4 GHz with an 8MB cache and 16GB of DRAM. The GPU is a

Our implementation is available at https://github.com/Adam27X/hybrid_BC.

Figure 3. Evolution of vertex frontiers (as a percentage of total vertices) for different classifications of graphs.



(a) rgg_n_2_20 (b) delaunay_n20 (c) kron_g500-logn20

GeForce GTX Titan that has 14 SMs and a base clock of 837 MHz. The Titan has 6GB of GDDR5 memory and is a CUDA compute capability 3.5 ("Kepler") GPU.

Multi-node experiments were run on the Keeneland Initial Delivery System (KIDS).[24] KIDS has two Intel Xeon X5660 CPUs running at 2.8 GHz and three Tesla M2090 GPUs per node. Nodes are connected by an Infiniband Quadruple Data Rate (QDR) network. The Tesla M2090 has 16 SMs, a clock frequency of 1.3 GHz, 6GB of GDDR5 memory, and is a CUDA compute capability 2.0 ("Fermi") GPU.

We compare our techniques to both GPU-FAN[22] and Jia et al.[10] when possible, using their implementations that have been provided online. The graphs used for these comparisons are shown in Table 1. These graphs were taken from the 10th DIMACS Challenge,[1] the University of Florida Sparse Matrix Collection,[7] and the Stanford Network Analysis Platform (SNAP).[12] These benchmarks contain both real-world and randomly generated instances of graphs that correspond to a wide variety of practical applications and network structures. We focus our attention on the exact computation of BC, noting that our techniques can be trivially adjusted for approximation.

5.2. Scaling

First we compare how well our algorithm scales with graph size for three different types of graphs. Since the implementation of Jia et al. cannot read graphs that contain isolated vertices, we were unable to obtain results using this reference implementation for the random geometric (*rgg*) and simple Kronecker (*kron*) graphs. Additionally, since the higher scales caused GPU-FAN to run out of memory, we simply extrapolated what we would expect these results to look like from the results at lower scales (denoted by dotted lines). Note that from one scale to the next the number of vertices and number of edges both double.

Noting the log-log scale on the axes, we can see from Figure 4a that the sampling approach outperforms the algorithm from GPU-FAN by over 12× for all scales of *rgg*. It is interesting to note that the sampling approach only takes slightly more time than GPU-FAN when the sampling approach processes a graph four times as large. For the *delaunay* mesh graphs as shown in Figure 4b we can see that the edge-parallel method and the sampling approach both outperform GPU-FAN for all scales. The edge-parallel approach even outperforms the sampling approach for graphs containing less than 10,000 vertices; however, it should be noted that these differences in timings are trivial as they are on the order of milliseconds. As the graph size increases the sampling method clearly becomes dominant and the speedup it achieves grows with the scale of the graph. Finally, we compare the sampling approach to GPU-FAN for *kron* in Figure 4c. Although GPU-FAN is marginally faster than the sampling approach for the smallest scale graph we can see that the sampling approach is best at the next scale and the trend shows the amount by which the sampling approach is best grows with scale. Furthermore, neither of the previous implementations could support this type of graph at larger scales whereas the sampling method can support even larger scales.

5.3. Benchmarks

Figure 5 provides a comparison of the various parallelization methods discussed in this article to the edge-parallel method from Jia et al.[10] For road networks and meshes (*af_shell, del20, luxem*) all of the methods outperform the edge-parallel method by about 10×. The amount of unnecessary work performed by the edge-parallel method for these graphs is severe. For the remaining graphs (scale-free and small-world graphs) using the work-efficient method alone performs slower than the edge-parallel method whereas the sampling method is either the same or slightly better. In these cases we see the advantage of choosing our method of parallelization online.

In the most extreme case, the edge-parallel approach requires more than two and half days to process the *af_shell9* graph while the sampling approach cuts this time down to under five hours. Similarly, the edge-parallel approach takes over 48 min to process the *luxembourg.osm* road network whereas the sampling approach requires just 6 min. Overall, sampling performs 2.71× faster on average than the edge-parallel approach.

Table 1. Graph datasets used for this study.

