

COMMUNICATIONS

CACM.ACM.ORG

OF THE

ACM

07/2020 VOL.63 NO.07

Domain-Specific Hardware Accelerators

Spectre Attacks

A Domain-Specific Supercomputer for
Training Deep Neural Networks

The Quantum Threat

A Computational Lens on Economics

Association for
Computing Machinery

acm



SIGGRAPH THINK
2020 S2020.SIGGRAPH.ORG BEYOND

THINK BEYOND

[S2020.SIGGRAPH.ORG](https://s2020.siggraph.org)

SIGGRAPH 2020 offers inspiration, putting the latest in art and tech at your fingertips. Discover inspiring content that demonstrates the latest advancements in computer graphics and interactive techniques research, education, applications and entertainment.

The 47th International Conference & Exhibition
on Computer Graphics and Interactive Techniques



Sponsored by ACM SIGGRAPH



ACM BOOKS Collection II

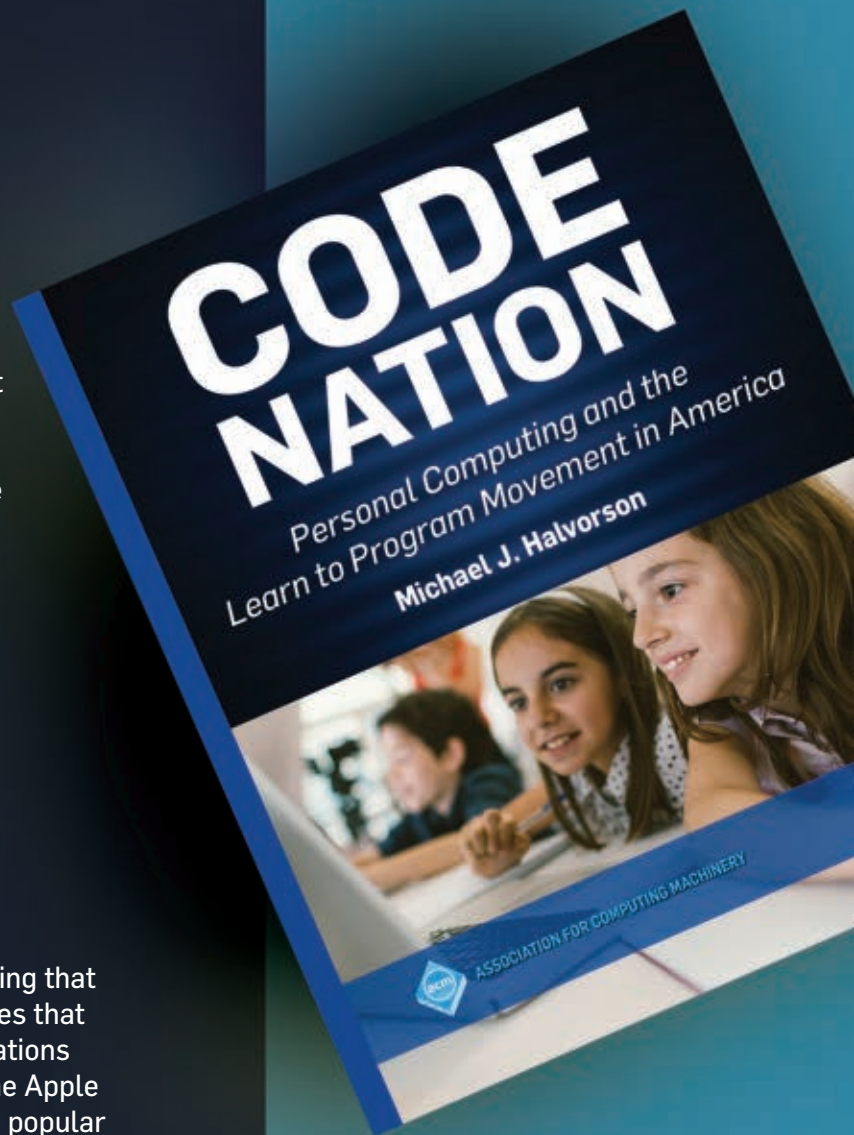
Code Nation explores the rise of software development as a social, cultural, and technical phenomenon in American history. The movement germinated in government and university labs during the 1950s, gained momentum through corporate and counterculture experiments in the 1960s and 1970s, and became a broad-based computer literacy movement in the 1980s. As personal computing came to the fore, learning to program was transformed by a groundswell of popular enthusiasm, exciting new platforms, and an array of commercial practices that have been further amplified by distributed computing and the Internet. The resulting society can be depicted as a “Code Nation”—a globally-connected world that is saturated with computer technology and enchanted by software and its creation.

Code Nation is a new history of personal computing that emphasizes the technical and business challenges that software developers faced when building applications for CP/M, MS-DOS, UNIX, Microsoft Windows, the Apple Macintosh, and other emerging platforms. It is a popular history of computing that explores the experiences of novice computer users, tinkerers, hackers, and power users, as well as the ideals and aspirations of leading computer scientists, engineers, educators, and entrepreneurs. Computer book and magazine publishers also played important, if overlooked, roles in the diffusion of new technical skills, and this book highlights their creative work and influence.

Code Nation offers a “behind-the-scenes” look at application and operating-system programming practices, the diversity of historic computer languages, the rise of user communities, early attempts to market PC software, and the origins of “enterprise” computing systems. Code samples and over 80 historic photographs support the text. The book concludes with an assessment of contemporary efforts to teach computational thinking to young people.

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



CODE NATION

*Personal Computing and
the Learn to Program
Movement in America*

Michael J. Halvorson

ISBN: 978-1-4503-7757-7

DOI: 10.1145/3368274

Departments

- 5 **Vardi's Insights**
A Computational Lens on Economics
By Moshe Y. Vardi
-
- 7 **Career Paths in Computing**
Challenge Yourself
by Reaching for the Highest Bar
By Yosuke Ozawa
-
- 9 **Letters to the Editor**
Computing's Role
in Climate Warming
-
- 10 **BLOG@CACM**
Transitioning to Distance Learning
and Virtual Conferencing
John Arquilla considers responses to the coronavirus pandemic, while Mark Guzdial ponders the impacts of competitive enrollment.
-
- 25 **Calendar**

Last Byte

- 112 **Upstart Puzzles**
Strategic Paddling
Choosing how to best navigate turbulent current events.
By Dennis Shasha

News



- 12 **The Quantum Threat**
Cryptographers are developing algorithms to ensure security in a world of quantum computing.
By Gregory Mone
-
- 15 **Your Wish Is My CMD**
Artificial intelligence could automate software coding.
By Neil Savage
-
- 17 **Reducing and Eliminating E-Waste**
We need to mitigate the environmental impact of disposing of electronics at their end of useful life.
By Keith Kirkpatrick

Viewpoints

- 20 **Legally Speaking**
AI Authorship?
Considering the role of humans in copyright protection of outputs produced by artificial intelligence.
By Pamela Samuelson
-
- 23 **Economic and Business Dimensions**
Proposal: A Market for Truth to Address False Ads on Social Media
Guaranteeing truth in advertising.
By Marshall W. Van Alstyne
-
- 26 **Computing Ethics**
For Impactful Community Engagement: Check Your Role
Toward a more equitable distribution of the benefits of technological change.
By Kathleen H. Pine, Margaret M. Hinrichs, Jieshu Wang, Dana Lewis, and Erik Johnston
-
- 29 **Viewpoint**
Consumers vs. Citizens in Democracy's Public Sphere
Attempting to balance the challenging trade-offs between individual rights and our obligations to one another.
By Allison Stanger
-
- 32 **Viewpoint**
Call For A Wake Standard for Artificial Intelligence
Suggesting a Voice Name System (VNS) to talk to any object in the world.
By Brian Subirana



Practice

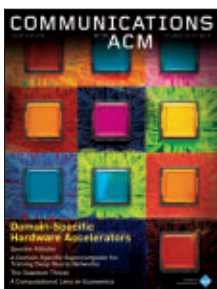


40

36 **The Best Place to Build a Subway**
Building projects despite
(and because of) existing
complex systems.
By Pat Helland

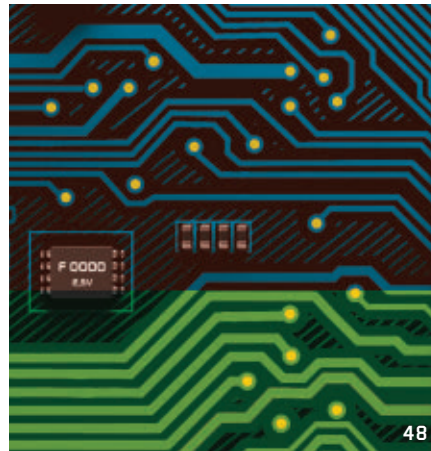
40 **Demystifying Stablecoins**
Cryptography meets monetary policy.
*By Jeremy Clark, Didem Demirag,
and Seyedehmahsa Moosavi*

Articles' development led by [acmqueue](https://queue.acm.org)
queue.acm.org



About the Cover:
Domain-specific accelerators are one of the few ways to continue scaling the performance and efficiency of computing hardware. This month's cover story (p. 48) explores the effectiveness and future of DSAs. Cover by Matt Herring, using imagery by Bet_Noire/Getty Images.

Contributed Articles



48

48 **Domain-Specific Hardware Accelerators**
DSAs gain efficiency from specialization and performance from parallelism.
By William J. Dally, Yatish Turakhia, and Song Han



Watch the authors discuss this work in this exclusive *Communications* video.
<https://cacm.acm.org/videos/domain-specific-accelerators>

58 **The Data Science Life Cycle: A Disciplined Approach to Advancing Data Science as a Science**
A cycle that traces ways to define the landscape of data science.
By Victoria Stodden

67 **A Domain-Specific Supercomputer for Training Deep Neural Networks**
Google's TPU supercomputers train deep neural networks 50x faster than general-purpose supercomputers running a high-performance computing benchmark.
By Norman P. Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson

Review Articles

80 **Some Simple Economics of the Blockchain**
Blockchain technology can shape innovation and competition in digital platforms, but under what conditions?
By Christian Catalini and Joshua S. Gans



Watch the authors discuss this work in this exclusive *Communications* video.
<https://cacm.acm.org/videos/economics-of-the-blockchain>

Research Highlights

92 **Technical Perspective**
Why 'Correct' Computers Can Leak Your Information
By Mark D. Hill

93 **Spectre Attacks: Exploiting Speculative Execution**
By Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom

102 **Technical Perspective**
ASIC Clouds: Specializing the Datacenter
By Parthasarathy Ranganathan

103 **ASIC Clouds: Specializing the Datacenter for Planet-Scale Applications**
By Michael Bedford Taylor, Luis Vega, Moein Khazraee, Ikuo Magaki, Scott Davidson, and Dustin Richmond



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.

Executive Director and CEO
Vicki L. Hanson
Deputy Executive Director and COO
Patricia Ryan
Director, Office of Information Systems
Wayne Graves
Director, Office of Financial Services
Darren Ramdin
Director, Office of SIG Services
Donna Cappel
Director, Office of Publications
Scott E. Delman

ACM COUNCIL
President
Gabriele Kotsis
Vice-President
Jack Feigenbaum
Secretary/Treasurer
Elisa Bertino
Past President
Cherri M. Pancake
Chair, SGB Board
Jeff Jortner
Co-Chairs, Publications Board
Jack Davidson and Joseph Konstan
Members-at-Large
Nancy M. Amato; Tom Crick;
Susan Dumais; Mehran Sahami;
Alejandro Saucedo
SGB Council Representatives
Sarita Adve and Jeanna Neeffe Matthews

BOARD CHAIRS
Education Board
Mehran Sahami and Jane Chu Prey
Practitioners Board
Terry Coatta

REGIONAL COUNCIL CHAIRS
ACM Europe Council
Chris Hankin
ACM India Council
Abhiram Ranade
ACM China Council
Wenguang Chen

PUBLICATIONS BOARD
Co-Chairs
Jack Davidson and Joseph Konstan
Board Members
Phoebe Ayers; Nicole Forsgren; Chris Hankin;
Mike Heroux; Nenad Medvidovic;
Tulika Mitra; Michael L. Nelson;
Sharon Oviatt; Eugene H. Spafford;
Stephen N. Spencer; Divesh Srivastava;
Robert Walker; Julie R. Williamson

ACM U.S. Technology Policy Office
Adam Eisgrau
Director of Global Policy and Public Affairs
1701 Pennsylvania Ave NW, Suite 200,
Washington, DC 20006 USA
T (202) 580-6555; acmpo@acm.org

Computer Science Teachers Association
Jake Baskin
Executive Director

STAFF
DIRECTOR OF PUBLICATIONS
Scott E. Delman
cacm-publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Lawrence M. Fisher
Web Editor
David Roman
Editorial Assistant
Danbi Yu

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Production Manager
Bernadette Shade
Intellectual Property Rights Coordinator
Barbara Ryan
Advertising Sales Account Manager
Ilia Rodriguez

Columnists
David Anderson; Michael Cusumano;
Peter J. Denning; Mark Guzdial;
Thomas Haigh; Leah Hoffmann; Mari Sako;
Pamela Samuelson; Marshall Van Alstyne

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

WEBSITE
<http://cacm.acm.org>

WEB BOARD
Chair
James Landay
Board Members
Marti Hearst; Jason I. Hong;
Jeff Johnson; Wendy E. Mackay

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

ACM ADVERTISING DEPARTMENT
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 626-0686
F (212) 869-0481

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

Media Kit acmmegasales@acm.org

Association for Computing Machinery (ACM)
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 869-7440; F (212) 869-0481

EDITORIAL BOARD
EDITOR-IN-CHIEF
Andrew A. Chien
aic@cacm.acm.org
Deputy to the Editor-in-Chief
Morgan Denlow
cacm.deputy.to.aic@gmail.com
SENIOR EDITOR
Moshe Y. Vardi

NEWS
Co-Chairs
Marc Snir and Alain Chesnais
Board Members
Tom Conte; Monica Divitini; Mei Kobayashi;
Rajeev Rastogi; François Sillion

VIEWPOINTS
Co-Chairs
Tim Finin; Susanne E. Hambrusch;
John Leslie King
Board Members
Terry Benzel; Michael L. Best; Judith Bishop;
Lorrie Cranor; Boi Falting; James Gimmelman;
Mark Guzdial; Haym B. Hirsch; Anupam Joshi;
Richard Ladner; Carl Landwehr; Beng Chin Ooi;
Francesca Rossi; Len Shustek; Loren Terveen;
Marshall Van Alstyne; Jeannette Wing;
Susan J. Winter

□ PRACTICE
Co-Chairs
Stephen Bourne and Theo Schlossnagle
Board Members
Eric Allman; Samy Bahra; Peter Bailis;
Betsy Beyer; Terry Coatta; Stuart Feldman;
Nicole Forsgren; Camille Fournier;
Jessie Frazelle; Benjamin Fried; Tom Killalea;
Tom Limoncelli; Kate Matsudaira;
Marshall Kirk McKusick; Erik Meijer;
George Neville-Neil; Jim Waldo;
Meredith Whittaker

CONTRIBUTED ARTICLES
Co-Chairs
James Larus and Gail Murphy
Board Members
Robert Austin; Kim Bruce; Alan Bundy;
Peter Buneman; Jeff Chase;
Premkumar T. Devanbu; Jane Cleland-Huang;
Yannis Ioannidis; Trent Jaeger; Somen Jha;
Gal A. Kaminka; Ben C. Lee; Igor Markov;
Lionel M. Ni; Doina Precup; Shankar Sastry;
m.c. schraefel; Ron Shamir; Hannes Werthner;
Reinhard Wilhelm

RESEARCH HIGHLIGHTS
Co-Chairs
Azer Bestavros, Shriram Krishnamurthi,
and Orna Kupferman
Board Members
Martin Abadi; Amr El Abbadi;
Animashree Anandkumar; Sanjeev Arora;
Michael Backes; Maria-Florina Balcan;
David Brooks; Stuart K. Card; Jon Crowcroft;
Alexei Efros; Bryan Ford; Alon Halevy;
Gernot Heiser; Takeo Igarashi;
Srinivasan Keshav; Sven Koenig;
Ran Libeskind-Hadas; Karen Liu; Greg Morrisett;
Tim Roughgarden; Guy Steele, Jr.;
Robert Williamson; Margaret H. Wright;
Nicholai Zeldovich; Andreas Zeller

SPECIAL SECTIONS
Co-Chairs
Sriram Rajamani, Jakob Rehof, and Haibo Chen
Board Members
Tao Xie; Kenjiro Taura; David Padua

ACM Copyright Notice
Copyright © 2020 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@hq.acm.org or fax (212) 869-0481.

For other copying of articles that carry a code at the bottom of the first or last page or screen display, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center; www.copyright.com.

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, 1601 Broadway, 10th Floor New York, NY 10019-7434 USA. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER
Please send address changes to *Communications of the ACM*
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA

Printed in the USA.



Association for Computing Machinery





Moshe Y. Vardi

DOI:10.1145/3402561

A Computational Lens on Economics

THE COVID-19 PANDEMIC is a dual crisis. On one hand, it is a global health crisis with millions of cases and hundreds of thousands of deaths. At the same time, decisions by individuals and governments in response to the pandemic have led to a severe economic slowdown, the likes of which has not been seen since the Great Depression in the 20th century. But, as I wrote in a May 2020 column, economics can be argued to be one of the roots of this dual crisis. I quoted William Galston, who wrote: “What if the relentless pursuit of efficiency, which has dominated American business thinking for decades, has made the global economic system more vulnerable to shocks?” This relentless pursuit of efficiency prevented us from investing in getting ready for a pandemic, in spite of many warnings over the past several years, and pushed us to develop a global supply chain that is quite far from being resilient. Does computer science have anything to say about the relentless pursuit of economic efficiency? Quite a lot, actually.

Economic efficiency means goods and factors of production are distributed or allocated to their most valuable uses and waste is eliminated or minimized. Free-market advocates argue that through individual self-interest and freedom of production as well as consumption, economic efficiency is achieved and the best interest of society, as a whole, are fulfilled. But efficiency and optimality should not be conflated. A fundamental theorem in economics states that under certain assumptions a market will tend toward a competitive, Pareto-optimal equilibrium; that is, economic efficiency is achieved. But how well does such an equilibrium serve the best interest of society?