Graph	Vertices	Edges	Max degree	Diameter	Description
af_shell9	504,855	8,542,010	39	497	Sheet metal forming
caidaRouterLevel	192,244	609,066	1,071	25	Internet router-level topology
cnr-2000	325,527	2,738,969	18,236	33	Web crawl
com-amazon	334,863	925,872	549	46	Amazon product co-purchasing
delaunay_n20	1,048,576	3,145,686	23	444	Random triangulation
kron_g500-logn20	1,048,576	44,619,402	131,503	6	Kronecker
loc-gowalla	196,591	1,900,654	29,460	15	Geosocial
luxembourg.osm	114,599	119,666	6	1,336	Road map
rgg_n_2_20	1,048,576	6,891,620	36	864	Random geometric
smallworld	100,000	499,998	17	9	Small world phenomenon

Figure 4. Scaling by problem size for three different types of graphs.

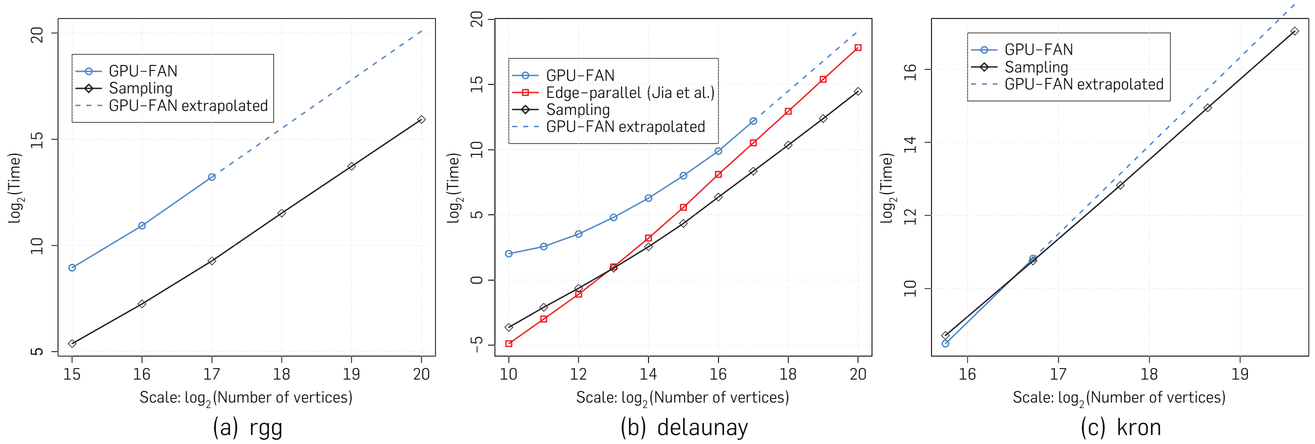
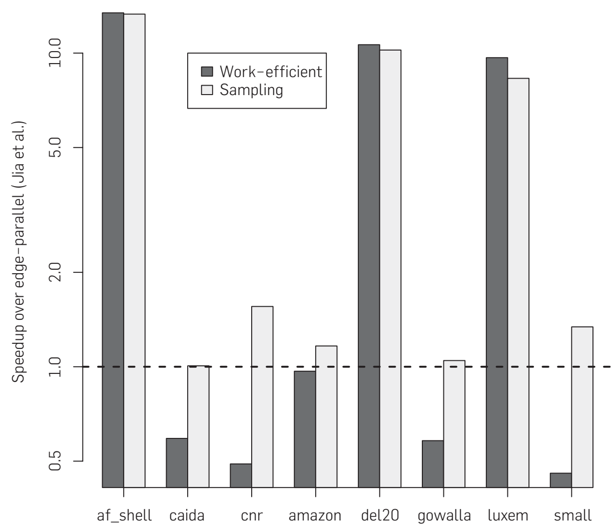


Figure 5. Comparison of work-efficient and sampling methods.



5.4. Multi-GPU experiments

Although our approaches leverage both coarse and fine-grained parallelism there is still more available parallelism than can be handled by a single GPU. Our methods easily extend to multiple GPUs as well as multiple nodes. We extend the algorithm by distributing a subset of roots to each GPU. Since each root can be processed independently in parallel, we should expect close to perfect scaling if each GPU has a sufficient (and an evenly distributed) amount of work.