In 1999, Elias Koutsoupias and Christos Papadimitriou undertook to study the optimality of equilibria from a computational perspective. In the analysis of algorithms, we often compare the performance of two algorithms (for example, optimal vs. approximate or offline vs. online) by studying the ratio of their outcomes. Koutsoupias and Papadimitriou applied this perspective to the study of equilibria. They studied systems in which non-cooperative agents share a common resource, and proposed the ratio between the worst possible Nash equilibrium and the social optimum as a measure of the effectiveness of the system. This ratio has become known as the “Price of Anarchy,” as it measures how far from optimal such non-cooperative systems can be. They showed that the price of anarchy can be arbitrarily high, depending on the complexity of the system. In other words, economic efficiency *does not* guarantee the best interests of society, as a whole, are fulfilled.

A few years later, Constantinos Daskalakis, Paul Goldberg, and Papadimitriou asked how long it takes until economic agents converge to an equilibrium. By studying the complexity of computing mixed Nash equilibria, they provide evidence that there are systems in which convergence to such equilibria can take an exceedingly long time. The implication of this result is that economic systems are very unlikely ever to be in an equilibrium, because the underlying variables, such as prices, supply, and demand are very likely to change while the systems are making their slow way toward convergence. In other words, economic equilibria, a central concept in economic theory, are mythical rather than real phenomena.

This is not an argument against free markets, but it does oblige us to view them through a pragmatic, rather than ideological, lens.

But one does not need too sophisticated analysis to conclude that individual self-interest—expressed in an extreme form by the “Greed is good” speech in the 1987 movie *Wall Street*—does not necessarily lead to an optimal outcome. After all, every computer-science graduate has learned about “greedy algorithms,” which make the locally optimal choice at each stage with the intent of finding a global optimum. While such algorithms sometimes do yield a global optimum, they most typically do not. In fact, designing algorithms that do find global optima is a major topic of algorithmic research.

Our digital infrastructure, which has become a key component of the economic system in developed countries, is one of the few components that did not buckle under the stress of COVID-19. Indeed, last March many sectors of our economy switched in haste to the WFH mode, “working from home.” This work from home, teach from home, and learn from home was enabled (to an imperfect degree, in many cases) by the Internet. From its very roots of the Arpanet in the 1960s, resilience, enabled by seemingly inefficient redundancy, was a prime design goal for the Internet. Resilience via redundancy is one of the great principles of computer science. Pay attention, economics!

Follow me on Facebook and Twitter. 

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

Copyright held by author.

ACM Gordon Bell Special Prize for HPC-Based COVID-19 Research

Call for Nominations

The Gordon Bell Special Prize for High Performance Computing-Based COVID-19 Research will be awarded in 2020 and 2021 to recognize outstanding research achievement towards the understanding of the COVID-19 pandemic through the use of high performance computing.

The purpose of the award is to recognize the innovative parallel computing contributions towards the solution of the global crisis. Nominations will be selected based on performance and innovation in their computational methods, in addition to their contributions towards understanding the nature, spread and/or treatment of the disease.

Teams may apply for the award. Nominations will be evaluated on the basis of the following considerations:

- Evidence of important algorithmic and/or implementation innovations
- Clear improvement over the previous state of the art
- Performance is not dependent on an architecture that is specialized or cannot be replicated
- Detailed performance measurements demonstrate the submission's claims in terms of scalability (strong as well as weak scaling), time to solution, and efficiency in using bottleneck resources (such as memory size or bandwidth, communications bandwidth, I/O), as well as peak performance.
- Achievement is generalizable, in the sense that other scientists can learn and benefit from the innovations
- Although solving an important scientific or engineering challenge is important to demonstrate/justify the work, scientific outcomes alone are not sufficient for this prize.

Financial support of this \$10,000 award is provided by Gordon Bell, a pioneer in high performance and parallel computing.

Nominations for the 2020 award are due on October 8, 2020.

**For more information and to
submit nominations, please visit:**

<https://awards.acm.org/bell/covid-19-nominations>





CAREER PATHS IN COMPUTING

DOI:10.1145/3399746

Computing enabled me to . . .

Challenge Yourself by Reaching for the Highest Bar



NAME

Yosuke Ozawa

BACKGROUND

Born and raised in Miyagi, Japan

CURRENT JOB TITLE/EMPLOYER

CEO, Epistra Inc.

EDUCATION

**Ph.D. Systems Biology,
University of Keio**

CHALLENGE YOURSELF AND reach for the highest bar. If you succeed, keep pushing the boundaries.” This is what my friend Hassan Hajji advised when I started my career at IBM Research Tokyo in 2002, and these words have been a guiding force in my career ever since.

At IBM, I was challenged to learn as much as possible about the research process in an industrial lab (prototyping ideas, patenting, publishing results), and it dovetailed nicely with my desire to work toward a Ph.D. in systems biology.

After receiving my doctorate, which allowed me to enhance my skills in computational and mathematical analysis to understand complex biological systems, I was ready for a new challenge. I left Japan to work in the U.K. at

a small startup, ecrebo,^a which provides a coupon-issuing system for retailers who seek to attract customers based on their individual purchasing habits.

I was responsible for developing a backend server for the coupon system. It had to be able to analyze the contents of the receipt, determine whether it met the conditions for issuing the coupon, and return it within three seconds, including communication time with the POS system. A fun challenge, right? Well, there was a catch: Marks & Spencer signed up for a trial to start in two months. With very limited time, I buckled down and developed a simple, fast system in a month by devising a new data structure for the coupons. That gave me another month for testing and refinement. To my relief, the system worked without any bugs, processing millions of transactions from the very first day.

I am proud to say ecrebo grew rapidly. In 2015, the *Financial Times*’ list of fastest-growing companies ranked it 17th in U.K. and 83rd in Europe.^b The system I built is still in use today.

Ready for a new challenge, I turned my attention to starting my own company. I knew I wanted to help find solutions to critical problems facing society and to put my background in computational biology to use. I returned to Japan and found Epistra.^c My goal was to reduce the time and cost of research and development products involving life sciences. Typically, R&D for life-critical drugs can take more than 15 years and cost around \$480 million.

a <https://www.ecrebo.com/>

b <https://ig.ft.com/ft-1000/>

c A combination of epistemic, epic, and straggle. <https://www.epistra.jp/>

For example, vaccines for many tropical diseases have been neglected in developing countries, where most citizens cannot afford them. Large pharmaceutical companies are reluctant to invest in the development of drugs when the financial return is dubious. For new, infectious diseases such as COVID-19, however, quick development of drugs for treatment and prevention is acute.

Epistra created a software to accelerate R&D processes in the life sciences using AI and robotics. Specifically, we specialize in the development of automatic optimization software and services for sample preparation, pre-treatment, and setting of measurement equipment. Our technical approach combines evaluating results using image recognition and experimental design using mathematical optimization, allowing us to reduce the number of trials and errors. Our team succeeded in improving the differentiation efficiency of induced pluripotent stem (iPS) cells into retinal pigment epithelium in a joint study with Dr. Masayo Takahashi, one of the pioneers in the field of human study of iPS cell-based therapy. Potential applications include therapies for heart disease and Parkinson’s disease.

In this year’s Extreme Tech Challenge,^d Epistra was selected one of the top 10 companies in Japan using technology to develop solutions to global challenges with sustainable development goals. I know this is only the beginning of my work, and I will continue to tackle difficult problems in the life sciences using cutting-edge technologies in biological and computer sciences. ■

d <https://re-how.net/product/398753/>

volume

01

number

01

FIRST

ISSUE

PUBLISHED

*Digital Threats:
Research and Practice*
is now available in
the ACM Digital Library



Digital Threats: Research and Practice (DTRAP) is a peer-reviewed open access journal that targets the prevention, identification, mitigation, and elimination of digital threats. DTRAP aims to bridge the gap between academic research and industry practice. Accordingly, the journal welcomes manuscripts that address extant digital threats, rather than laboratory models of potential threats, and presents reproducible results pertaining to real-world threats.



Association for
Computing Machinery

<https://dtrap.acm.org>

DOI:10.1145/3402933

Computing's Role in Climate Warming

AS A COMPUTER scientist, I was embarrassed to read the Viewpoint “Conferences in an Era of Expensive Carbon” (Mar. 2020) from four fellow computer scientists. If these scholars truly believe what they write, that humans are causing the planet to warm, that they are not just eagerly joining the herd, then they need to show the way. The job of computer scientists is to make bits dance on the head of a circuit and that requires the field’s lifeblood—electricity. Since 63% of U.S. electricity is currently generated from fossil fuels, they need to reduce their electricity usage by that amount. They must take seriously their own stated beliefs and stop the planet from further warming. They need to immediately cut their time on the computer by 63%. In the classroom, they need to turn the projector off for most of their lectures. At home, they need to shut off lights and appliances for a majority of each day. If they drive an electric car, they need to reduce trip lengths by two-thirds. If they don’t take these CO₂-mitigating steps, then they don’t really believe there is a problem. And the solution they propose becomes similar to many academic exercises that professors put their students through.

Daniel Ouellette, Detroit, MI, USA

Authors’ Response

We entirely agree that everyone concerned about climate change must urgently translate their concern into action. To be effective, such action should focus on the biggest opportunities for reduction. Air travel to conferences is the biggest contributor to our own individual carbon footprints, by a huge margin; we suspect the same is true for many scientific researchers and academics. However, since conferences also serve an extremely valuable social function, it would be rash to advocate simply taking them away. Instead, organizations like ACM should help create a future in which their scientific meetings are sustainable. Our proposals for publicizing

carbon footprints and putting a price on carbon are steps in this direction.