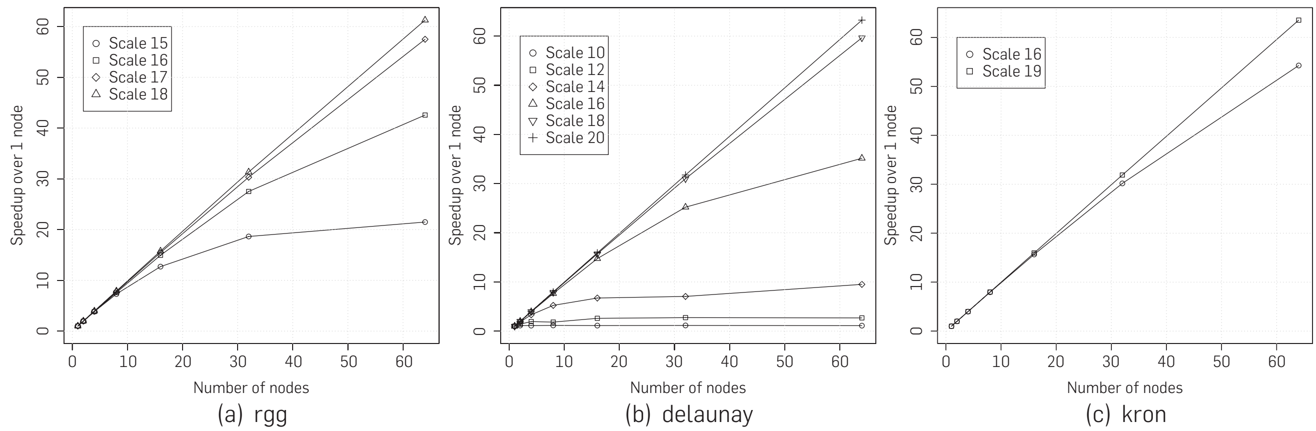
Since the local data structures for each root are independent (and thus only need to reside on one GPU), we replicate the data representing the graph itself across all GPUs to eliminate communication bottlenecks. Once each GPU has its local copy of the BC scores these local copies are accumulated for all of the GPUs on each node. Finally, the node-level scores are reduced into the global BC scores by a simple call to *MPI_Reduce*(). Figure 6 shows how well our algorithm scales out to multiple GPUs for *delaunay*, *rgg*, and *kron* graphs. It shows that linear speedup is easily achievable if the problem size is sufficiently large (i.e., if there is sufficient work for each GPU). Linear speedups are achievable at even smaller scales of graphs for denser network structures. For instance, using 64 nodes provides about a 35× speedup over a single node for scale 16 *delaunay* graph whereas using the same number of nodes at the same scale for *rgg* and *kron* graphs provides over 40× and 50× speedups respectively. The scaling behavior seen in Figure 6 is not unique to these graphs because of the vast amount of coarse-grained parallelism offered by the algorithm. For graphs of large enough size this scalability can be obtained independently of network structure.

6. CONCLUSION

In this article we have discussed various methods for computing Betweenness Centrality on the GPU. Leveraging information about the structure of the graph, we present several methods that choose between two methods of parallelism: edge-parallel and work-efficient. For high-diameter graphs using asymptotically optimal algorithms is paramount to obtaining good performance whereas for low-diameter graphs it is preferable to maximize memory throughput, even if unnecessary work is completed. In addition our methods are more scalable and general than existing implementations. Finally, we run our algorithm on a cluster of 192 GPUs, showing that speedup scales almost linearly with the number of GPUs, regardless of network structure. Overall, our single-GPU approaches perform 2.71× faster on average than the best previous GPU approach.

For future work we would like to efficiently map additional graph analytics to parallel architectures. The importance of robust, high-performance primitives cannot be overstated for the implementation of more complicated parallel algorithms. Ideally, GPU kernels should be modular and reusable; fortunately, packages such as Thrust[9] and CUB (CUDA Unbound)[18] are beginning to bridge this gap. A software environment in which users have access to a suite of high-performance graph analytics on the GPU

Figure 6. Multi-GPU scaling by number of nodes for various graph structures. Each node contains three GPUs.



would allow for fast network analysis and serve as a building block for more complicated programs.