Benjamin C. Pierce, Philadelphia, PA, USA

Jens Palsburg, Los Angeles, CA, USA

Michael Hicks, College Park, MD, USA

Crista Lopes, Irvine, CA, USA

Editor-in-Chief’s Response

I applaud the advocacy of real change to reduce the carbon impact of the computing community. We have many constructive dimensions to consider, including reducing travel for conferences (virtual!), but also directly in the carbon emissions tied to both the creation of computing hardware and its operation (see “What Do DDT and Computing Have in Comment?” (June 2020, p. 5) and “Owning Computings Environmental Impact,” (Mar. 2019, p. 5) for ideas on how to reduce carbon emissions while society continues to reap growing benefits from computing. Ouellette further suggests conservation techniques; these can be effective, but “bright green” approaches that seek to maintain and expand activity, while reducing environmental damage are certainly easier to adopt.

Andrew A. Chien, Chicago, IL, USA

Location, Location, Location

In the article “Can You Locate Your Locations Data?” (Sept. 2019, p. 19) Sarah Underwood discussed several approaches to mitigation, though indicating there’s no one solution. An approach not discussed, but which would certainly provide added privacy for those more aware of and concerned, lies in the phone itself.

Two apps not at present available, as far as I know, could help. One would limit access to GPS data to privileged apps that do not send that data outside the device. The second would disable the communications capabilities altogether, so there would be no tracking through network transactions, thus reducing the phone (or tablet) to a pocket computer.

As an aside, there should be corresponding capabilities to disable the camera(s) and microphone.

Navigation could be achieved either by accessing a map application prior to

travel and saving it, or downloading maps, such as those provided for personal GIS/GPS systems, or loading a basic GIS and its maps for local navigation on site with no outside communication.

For those who do care deeply about such issues, these two capabilities would greatly mitigate the perceived risks in being tracked on personal errands or sensitive professional activities and would not depend on the ambiguities—or outright lacunae—in the provisions of multiple corporate privacy policies. Moreover, they could be implemented without extensive legal or legislative effort, which can be difficult and/or expensive to enforce, and often of questionable effectiveness.

N.L. Sizemore, Sierra Vista, AZ, USA

Editor-in-Chief’s Response

An interesting technical solution which begs the question, if the change is not compelled by government, who has the incentive to drive its creation and widespread adoption? Given the economics of scale, and the current situation that privacy seems only to be a compelling concern for a minority (for example, DuckDuckGo remains a minor player), what could tip the scales and drive creation of a vibrant growing alternative ecosystem?

Andrew A. Chien, Chicago, IL, USA

An Army Lesson

J. Paul Reed’s article “Beyond the ‘Fix-It’ Treadmill” (May 2020, p. 58) was quite interesting and certainly a step in the right direction. Although it post-dates my retirement from the U.S. Army by 16 years, you might find interest in the report: “Army Lessons Learned Program.”¹ In my day, we used to simply do AARs—After Action Reviews. Technical, communication, and people vectors are involved when done correctly.

Reference

1. Department of the Army. Army Lessons Learned Program, 2017; https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/ARN2887_AR11-33_Web_FINAL.pdf

Carl A. Singer, Passaic, NJ, USA

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.



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

DOI:10.1145/3398386

<http://cacm.acm.org/blogs/blog-cacm>

Transitioning to Distance Learning and Virtual Conferencing

John Arquilla considers responses to the coronavirus pandemic, while Mark Guzdial ponders the impacts of competitive enrollment.



John Arquilla
The COVID Catalyst

<https://bit.ly/35eGSpq>

April 27, 2020

The coronavirus pandemic has, like its predecessors from the Black Plague to the Spanish Flu, once again demonstrated the great vulnerability of social and economic systems to microbes. Yet, it may be that there is an important difference this time around.

The bubonic plague of the 14th century completely disrupted the huge Mongol Empire and killed off a third of the population of Europe. The flu that hit at the end of World War I killed tens of millions as well. But this time around, advances in medicine—and skillful use of information technology for hotspot detection and backtracking contacts, among other functions—will, in addition to sheltering in place, keep the cost of corona in lives lost relatively low. That the world has recourse to such boons to lifesaving ought to be seen as very good news, despite the huge economic costs and serious psychological damage inflicted by the virus. And, if we're

able to look a little deeper, there may be even more good news. For beyond reducing the toll taken in lives by COVID, some mitigating measures put in place may have profoundly beneficial effects if they are continued, or perhaps even expanded upon. Three areas of activity come quickly to mind.

The most obvious improvement, literally visible already, has been to air quality in metropolitan areas. Vast numbers of people can work from home—in many places around the world—sharply reducing pollution caused by commuting in automobiles. Clearly, many people will still have to go to physical workplaces outside the home, but all who don't have to should keep doing their jobs remotely. This will have tremendous benefit for those living in the urban areas blighted by pollution, and also will contribute usefully to the larger fight against global warming.

The response of the educational sector is less well developed at this point, but the use of networking systems has proved there is a way to continue to educate via distance learning. This is surely less effective at the elementary level, but

the possibilities abound for high school, college-level and postgraduate education. During this quarter, I am remote-teaching master's students at the military school where I work, and find that there is in some ways a deeper, more tutorial quality that has emerged in the absence of the formal classroom setting.

This experience has caused me to muse about teaching my regular seminars “from a distance” as well, but with more than the usual flat-screen TV-style connection. Instead, we should pursue the kind of immediacy that virtual reality provides. This is certainly an area of advancing technology that could have profound effects on education, at many levels. VR might even prove an interesting way to bring actors and audiences together in “theaters” made of bits and bytes. Think of concerts, too, and a range of other kinds of group activities that can be conducted via well-designed VR.

The third area of opportunity that COVID may catalyze is the possibility of networking medical research. If Metcalfe's Law, about the power of networks being a strong multiple, perhaps the square, of the number of interconnected nodes, then it is time to put in place a global medical network. To some extent, this is already being done, but it can be built upon. What we don't want to see is what is happening right now: medical research activities are being subjected to a steady stream of hacks. Officials of the U.S. government—including the Secretary of State—have gone so far as to accuse particular foreign powers of being behind these activities. Of course, perpetrator

ambiguity remains a problem, and these actions might also be by criminals who intend to sell whatever they steal.

We have international policing entities that surely will need to be increasingly attentive to this threat. But the larger point is about the need for nations to begin to think less in terms of power as gained through the control of information, and more in terms of the value created, for all, by *sharing* it. And not only as relates to medical research. Imagine the power of the “global mind” that has the potential to emerge. There is scarcely a problem bedeviling the world that would not succumb to this kind of networked, collective intelligence.

Beyond the three areas of opportunity discussed here, there are surely other ways in which COVID can catalyze progress—by reshaping governance and statecraft while improving global public discourse in more participatory ways, and with greater immediacy, for example. But that is a far reach. For now, the focus should be on how the response to COVID has opened up the possibility of making quantum leaps in environmental protection, education, and global health research.

Progress in each of these areas, however, is wholly dependent upon robust cybersecurity. Without a solid virtual foundation, the ability to move forward in any of these areas will always be held at risk. And in a world still too wedded to the firewall-and-antiviral paradigm—rather than, say, to ubiquitous use of strong crypto and cloud computing—what COVID catalyzes may end up producing a fizzle instead of a fountainhead for transformation.



Mark Guzdial
Students Get the Idea
They're Unwanted
When There
Are Enrollment
Barriers: Touring

the Best of SIGCSE 2020

<https://bit.ly/2KZcjdY>

May 2, 2020

The ACM Special Interest Group on CS Education (SIGCSE) cancelled its technical symposium in Portland the morning the conference was scheduled to start. I was there. My wife (Barbara Ericson, <https://bit.ly/3c5sjag>) and I arrived on the evening of Wednesday, March 11, a half-hour before Oregon's governor

banned large meetings. The next morning, I got the announcement that the conference was closed. I visited with others who had arrived for the first day (all of us observing social distancing), met with collaborators during the day, then flew back home on Friday, March 13.

While it was disappointing that the conference was cancelled, I'm happy to see that all the papers are posted in the ACM Digital Library (at <https://bit.ly/3dbNXd0>). I've been spending time looking through the papers, wishing I'd have had the opportunity to hear the presentations and talk to the authors.

Let me tell you about the easiest paper to recommend: one of the Best Paper awardees for CS Education Research: Competitive Enrollment Policies in Computing Departments Negatively Predict First-Year Students' Sense of Belonging, Self-Efficacy, and Perception of Department, by An Nguyen and Colleen M. Lewis of Harvey Mudd College (<https://bit.ly/2W5Yyk1>). The punchline of the paper is in the title, and might be described as “if you send students the message that they're unwanted, they're going to feel unwanted.”

Nguyen and Lewis look at a dataset from the Computing Research Association based on a survey of 1,245 first-year students. They looked at four outcome measures:

- ▶ *Perception of department as welcoming* indicated by agreement with phrases like “My department cares about its students.”

- ▶ *Sense of Belonging* indicated by agreement with phrases like “I feel like I belong in computing.”

- ▶ *Self-efficacy* (a sense you can achieve tasks in a domain) indicated by agreement with phrases like “I am confident that I can pass my classes.”

- ▶ *Growth mindset* (a sense you can always get better at something through effort) indicated by agreement with phrases like “Anyone has the ability to learn computing and be good at it.”

They defined a department as having *competitive enrollment* if students had to apply to become a computing major, or if a student needs to meet grade thresholds (beyond simply passing) to become a computing major. They found students in departments with competitive enrollment had a lower sense of being welcomed and a lower sense of self-efficacy. If students

had prior experience, they still had a sense of belonging in computing, but that wasn't true for students who didn't have prior experience in belonging. Overall, female, Black, and Latinx first-year CS students had a lower sense of belonging and lower self-efficacy.

Departments are using competitive enrollment as a way of managing rapidly rising enrollment, or just to make sure that the students who get to the upper-level classes succeed. In any case, these moves are a *barrier* to students. The results from this paper suggest the moves are having an impact on students.

Some of the discussion about this paper on Twitter points out that the effect isn't all that strong; statistically significant, but not a large effect size. That makes sense to me. These are subtle and likely indirect effects. If a CS professor said to a student, “You'll never make it in CS. You don't belong,” and then the student consequently showed a decrease in self-efficacy and sense of belonging, we would say the mechanism would be pretty clear and the effect would be direct. In this case, the competitive enrollment barriers may not take effect until the second or third years of undergraduate, and this study is looking at all first-year students. The direct impact is maybe some form of social pressure on students (such as talking to other students facing these barriers), or the student dreading a future date when they would have to be judged. Competitive enrollment tells students not everyone is welcome, and first-year students seem to be responding to that.

We can't know the future of CS enrollment. Will CS still have huge enrollment next year, when the world is still dealing with COVID? How will competitive enrollment measures need to change in the future? This is an important study to realize there are likely effects of barriers, even for students in their first year.

There is a lot more great stuff in SIGCSE 2020. I recommend taking a stroll around the proceedings.

John Arquilla is Distinguished Professor of Defense Analysis at the United States Naval Postgraduate School. The views expressed are his alone. **Mark Guzdial** is professor of electrical engineering and computer science in the College of Engineering, and professor of information in the School of Information, of the University of Michigan.

© 2020 ACM 0001-0782/20/7 \$15.00

The Quantum Threat

Cryptographers are developing algorithms to ensure security in a world of quantum computing.

THE SECURITY OF modern communications could soon expire. The precise day, month, or year is impossible to predict because the technology capable of cracking our codes—practical, robust quantum computers—does not exist. Yet experts insist that the time to prepare is now.

“It’s beyond something you can just ignore, even though we still don’t know when it will happen,” says mathematician Michele Mosca of the Institute for Quantum Computing at the University of Waterloo in Canada. “The chance of it happening in five, 10, or 20 years is not a risk you can accept. It’s a systematic threat to the global economy, and it’s real enough that you definitely have to plan for it now.”

Today’s most successful cryptographic systems, which we depend upon to secure online transactions and communications, rely on one of two mathematical problems: factoring large numbers or finding discrete algorithms. If you’re using online encryption, buying something, or even downloading a software update for your PC, the security of that exchange probably relies on the difficulty of these mathematical challenges. For today’s computers, these problems are all but impossible to solve.

In 1994, mathematician Peter Shor described an algorithm that could unlock these codes, but it was not designed



for machines that speak in the common digital language of 1s and 0s. Shor’s algorithm is designed to run on a quantum computer—one that uses quantum bits, or qubits. These qubits could exist in a superposition of an exponential number of states, and a quantum computer running Shor’s algorithm could solve both of the difficult mathematical problems that underpin most cryptography.

While the threat is theoretical, it still has contemporary implications. A malicious actor could store a cache of email encrypted with today’s cryptographic approaches, and then use quantum computing to unlock them some 10 or 15 years in the future.

Government, academia, and industry are working diligently to prepare for this post-quantum-computing world. The German government,

for example, is funding seven initiatives, including efforts to increase collaboration between academia and industry. However, most experts currently are focused on the Post-Quantum Cryptography Standardization project being run by the U.S. National Institute of Standards and Technology (NIST). The goal of the project, which has many hallmarks of a competition but declines to be defined as such, is to spark the development, refinement, and testing of cryptographic algorithms that could maintain security in a world of quantum computers.

Living on the Edge

In one sense, the threat of quantum computers is not entirely new, as there is always a risk cryptography can be broken. “We live on the edge because none of the cryptographic systems we use are proven secure in the sense that there’s no mathematical proof that these things cannot be broken,” says Massachusetts Institute of Technology mathematician Vinod Vaikuntanathan. “A bunch of smart people try to attack them for 10 to 20 years, and if they can’t, then we say it’s probably good enough. But we’re always at risk of someone coming up with a clever algorithm for factoring large numbers or, with quantum computers, we’re at the risk of the underlying computing technology changing so dramatically that new attacks suddenly become possible.”

Since the problems on which most cryptography relies would be solvable in a post-quantum world, experts have to find a new, harder mathematical problem. Luckily, this effort has been underway for some time. “People have been trying to build post-quantum-secure cryptography for 20 years,” says cryptography researcher Vadim Lyubashevsky of IBM Research, Zurich. “This was already a mature field.”

Narrowing the Field

The aim of the NIST program is to transform this theoretical work into practical methods and results that can be widely attacked and tested. The initial call for proposals yielded 82 submissions, of which 69 were accepted. NIST researchers and members of the

“It’s not a paint job. It’s like replacing every brick in the foundation of your house. You can’t just throw on some quantum-safe fairy dust at the last minute.”

cryptography community quickly set about testing the approaches. Some were broken within a few weeks; others survived longer. When a group or individual breaks a technique, the result is announced in an NIST forum. The prize? Peer recognition. By 2019, the field narrowed to 26 candidates, and that number will be winnowed down again this year or early next.

NIST mathematician Dustin Moody, director of the program, expects approximately a dozen candidates will emerge as finalists. NIST has encouraged several teams to merge, as their approaches were similar, and some have followed these recommendations, while others remained independent. One of the more popular methods among the entries hinges on lattices, or grids, and the difficulty of finding a point close to the origin in a lattice defined by a basis of long vectors. In two dimensions, the lattice problem can be relatively simple, but “When you go to 1,000 dimensions, this becomes insanely hard,” says Vaikuntanathan. “Nobody has been able to put a dent in this problem.”

The other advantage of the lattice-based approach is that cryptography researchers have been testing it—and attacking it without success—for years. “This gives you confidence in the security,” says Moody. “It gives you confidence that the hard problem we’re going to rely on really is going to be a hard problem.”

At the same time, Moody stresses, there are no favorites, and several other

ACM Member News

WHERE CYBERSECURITY AND DATA SCIENCE MEET



ACM Fellow Bhavani Thuraisingham is a Founders Chair Professor and Executive Director of the

Cyber Security Research and Education Institute at the University of Texas at Dallas.

Thuraisingham received an undergraduate degree in mathematics and physics from the University of Ceylon (now Sri Lanka), a master’s degree in Mathematical Logic and Foundations of Computer Science at the University of Bristol in the U.K., and a Ph.D. in Theory of Computation from the University of Wales in the U.K. “I also earned a higher doctorate (D. Eng) from the University of Bristol, England, for my published work in secure data management,” she adds.

Over Thuraisingham’s nearly 40-year career, she has worked in industry at the non-profit MITRE Corp., at a federal laboratory, and for the U.S. National Science Foundation. In 2004, she joined the faculty of The University of Texas at Dallas as a professor of computer science and director of the university’s Cyber Security Research Center.

Thuraisingham’s research focus is on the intersection of cybersecurity and data science. “One area my research has centered on is applying machine learning to detect malevolent threats, such as malicious code and malware,” she says.

More recently, she has focused on adversarial machine learning, an ongoing process of introducing countermeasures against evolving threats. “Malicious threats keep updating and adapting, so cybersecurity measures have to as well.”

Thuraisingham is a passionate advocate for women in computing. She supports, motivates, and encourages women to become involved in cybersecurity, data science, and artificial intelligence by organizing conferences and workshops and giving motivational addresses.

—John Delaney

approaches have their own strengths. Mosca notes that advocates of another popular option—a code-based approach—would argue their scheme has been around and tested even longer.

Practical Considerations

The program is not merely evaluating the strength or security of the algorithms; the practical implications of real-world implementation are essential, too. For example, if the schemes are going to run on embedded systems such as Internet of Things (IoT) devices, they will need to be sufficiently lightweight that they will not demand significant compute or storage resources.

“In general, post-quantum cryptography schemes will require more resources,” says cryptographic engineer Ruben Niederhagen of the Fraunhofer Institute for Secure Information Technology (SIT) in Germany. “That’s fine if you have a large device like a smartphone or a server, but if you go down to embedded systems, which have limited computational resources and limited storage, this gets very tricky.”

Speed will be an essential quality, too. Online transactions need to be fast, so new quantum-safe algorithms should not slow down exchanges excessively, and Lyubashevsky says some post-quantum techniques may actually prove faster.

Another factor to consider will be the size of the keys that need to be exchanged. The lattice-based encryption approach, for example, might result in keys that are 8,000 bits long, instead of the 2,048-bit keys exchanged with the popular RSA technique. “If you

hardcoded that your packets are going to be less than 3,000 bits, you’re going to be in trouble,” says Lyubashevsky. Systems that have placed a limit on key size will have to be adjusted, or they will not be able to run these new cryptographic standards. Also, the exchange of keys needs to be fast enough that the operation does not time out.