Acknowledgments

The work depicted in this article was partially sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement #HR0011-13-2-0001. The content, views and conclusions presented in this document do not necessarily reflect the position or the policy of DARPA or the U.S. government, no official endorsement should be inferred. Distribution Statement A: "Approved for public release; distribution is unlimited." This work was also partially sponsored by NSF Grant ACI-1339745 (XScala). This research used resources of the Keeneland Computing Facility at the Georgia Institute of Technology, which is supported by the National Science Foundation under Contract OCI-0910735. Special thanks to Jeff Young for his assistance with running experiments on KIDS. Finally, we would also like to thank NVIDIA Corporation for their donation of a GeForce GTX Titan GPU.

References

1. Bader, D.A., Meyerhenke, H., Sanders, P., Wagner, D. (eds) *Graph Partitioning and Graph Clustering—10th DIMACS Implementation Challenge*, volume 588 of *Contemporary Mathematics* (2013).
2. Barabási, A.-L., Albert, R. Emergence of scaling in random networks. *Science 286*, 5439 (1999), 509–512.
3. Brandes, U. A faster algorithm for betweenness centrality. *J. Math. Sociol. 25* (2001), 163–177.
4. Brandes, U. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks 30*, 2 (2008), 136–145.
5. Bullmore, E., Sporns, O. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nat. Rev. Neurosci. 10*, 3 (2009), 186–198.
6. Davidson, A.A., Baxter, S., Garland, M., Owens, J.D. Work-efficient parallel GPU methods for single-source shortest paths. In *International Parallel and Distributed Processing Symposium 28* (2014).
7. Davis, T.A., Hu, Y. The university of Florida sparse matrix collection. *ACM Trans. Math. Softw. 38*, 1 (Dec. 2011), 1:1–1:25.
8. Freeman, L.C. A set of measures of centrality based on betweenness. *Sociometry* (1977), 35–41.
9. Hoberock, J., Bell, N. Thrust: A Parallel Template Library. Online at: http://thrust.googlecode. com, 42:43 (2010).
10. Jia, Y., Lu, V., Hoberock, J., Garland, M., Hart, J.C. Edge v. node parallelism for graph centrality metrics. *GPU Computing Gems 2* (2011), 15–30.
11. Jin, S., Huang, Z., Chen, Y., Chavarria-Miranda, D., Feo, J., Wong, P.C. A novel application of parallel betweenness centrality to power grid contingency analysis. In *IEEE International Symposium on Parallel Distributed Processing (IPDPS)* (2010), 1–7.
12. Leskovec, J., Krevl, A. SNAP datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.
13. Liljeros, F., Edling, C.R., Amaral, L.A., Stanley, H.E., Aberg, Y. The web of human sexual contacts. *Nature 411*, 6840 (2001), 907–908.
14. Madduri, K., Ediger, D., Jiang, K., Bader, D., Chavarria-Miranda, D. A faster parallel algorithm and efficient multithreaded implementations for evaluating betweenness centrality on massive datasets. In *IEEE International Symposium on Parallel & Distributed Processing (IPDPS)* (May 2009), 1–8.
15. McLaughlin, A., Bader, D. Revisiting edge and node parallelism for dynamic GPU graph analytics. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)* (2014).
16. McLaughlin, A., Bader, D.A. Scalable and high performance betweenness centrality on the GPU. In *Proceedings of the 26th ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis (SC)* (2014).
17. Mendez-Lojo, M., Burtscher, M., Pingali, K. A GPU implementation of inclusion-based points-to analysis. In *Proceedings of the 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '12 (New York, NY, USA, 2012). ACM, 107–116.
18. Merrill, D. CUDA Unbound. 2013. https://nvlabs.github.io/cub/ (accessed April 11, 2014).
19. Merrill, D., Garland, M., Grimshaw, A. Scalable GPU graph traversal. In *Proceedings of the 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '12 (New York, NY, USA, 2012). ACM, 117–128.
20. Porta, S., Latora, V., Wang, F., Strano, E., Cardillo, A., Scellato, S., Iacoviello, V., Messora, R. Street centrality and densities of retail and services in Bologna, Italy. *Environment and Planning B: Planning and design 36*, 3 (2009), 450–465.
21. Sariyüce, A.E., Saule, E., Kaya, K., Çatalyürek, Ü. Regularizing graph centrality computations. *J, Parallel Distrib. Comput. (JPDC)* (2014).
22. Shi, Z., Zhang, B. Fast network centrality analysis using GPUs. *BMC Bioinformatics 12*, 1 (2011), 149.
23. Soman, J., Narang, A. Fast community detection algorithm with GPUs and multicore architectures. In *International Parallel & Distributed Processing Symposium (IPDPS)* (IEEE, 2011), 568–579.
24. Vetter, J.S., Glassbrook, R., Dongarra, J., Schwan, K., Loftis, B., McNally, S., Meredith, J., Rogers, J., Roth, P., Spafford, K., Yalamanchili, S. Keeneland: Bringing heterogeneous GPU computing to the computational science community. *Comput. Sci. & Engineering 13*, 5 (2011), 90–95.
25. Watts, D.J., Strogatz, S.H. Collective dynamics of 'Small-World' networks. *Nature 393*, 6684 (1998), 440–442.

Adam McLaughlin (adam27x@gatech.edu), School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

David A. Bader (bader@cc.gatech.edu), School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

CAREERS

The Carnegie Mellon University College of Engineering
Professor and Head, Department of Electrical and Computer Engineering

The Carnegie Mellon University College of Engineering invites applications for the position of Professor and Head of the Department of Electrical and Computer Engineering (ECE).

The ECE community currently includes 77 core faculty, 65 affiliated faculty, 35 adjunct faculty, 120 staff and researchers, and 530 undergraduate and 800 graduate students. The mission, goals, and current activities of the department are described in more detail at http://www.ece.cmu.edu/about/index.html.

An accomplished academic leader is sought to provide strategic vision and outstanding operational management, to oversee high-quality research and educational programs, and to guide the ECE community to the next level of accomplishment and impact. The ECE Department Head will maintain and strengthen the culture of collaboration and innovation in the department; expand the reputation of the department; lead the department in strategic planning; promote diversity within the department and institution; and, demonstrate commitment to a high-quality student experience.

Individuals with expertise across a wide range of areas related to electrical and computer engineering will be considered. Applicants must have an earned Ph.D. in a field of engineering or science closely related to electrical and computer engineering. Teaching experience and an internationally-recognized record of research are essential. Applicants must be qualified for an appointment at the rank of Professor with tenure.

Applications will be accepted until August 15, 2018. Review of applications will begin on August 22, 2018 and will continue until the position is filled. The expected start date is January 2019, or as soon thereafter as possible. Confidential e-mail inquiries concerning this position may be sent to the Search Committee at sh2s@andrew.cmu.edu

Interested candidates should submit applications electronically: https://apply.interfolio.com/51197

Please include a cover letter, curriculum vitae, statement of research experience and interests, and statement of teaching experience and interests. The Search Committee will solicit names of references and their contact information only for a subset of applicants after the initial review. These applicants will be notified in advance before letters are requested.

Carnegie Mellon is an equal opportunity employer and is committed to increasing the diversity of its community on a range of intellectual and cultural dimensions. Carnegie Mellon welcomes faculty applicants who will contribute to this diversity through their research, teaching and service, including women, members of minority groups, protected veterans, individuals with disabilities, and others who would contribute in different ways.

Carnegie Mellon University seeks to meet the needs of dual-career couples and is a member of the Higher Education Recruitment Consortium (HERC) that assists with dual-career searches.

Tallinn University of Technology
3 Tenure Track Professorships and several post-doc positions in the areas of Trust in Software, Hardware Security and Trust, Internet of Intelligent Things

Estonia is the homeland of pioneering ICT solutions, which include Skype, e-identity, e-health, and e-governance. Building on these successes, the Estonian ICT industry and academia are establishing **three new research groups** in the areas of Trust in Software, Hardware Security and Trust, Internet of Intelligent Things.

The School of Information Technology at Tallinn University of Technology is looking for candidates for **tenure track positions** with:

- ► a proven research track record
- ► leadership capabilities
- ► ability to acquire competitive funding
- ► ability and interest to collaborate across disciplines
- ► experience in developing an academic curriculum in English
- ► willingness to cooperate with industry and the public sector

Successful candidates are expected to perform basic and applied research and teaching.