Regardless of which algorithms emerge from the competition, experts say it will still take years to implement a post-quantum approach. “If I say it is five years away, and tell a random enterprise they have to fix all their systems, there is no practical way they could do it,” says Mosca. “Even a simple fix takes a long time. It’s not a paint job. It’s like replacing every brick in the foundation of your house. You can’t just throw on some quantum-safe fairy dust at the last minute. You have to really start planning and working on it now.”

This is especially true for large enterprises. Niederhagen and his Fraunhofer colleagues work closely with major companies, studying their products or product lines to determine which will need to be post-quantum secure. Automobiles, power plants, trains—each of these relies on embedded devices, and the products they are manufacturing today will likely still be operational in 15 years, at which point quantum computers could be operational.

No Silver Bullet

Niederhagen says researchers who focus more on implementation are in something of a holding pattern as

they wait for the new NIST standards to be released. While no date has been set, and no clear winner has emerged, Moody does offer some sense of the end-results.

Moody and other experts expect several algorithms to be selected, instead of a single victor. “There’s no perfect silver-bullet winner that will have all the properties everyone wants,” Moody says. But the work of the remaining teams, and the efforts of the larger cryptography community to attack their algorithms, should bring us closer to a more secure post-quantum future. “We’re all working for the same purpose,” Moody notes. “We want strong cryptography to protect against a future with quantum computers.” **Q**

Further Reading

Bernstein, D.J., Buchmann, J., and Dahmen, E. *Post-Quantum Cryptography*, Springer, 2009.

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., and Smith-Tone, D. *Report on Post-Quantum Cryptography*, NISTIR 8105.

Kaye, P., Laflamme, R., and Mosca, M. *An Introduction to Quantum Computing*, Oxford University Press, 2007.

Lyubashevsky, V., and Seiler, G. *NTTRU: Truly Fast NTRU Using NTT*, *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019.

As We Enter a New Quantum Era
<https://www.youtube.com/watch?v=vWP4LF2hz80>

Gregory Mone is a Boston-based science writer and the author, with Bill Nye, of *Jack and the Geniuses: At the Bottom of the World*.

© 2020 ACM 0001-0782/20/7 \$15.00

Milestones

ACM Members Elected to National Academy of Sciences

Eight members of ACM were among the 120 members and 26 international members recently elected to the National Academy of Sciences in recognition of their continuing achievements in original research.

The honored scientists were:

Vinton G. Cerf, vice president and chief Internet evangelist, Google Inc., Reston, VA.

Ronald Fagin, IBM Almaden Research Center, San Jose, CA.

Thomas Henzinger, university

president, Institute for Science and Technology Austria.

Yonggang Huang, Walter P. Murphy Professor of Civil and Environmental Engineering and Mechanical Engineering in the department of mechanical engineering of the McCormick School of Engineering of Northwestern University, Evanston, IL.

Elizabeth A. Kellogg, Robert E. King Distinguished Investigator at the Donald Danforth Plant

Science Center, St. Louis, MO.

Jennifer Rexford, Gordon Y.S. Wu Professor in Engineering, and chair of the department of computer science, at Princeton University, Princeton, NJ.

Jeffrey D. Ullman, Stanford W. Ascherman Professor of Computer Science (Emeritus) at Stanford University, Stanford, CA.

Bonnie Berger, associate member of The Broad Institute of the Massachusetts Institute of Technology (MIT) and

Harvard University, and Simons Professor of Mathematics in the department of mathematics and professor of electrical engineering and computer science at MIT.

The National Academy of Sciences is a private, non-profit society established by an Act of Congress and charged with providing independent, objective advice to the nation on matters related to science and technology.

Your Wish Is My CMD

Artificial intelligence could automate software coding.

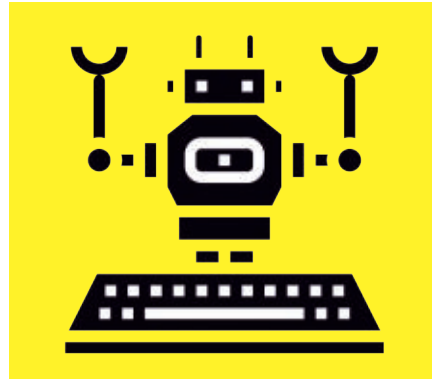
AS ARTIFICIAL INTELLIGENCE (AI) techniques advance, they are beginning to automate tasks that, until recently, only humans could perform—tasks such as translating text from one language to another or making medical diagnoses. It seems only logical to turn that computer power on computers themselves and use AI to automate programming.

In fact, computer scientists are working on just that idea, using various AI techniques to develop new methods of automating the writing of code. “The ultimate goal of this is that you would have professional software engineers not actually write code anymore,” says Chris Jermaine, a professor of computer science at Rice University in Houston, TX. Instead, the engineer would tell a computer what a piece of software should do, and the AI system would write the code, perhaps stopping along the way to pose questions to the engineer. “A software engineer becomes much more of a designer than somebody who deals with the low-level details,” Jermaine says.

Such a vision, Jermaine and other computer scientists say, lies decades in the future, and it is not entirely clear how to achieve it. Meanwhile, researchers are applying AI techniques to narrower problems and coming up with some promising solutions.

A Vintage Idea

The concept of program synthesis, in which a user specifies an intention and a programming language and the machine creates the code, actually dates back to the early days of AI in the 1950s, says Swarat Chaudhuri, an associate professor of computer science at the University of Texas, Austin. These days, most people think of statistical methods such as neural networks when they talk of AI, but back then, the field was focused on symbolic descriptions, he says. Theorem provers, which use computer programs to automatically come



up with formal mathematical proofs, exemplify that type of AI.

Combining both the symbolic and statistical approaches to AI can help to solve the challenge of program synthesis, Chaudhuri says. Say, for instance, that you want to write a program to read a file. You might start by providing a neural network with some keywords, such as “read” and “file.” The neural net could then go through a corpus of thousands of programs, perhaps as collected on GitHub, a Microsoft-owned repository of code. The neural net could identify the type of program structure associated with the keywords, providing a skeleton of what the desired program should look like.

Machine learning, though, cannot accomplish the whole task. “Neural nets are actually really bad at doing things precisely. They are definitely not able to do tasks like programming end to end,” Chaudhuri says. “Going from that high-level structural insight to a piece of code that’s going to pass a type checker, that you can paste into your code window and it’s not going to complain, that’s a leap.”

The next step is to use symbolic methods to fill in the low-level details, such as which variable to use in a particular place within the code, by searching through all the possible variables that could be placed there. Chaudhuri, then at Rice, and Jermaine developed a system that uses these methods to figure out the specifications of a program. The system, called PlinyCompute, was

funded starting in 2014 with a four-year, \$11-million grant from the U.S. Defense Advanced Projects Research Agency.

While something like PlinyCompute can identify a localized pattern such as the section of a program responsible for reading a file, nothing is yet capable of looking at the overall structure of more elaborate programs and discovering the patterns of how such smaller tasks fit together. PlinyCompute is not able to write codes longer than 50 or 60 lines. “Using machine learning to look at these kind of meta-level patterns is one thing that people just really haven’t looked at,” Jermaine says. “No one is really looking at it because it just seems so hard.”

Program synthesis is successful when it is limited to small problems in tightly defined domains, says Marc Brockschmidt, a researcher in Microsoft Research’s Programming Principles and Tools group in Cambridge, U.K. The difficulty lies in going to a more ambitious scale, because of the challenge of specifying what the programmer wants. The most common ways to tell the computer the desired outcome are either to use natural language or to show it a set of examples and ask it to learn from them. “The problem really is that both natural language and examples are a very weak way of specifying what behavior you want,” Brockschmidt says. “You wouldn’t expect that any system, no matter how far we get in research, would be able to go from ‘write an operating system’ to produce something like Windows 10 or macOS, just because when I say ‘write an operating system,’ there’s a lot of assumptions that I have and a lot of different ways of implementing this task that are not captured by my description.”

In 2017, Brockschmidt and his colleagues at Microsoft Research developed a program called DeepCoder, which performed program synthesis by having a neural network learn from a series of examples of the output that would be expected for a given input.

DeepCoder required the use of a domain-specific programming language, which contains more restrictions than a full-featured programming language. They applied their approach to challenges posed on programming competition websites and found they were able to solve some of them better than other approaches could. DeepCoder only worked on the simplest challenges, however, and Brockschmidt decided to pursue other approaches to program synthesis.

AI has begun to find its way into commercial tools for software development. So far, the most widespread use of machine learning in the software industry is for code autocompletion, Brockschmidt says. For instance, Microsoft's integrated development environment for programmers, Visual Studio, now includes IntelliCode. IntelliCode scans GitHub to identify patterns in coding, and uses what it learns to provide suggestions as the programmer types in statements. It also suggests arguments—values that are passed between programs—and tries to infer the formatting style being used, to keep the code consistent.

Eclipse, the integrated development environment for Java, also uses AI to make autocompletion suggestions, and the startup Kite does the same for Python. Another startup, DeepCode, spun out of Swiss technical university ETH Zurich, applies machine learning to reviewing software once it has been written, in order to uncover security bugs. A beta version of the company's software is available for code developed in Visual Studio.

A Sparsity of Data

One difficulty in teaching a machine to program is a lack of data. While there is plenty of existing code collected in GitHub, or in the in-house collections of companies such as Google, very little of it has labels describing the developer's intention. There may be a few keywords or some textual notes, but that is uncommon and often of limited value. "Often what the user wanted when they wrote the particular piece of code is not very well documented," says Armando Solar-Lezama, head of the computer-assisted programming group at the Massachusetts Institute of Technology, Cambridge, MA. With no way to know

the intention behind existing code, the computer cannot predict how to go from what a developer asks for to new code. If a programmer has to spend a lot of time and effort writing a formal specification of what the program should do, program synthesis loses a lot of its value.

Divining a user's intention is one key aspect of automating programming, says Justin Gottschlich, head of machine programming research at Intel Labs in Santa Clara, CA. Intel established the research program last fall to encourage the automation of programming. Gottschlich and Solar-Lezama were two of the authors of a 2018 paper describing what they call the three pillars of machine programming. The first pillar, intention, is the ability of the machine to understand the programmer's goals. Invention is the ability of the computer to write a program that accomplishes those goals. The third pillar, adaptation, is about revising the software to make it more efficient and to correct errors.

Gottschlich considers the complete automation of writing software one of the field's grand challenges, one that could take decades to achieve. "You basically give the computer an intention specified in some manner—input/output examples, natural language, whatever—and then it builds the entire piece of software for you. That is an outrageous goal," he says.

Yet there are smaller aspects of the problem where machine programming already is outperforming humans, such as in generating tests to find performance bugs in software. Bugs that degrade the efficiency of a program can be hard to spot because they are not black-and-white errors. A program with performance bugs may still run, but much slower than you want. Intel developed a program called AutoPerf, based on some of the techniques of machine programming, and was able to detect a bug in the MySQL relational database management software that was degrading its performance by nearly 70%.

In fact, one of the benefits of applying AI to writing software should be the reduction of errors, thereby increasing efficiency and cutting development costs. It can also help with a shortage of programmers. According to a 2017 survey by Code.org, a non-profit that promotes computer

education, the U.S. had more than 500,000 unfilled jobs for coders, but was producing only 50,000 computer science graduates a year.

Rather than take jobs away from people, automating software creation could free programmers to focus on the more creative parts of their jobs. A machine programming system could act as an assistant to a program designer, taking care of the nitty gritty and querying the designer about exactly what he wants. "What you could have is a magnification effect where people are able to produce more and better software," says Jermaine. "And I think it would alleviate some of those terrible problems that we have right now with the lack of engineering capacity in the modern world."

Gottschlich says AI could even open up the power of programming to people who have no training in writing code. "We really want to enable the global population to be what I'm calling 'software creators'," he says. "If we realize this dream that we're setting out to conquer, the machines would do all the programming and the humans would focus mostly on intention." ■

Further Reading

Gottschlich, J., Solar-Lezama, A., Tatbul, N., Carbin, C., Rinard, M., Barzilay, R., Amarasinghe, S., Tenenbaum, J.B., and Mattson, T.

The Three Pillars of Machine Programming, ArXiv, 2018, arXiv:1803.07244v2

Balog, M., Gaunt, A.L., Brockschmidt, M., Nowozin, S., and Tarlow, D.

DeepCoder: Learning to Write Programs, 5th International Conference of Learning Interpretations, 2017, arXiv:1611.01989v2

Zou, J., Barnett, R.M., Lorigo-Botran, T., Lua, S., Monroy, C., Sikdar, S., Teymourian, K., Yuan, B., and Jermaine, C.

PlinyCompute: A Platform for High-Performance, Distributed, Data-Intensive Tool Development, SIGMOD '18: Proceedings of the 2018 International Conference on Management of Data, doi/10.1145/3183713.3196933

Kant, N.

Recent Advances in Neural Program Synthesis, ArXiv, 2018, arXiv:1802.02353v1

Machine Programming: What Lies Ahead <https://knowledge.wharton.upenn.edu/article/ai-machine-learning/>

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

Reducing and Eliminating E-Waste

We need to mitigate the environmental impact of disposing of electronics at their end of useful life.

IT IS HARD to imagine a world without electronic devices. From servers, personal computers, and storage devices to smartphones, tablets, and wearable devices, electronic devices drive businesses, entertain and enable consumers to interact with the world, and keep the world's information and physical infrastructure networks running on a 24/7 basis. Not surprisingly, the number of electronic devices in use, particularly those that are connected to a network, continues to grow.

A 2019 Cisco VNI report forecast that on a global basis, there will be 28.5 billion networked devices in use by 2022, up from 18.0 billion in 2017, and 3.6 networked devices per capita by 2022, up from 2.4 per capita in 2017. Further, a December 2019 Deloitte study found that U.S. households own an average of 11 connected devices, including seven devices featuring screens on which to watch content, with expectations that the number of devices will increase further, overall and per person, thanks to the growth of new Internet of Things (IoT) products and applications.

While this is certainly good news for device manufacturers, content providers, and wireless service providers, there has also been enormous growth in electronic waste.

According to United Nations 2018 estimates (the most recent available), the e-waste stream has reached 50 million metric tons annually on a global basis. Managing this waste appropriately is of considerable concern; throwing electronic equipment, often containing extremely harmful chemicals and elements, into landfills or shipping them off to other countries, is simply no longer an option, due to both environmental concerns and geopolitical issues, given that some countries are now refusing to process the



When electronic devices reach end of life, internal components must either be recaptured for reuse or disposed of in ways that do not harm the environment.

waste they have accepted in the past from foreign nations.

The management of e-waste can be accomplished in a variety of ways, but at the heart of the process is the separation of materials. E-waste materials are physically shredded to facilitate easier sorting and separation of plastics from metals and components. Magnets are then used to separate ferrous metals from other elements, and these iron and steel fragments can then be resold as recycled steel.

Hydrometallurgical processes are used to separate metals; e-waste items such as printed circuit boards are dissolved into leaching solutions consisting of sulfuric acid, hydrochloric acid, nitric acid, aqua regia, and alkalis. The desired metals can then be recovered via a number of processes, including electrorefining, precipitation, cementation, absorption, ion exchange, and solvent extraction.

Meanwhile, pyrometallurgical processes including incineration, smelting

in a furnace, dressing, sintering, and melting at high temperatures are used to burn electronic scrap waste, resulting in plastic being separated from the other components, while the metal oxides form a slag from which non-ferrous and precious metals can be recovered. An less-common technique uses biometallurgical processing, which leverages the physical-chemical interaction that occurs when metals are exposed to microorganisms such as algae, bacteria, and yeasts; when added to a solution containing the e-waste, these reactants can accumulate heavy and precious metals. Water separation technology can be used to separate the remaining waste stream, which is usually mostly plastic and glass.

While these processes certainly are effective, they can be expensive, in terms of the actual economic cost paid to separators, as well as the ecological cost of expending additional energy to carry out each process. At the current device recycling rate of

12%, according to the U.S. Environmental Protection Agency, the economic benefit of capturing and stripping out materials as a primary recycling strategy may be limited.

That's why there is a growing interest in reducing the amount of hazardous or hard-to-recycle material at the point of design and manufacture, including hazardous chemicals, rare earth materials, or composite plastics that cannot be easily separated and recycled. Significant progress has already been made via the "lightweighting" of televisions, as manufacturers have shifted from the production of cathode ray tubes (CRTs) to flat-panel (plasma, LCD, and LED) displays, resulting in a smaller amount of plastic per television set, plus a reduction in the use of other harmful chemicals (such as lead, cadmium, barium, and a number of fluorescent powders).

However, the shift to smaller and lighter electronics is not all good news.

"Over time, the waste stream in the United States has actually gotten lighter [in terms of weight]," says Callie Babbitt, an associate professor in the Department of Sustainability at the Golisano Institute for Sustainability at Rochester Institute of Technology. "But that doesn't mean it's getting easier to manage, because the waste stream is more complex," Babbitt says.

Indeed, in many devices, such as smartphones, tablets, and even wearable devices, more tightly integrated components are harder to separate, Babbitt says, and the devices themselves are often sealed so even removing a battery can be a time-consuming challenge, making end-of-life recycling efforts both tedious and expensive. Further, as devices get smaller, the material value of electronic devices is getting smaller, requiring a greater number of devices to be recycled to capture the same amount of precious metals that can be resold.

Special attention also is being paid to reusing materials (such as plastics, metals, or composites) that can easily be separated and broken down by waste processors. Some manufacturers even design components that can be saved and recycled directly, without needing to be broken down to their core materials.

"Some companies are thinking more about the product design as a whole,

making the product more easily reused or upgraded," Babbitt says, noting that manufacturers increasingly think about substituting rare materials with more commonly sourced ones, reducing the number of different types of plastics used in composites, and eliminating compound materials such as resins or films adhered to plastics.

Further, China's enactment in January 2018 of its National Sword policy has had major implications in the e-waste world. The country, which once accepted nearly 50% of the world's recyclable waste, banned the import of most plastics and other materials into China's recycling processors, forcing a shift from simply improving recycling rates to addressing the issue of e-waste before a product is even manufactured.

Designing products with consideration of the ease of recycling is a key element of the Electronic Product Environmental Assessment Tool (EPEAT), a program of the non-profit Green Electronics Council (GEC) that provides labeling for electronic products that meet certain criteria across a range of 12 categories, covering material and chemical usage, energy efficiency, recyclability, product lifespan, and product design. The EPEAT program is voluntary, and incorporates input from manufacturers; it can be thought of as a benchmark that can be utilized in a global market where environmental and recycling laws and regulations vary widely.