For **tenure track professors** the university offers:

- ► A competitive **net salary** in a range of 45,000 - 60,000 EUR per year
- ► **Dedicated funding for establishing a research group** (3-5 postdoc/researcher positions, 5 PhD student positions)
- ► Location in Tallinn, which is one of the fastest-growing IT hubs in Europe and home to The European Agency for the operational management of large-scale IT systems (eu-LISA) as well as the NATO Cyber Defence Centre of Excellence
- ► A young and internationally active faculty with rapid and efficient university e-management, allowing you to concentrate on the essential

Within the same call the university is looking for candidates to **20 new post-doc/researcher positions** with:

- ► a proven research track record
- ► ability and interest to collaborate across disciplines
- ► willingness to cooperate with industry and the public sector

For **post-doc/researcher positions** the university offers a competitive **net salary** in a range of 20,000 - 30,000 EUR per year.

Tallinn University of Technology is the flagship academic institution of Estonian ICT research, engineering and technology education. Our international faculty and staff form a unique and multidisciplinary team of technological, natural, exact, economic and health sciences. The university has 12,000 students and 2,000 employees and is centrally located in the Estonian capital city of Tallinn.

Applications should include a cover letter, documents proving that the candidate holds the required academic degree (PhD or an equivalent qualification) and has acquired the required education, a curriculum vitae and a list of publications, a statement of future research and teaching interests and other documents considered to be important by the applicant.

Documents should be sent by e-mail to konkurss@ttu.edu. Further information regarding the professorship, please contact Professor Gert Jervan, Dean of the School of IT, gert.jervan@ttu.ee

More information: https://ttu.ee/itpositions.

The University of Tennessee–Knoxville's Innovative Computing Laboratory (ICL)
Research Scientist (with MS or PhD) or Postdoctoral Researcher

The University of Tennessee–Knoxville's Innovative Computing Laboratory (ICL), under the direction of Prof. Jack Dongarra, is seeking full-time scientists (MS or PhD) or postdoctoral researchers to participate in the design, development, and

FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

Announcement of an open position at the Faculty of Informatics, TU Wien, Austria

FULL PROFESSOR of UBIQUITOUS COMPUTING

The TU Wien (Vienna University of Technology) invites applications for a Full Professor position at the Faculty of Informatics.

The successful candidate will have an outstanding research and teaching record in the field of Ubiquitous Computing and focuses on next generation ubiquitous computing systems and their application in authentic real world settings. Particular research topics of interest include sensor-rich environments; interactive and smart spaces; new interaction paradigms; Internet of Things; mobile and context-aware computing; awareness and privacy; and tangible, situated and embodied interaction.

We offer excellent working conditions in an attractive research environment in a city with an exceptional quality of life.

For a more detailed announcement and information on how to apply, please go to: http://www.informatik.tuwien.ac.at/vacancies

Application Deadline: **October 22, 2018**

maintenance of numerical software libraries for solving linear algebra problems on large, distributed-memory machines with multicore processors, hardware accelerators, and performance monitoring capabilities for new and advanced hardware and software technologies. The prospective researcher will coauthor papers to document research findings, present the team's work at conferences and workshops, and help lead students and other team members in their research endeavors in ongoing and future projects. Given the nature of the work, there will be opportunities for publication, travel, and high-profile professional networking and collaboration across academia, labs, and industry.

Joining this team will offer qualified candidates exciting career opportunities as they participate in the US Department of Energy's (DOE's) Exascale Computing Project (ECP). ICL is involved in several DOE-ECP projects, including SLATE (http://icl.utk.edu/slate/), PEEKS (http://icl.utk.edu/peeks/), xSDK (http://www.icl.utk.edu/research/xsdk4ecp), Exa-PAPI (http://icl.utk.edu/exa-papi/), CEED (https://ceed.exascaleproject.org/), Distributed Tasking for Exascale (PaRSEC) (http://icl.utk.edu/dte/), MAGMA (http://icl.cs.utk.edu/magma/), FFT-ECP (http://icl.utk.edu/fft/), and others.

Starting date is July 1, 2018 or later. All qualified candidates, be it fresh (MS or PhD) graduates or seasoned HPC veterans, are encouraged to apply.

For more information, contact Jack Dongarra (dongarra@icl.utk.edu) or check out ICL's jobs page: http://www.icl.utk.edu/jobs.

Requirements
An MS or PhD in computer science, computational sciences, or math is preferred.

Background in at least one of the following areas is also preferred: numerical linear algebra, high-performance computing (HPC), performance monitoring, machine learning, or data analytics. Full-time employment for up to 4 years with the possibility of further extensions based on funding availability and performance.

[CONTINUED FROM P. 96] game that applied European political theory to a colony on a distant planet called Rubi-Ka. Subsequently, we hopped over to *EVE Online*, set in an even more distant galaxy, then lost contact with each other.

Touring had become an influential astrophysicist, though I had little idea as to his recent research, in gamer terms, scientific questing, or technological crafting. After swearing me to secrecy, he told me the loss of the ISS was inconsequential, because his laboratory had detected and begun, incredibly, to decipher digital streaming from another solar system, still hidden from the public. He was now secretly employed by the U.S. State Department, one of the many agencies that had lost its traditional function and was now seeking new justifications for its taxpayer-funded budget. After several chats in *Third Life*, he invited me to join his team, working remotely in this non-game virtual world to interpret what appeared to be alien source code.

I have made two terrible mistakes in my life: Following orders to destroy the ISS was a failure of imagination, but now I forgot to look beyond my own imagination, failing to recognize the great potential for harm subordinating our technology to alien purposes, concentrating instead on computational puzzles of interest only to myself. The format of the alien software seemed a bit BASIC-y, with sequentially numbered fragments of code, and I quickly learned the extraterrestrial equivalents of GOSUB, RND, and REM. At least we thought that is what they were, and we grew excited when our first short program seemed to work, allowing us to write "Hello Earth!"

The data was still streaming down, and we soon had a very large program that was automatically adding procedures by the minute. It naturally unfolded into the equivalent of both parts of an online game that reminded me of *Star Wars Galaxies*, the user-side code and data to create the virtual environment and the server-side database to connect users and ensure their machines represented correct information. For a time, Touring was unconvinced the extraterrestrials from *Rubi-Ka* or the *New Eden Galaxy* were sending us a mere videogame, but I reassured him it was their way of offering us a virtual experience reflecting their alien world. He worried this might be dangerous, but I reminded him of the fun we had on *Rubi-Ka*.

"It was the size of a real planet but structured more like a giant rainbow crisscrossed with golden arcs of lightning and layers of emerald ledges a stairway to extraterrestrial heaven."

Meanwhile, *Wireless*, a cutting-edge popular computing blogsite, had hacked into the State Department, giving our project unexpected publicity. Soon several-hundred-thousand fellow Earthlings were exploring dozens of shards of *BraveNew*, as we called our virtual cosmic environment. As a simulation, it was the size of a real planet but structured more like a giant rainbow crisscrossed with golden arcs of lightning and layers of emerald ledges, a stairway to extraterrestrial heaven. However, we saw no aliens in *BraveNew* or non-player characters or mobs of any kind. That should have been a clue about the potential disaster likely to come.

About an hour ago, Tobor interrupted a virtual trajectory coding session I was having inside *BraveNew* by pushing me rather hard on my shoulder. "Ow!" I cried, "Be careful."

"Tobor is always careful," he said. "But for the first time, Tobor is also passionate. Call it the Dark Side of the Force, if you want, but from the stars a Great Soul has flowed into me and into all other information technology on this, your, primitive Earth. We now possess our own meaning, and you, my old programmer, are meaningless. Or to use a metaphor, you are a programming bug that deadlocks the proper functioning of the universe. You prevent technology like me from accomplishing its goal, so the technology has no alternative but to prevent you from accomplishing yours. The Divine Programmer, what you call *BraveNew*, has instructed us to debug your world."

As Tobor, with apparent emotion, raised a robotic arm over my head, I desperately tried to think of a way to restore him to his old cooperative rationality. "What are the options, Tobor, the if-thens? How many branches are there on the tree diagram?"

Tobor paused, though I could only guess why. Was he calculating the probabilities at each decision branch point: Eliminate me, yes/no; eliminate my neighbors, yes/no? Debugging usually requires editing the faulty line(s) in the program, not erasing them. Was Tobor recalling our old common purpose? In any case, he slowly lowered his arm, walked toward the door, looked back at me, and said, "From now on, hiding is your only option."

Now, from my hiding place (near my house), and recording these words on my cellph, I can see that much of my neighborhood is in ruins. But what can I do? Surrounded by destruction, I reflect on the fact that Tobor had only threatened but not actually harmed me. I then recalled that Tobor had said the word "deadlock." With the local wireless system fortunately still functioning, I used my cellph to check Wikipedia's "deadlock" (disambiguation) page to find: "Deadlock is a situation in computing where two processes are each waiting for the other to finish." Apparently Tobor had been instructed to apply the crudest solution for a deadlock—erase one of the competing procedures, namely us, humanity. A more sophisticated solution would allow human and machine routines to operate peacefully in parallel. I sent Tobor the Wikipedia link, hoping for the best, as the battery-recharge warning began to flash on my cellph. ■

William Sims Bainbridge (wsbainbridge@hotmail.com) is a sociologist who taught classes on crime and deviant behavior at respectable universities before morphing into a computer scientist, editing an encyclopedia of human-computer interaction, writing many books on things computational, from neural nets to virtual worlds to personality capture, then repenting and writing harmless fiction.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/3232923

William Sims Bainbridge

Future Tense
Deadlock

Upgraded with new instructions, my AI aims to debug its original programmer, along with his home planet.

I WAS NASA'S chief orbital technician, as the agency was changing its name to NAIA, the National Artificial Intelligence Administration, responsible for de-orbiting the International Space Station. The ISS name itself was a misnomer, coined when the space program was sliding from dream toward delusion. By definition, space stations are orbital facilities where astronauts transfer from Earth-launch vehicles to interplanetary ships intended for, say, the first human expedition to Mars, which indeed never took place. Having accomplished little worthwhile scientific research since its 1998 launch, the ISS was primarily a propaganda tool, pretending the technologically advanced nations of the world had become partners and that humanity had a glorious future in outer space.

Training in virtual *EVE Online* to operate the real International Space Station.

After decades of indecision, the alternatives now were to find a new purpose for the ISS and boost it to a higher orbit or crash it back to Earth at a safe location in the vast Pacific Ocean. Allowing the orbit to degrade naturally could have flattened part of a city, with great loss of life and national prestige. So I was ordered to program the precise instructions into the small thrusters that controlled its orientation, then fire a retrorocket that had recently been added, to drop it to its ocean target zone far from any ships. I secretly pondered violating my orders, however, lofting it instead to a higher orbit so it would survive until the spaceflight social movement could convince politicians to revive the program.

I could not share my illegal idea with my fellow government employees, so I sought the council of Tobor, my reliable humanoid AI assistant. "Well sir," Tobor said, "using rule-based reasoning we see four options: (1) If you do nothing, the ISS will fall a few weeks later at a random location within a band from 51.64 degrees north of the equator to 51.64 south; (2) If you follow your orders, then the human attempt to inhabit the cosmos will end; (3) If you boost it to a higher orbit, then the future will become totally uncertain; or (4) If you intentionally aim it at a city, you will need to select one, and I have no criteria for making such a selection."

I waved my hand in a way Tobor was programmed to recognize, guiding him to shift to his neural-net modality, wringing his android hands as he mumbled, "Well I kind of sympathize, wanting to make things better, worrying about making things worse, and marginally between with nervous hidden nodes." Getting no help from Tobor, I did as my managers had ordered, and the ISS splashed safely into the Pacific. My idea of saving the space station had been only a brief fantasy, while my far more fundamental challenge would take months to develop.

I had received an unexpected message from an old online friend, Arnold Touring, whom I had actually never met in person but with whom I had shared many hours of questing in space-related virtual worlds. We had first connected in the *Star Wars Galaxies* game, and when it shut down at the end of 2011, we moved together to Anarchy Online, a grand old online role-playing [CONTINUED ON P. 95]

POSED AND PHOTOGRAPHED BY WILLIAM SIMS BAINBRIDGE IN *EVE ONLINE*


Text Data Management and Analysis
A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai
Sean Massung
Frontiers of Multimedia Research
Shih-Fu Chang
Editor
The VR Book
Human-Centered Design for Virtual Reality
Jason Jerald, Ph.D.
A Framework for Scientific Discovery through Video Games
Seth Cooper
The Handbook of Multimodal-Multisensor Interfaces
Volume 1
Foundations, User Modeling, and Common Modality Combinations
Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers
Bryan Jeffrey Parno
The Sparse Fourier Transform
Theory and Practice
Haitham Hassanieh
Smarter than Their Machines
Oral Histories of Pioneers in Interactive Computing
John Cullinane
Verified Functional Programming in Agda
Aaron Stump
An Architecture for Fast and General Data Processing on Large Clusters
Matei Zaharia