Explains Patty Dillon, director of criteria development for the EPEAT program, "When we have these required and optional criteria, it's [de-

"Over time, the waste stream in the United States has actually gotten lighter, but that doesn't mean it's getting easier to manage, because the waste stream is more complex."

signed] to get the market to move in that direction where [a goal] might not be immediately achievable, but it is something that can be achieved in a one- to two-year time frame." Dillon says EPEAT often sets the levels for percentage of recycled content, energy efficiency, and product longevity, at easily achievable benchmarks to entice a wide range of manufacturers to join the program, and plans to increase these standards over time.

"For example, the initial computer standard had no requirement except that [manufacturers] declare how much recycled content was included," Dillon says. "There was no minimum level, but manufacturers could receive an optional point for 10% recycled content, while 25% recycled content got you two points. After a few years, we were able to go in and say, 'Oh, look at where the market is'; now we have a 2% required recycled content level, because we know you can do it."

EPEAT offers three levels of compliance to which manufacturers can design their products: Gold-rated products must meet all of the required criteria and at least 75% of the optional criteria; Silver-rated products must meet all of the required criteria and at least 50% of the optional criteria; and Bronze-rated products must meet all of the required criteria in their category.

Examples of required criteria vary by product category, but within the TV category, there are three required criteria: compliance with provisions of European Union (EU) Restriction of Hazardous Substances (RoHS) Directive; reporting on the amount of mercury in light sources; and compliance with the provisions of the EU Battery Directive. Meanwhile, there are 12 optional criteria, which are designed to push the industry forward in terms of reducing the use of certain substances, or eliminating specific materials altogether, such as using only non-mercury-containing light sources, or eliminating or reducing products' flame-retardant material content.

The goal behind the labeling is to make both large and consumer purchasers of electronics more cognizant of the financial and environmental benefits of "greener" electronic design, as well as encouraging manufacturers to continually improve their product de-

signs with regard to the issue of e-waste.

One of the primary components likely to become a concern over the next several years is the battery, used in many electronic devices today, but which may become even more prevalent as electric vehicles become more commonplace over the next decade.

Launched last year, the ReCell Center is a national collaboration of Argonne National Laboratory, the National Renewable Energy Lab, Oak Ridge National Laboratory, Michigan Technological University, the University of California at San Diego, Worcester Polytechnic Institute, and battery industry participants, that is funded by a three-year, \$15-million U.S. Department of Energy grant. ReCell's primary goal is to bring industry participants and researchers together to advance battery recycling technologies that can actually be used in a real-world commercial environment.

ReCell is focused on a few key areas related to battery technology, including identifying and developing materials that can be easily and cheaply recycled; developing direct cathode recycling (a process involving the retrieval of cathode materials from spent lithium batteries, and then regenerating the cathode materials by adding additional lithium into the structure of cathode material to create new battery-grade material) to eliminate the need for expensive hydrometallurgical and pyrometallurgical processes to recover specific battery metals; developing processes for recovering other battery materials; and using modeling and analysis to determine the best materials and chemistries for current and future batteries.

"The goal behind ReCell is to make battery recycling economically attractive," says Jeff Spangenberg, the organization's director. "At the national lab level, we create a lot of really cool technology, but we need industry [partners] to commercialize these technologies. We had as many industrial stakeholders as we could to come and meet with us to make sure that what we were doing made sense, and to help direct the work that we do. If nobody's ever to commercialize the technology because they know it will never work in the business world, there's no sense in us doing it."

Though ReCell is primarily focused

"Our goal is to make environmentally sound, economically attractive, recycling for lithium-ion batteries. No matter how it happens, it's a win."

on lithium batteries used in electric vehicles, Spangenberg notes that batteries used in consumer products may also benefit from ReCell's efforts, noting that many electronic devices, such as smartphones, are designed so that it's extremely difficult to even remove the battery, driving up the labor cost to recycle the battery technology. Any gains that can be made by ReCell's efforts in the EV battery space likely will show benefits in the adjacent market of consumer devices, which also rely heavily on Li-ion battery technology.

"I think that it will be, and hopefully some of these technologies will transition into other areas of recycling," Spangenberg says. "Hopefully they will gain some advantages from what we're doing. Remember, our goal is to make environmentally sound, economically attractive, recycling for lithium-ion batteries. No matter how it happens, it's a win." **■**

Further Reading

World Economic Forum, A New Circular Vision for Electronics Time for a Global Reboot, <http://bit.ly/2ToM0l9>

Cisco Annual Internet Report, VNI Forecast Tool (Device Forecasts), <http://bit.ly/2Py5RgE>

EPEAT Criteria, Green Electronics Council, <https://greenelectronicscouncil.org/epeat-criteria/>

Status of electronic waste recycling techniques: a review, *Environmental Science and Pollution Research* 25(4), May 2018, <http://bit.ly/2PzBpmm>

Keith Kirkpatrick is principal of 4K Research & Consulting, LLC, based in New York, USA.

© 2020 ACM 0001-0782/20/7 \$15.00

ACM News

ACM Members Named to AAAS

Eight members of ACM, seven of whom are ACM Fellows, were among the 276 artists, scholars, scientists, and leaders in the public, non-profit, and private sectors recently named 2020 members of the American Academy of Arts & Sciences (AAAS).

The newest AAAS members include:

- ▶ Sarita V. Adve, Richard T. Cheng Professor of Computer Science of the University of Illinois at Urbana-Champaign.
 - ▶ Thomas A. Henzinger, president of Austria's Institute of Science and Technology (IST).
 - ▶ Margaret Martonosi, a professor in the computer science department of Princeton University whose research focuses on computer architecture and mobile computing, particularly as they relate to power efficiency.
 - ▶ Fernando C.N. Pereira, a vice president and Engineering Fellow at Google, where he leads research and development in natural language understanding and machine learning.
 - ▶ Ronitt Rubinfeld, Edwin Sibley Webster Professor of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology.
 - ▶ Eugene H. Spafford, executive director emeritus of the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University, and a professor in the university's department of computer science.
 - ▶ Mihalís Yannakakis, Percy K. and Vida L.W. Hudson Professor of Computer Science at Columbia University.
 - ▶ Alexander A. Razborov, Andrew McLeish Distinguished Service Professor at the University of Chicago.
- The complete AAAS Class of 2020 is listed at <https://www.amacad.org/new-members-2020>.



DOI:10.1145/3401718

Pamela Samuelson

Legally Speaking AI Authorship?

Considering the role of humans in copyright protection of outputs produced by artificial intelligence.

SINCE THE MID-1960S, intellectual property (IP) law specialists have debated whether computers or computer programs can be “authors” whose outputs can be copyrighted.⁶ The U.S. Congress was so befuddled about this issue in the mid-1970s that it created a special Commission on New Technological Uses of Copyrighted Works (CONTU) to address this and a few other computer-related issues.⁴

A second burst of interest in AI authorship broke out in the mid-1980s. Congress once again commissioned a study, this time from its Office of Technology Assessment (OTA), to address this and other controversial computer-related issues. OTA did not offer an answer to the question, perhaps in part because at that time, it was a “toy problem” because no commercially significant outputs of AI or other software programs had yet been generated.⁵

But deep learning and other AI breakthroughs have caused IP professionals to rethink the AI authorship issue.^{1,2} For example, *The Next Rembrandt* video features a group of art experts and computer scientists discussing how they collaborated to digitize many Rembrandt paintings, develop models

of particular features of the paintings, and then create a Rembrandt-like portrait of a man with facial hair wearing a hat and looking to the right.⁶ The resulting AI-generated painting really does look like a Rembrandt. The video does not address how the team that brought this painting into being thinks about the copyright issues. But I couldn’t help myself. That painting shows the copyrightability of AI outputs is no longer a toy problem.

In the U.K. and New Zealand, that painting would be eligible for a short term of copyright protection because those nations passed laws permitting this approximately three decades ago. The question is open, however, in the U.S. and in most of the rest of the world.

AI software may be the author-in-fact of such outputs, but is it an author-in-law who can own a copyright?

In February 2020, the U.S. Copyright Office and the World Intellectual Property Law Organization (WIPO) held an all-day conference in Washington, D.C., to consider how copyright should be applied to AI outputs. The first litigated cases about copyright in AI outputs have been decided in China.

This column reviews the reasons why copyright professionals find this such a bedeviling issue. AI software may be the author-in-fact of such outputs, but is it an author-in-law who can own a copyright? Which, if any, human is entitled to claim copyrights in such outputs?

CONTU and Copyright Office on AI Authorship

The CONTU’s 1979 report concluded that there was “no reasonable basis for considering that a computer in any way contributes authorship to a work produced through its use.”⁴ It regarded computers and computer programs as tools with which works could be created much like cameras enable the creation of copyrightable photographs.

The U.S. Copyright Office has in the past rejected claims of copyright in some non-AI machine-generated works. The Office, for instance, refused to register a claim of copyright in a software-



An AI-generated painting, *The Next Rembrandt* (left) is the result of a collaborative effort using models of features from many Rembrandt paintings.

generated colored version of a black-and-white public domain movie. A machine-generated splattering of colors on a canvas, which looked something like a Jackson Pollack painting, was likewise refused registration. The Office recently reiterated its legal position on this issue: “only works created by a human can be copyrighted under United States law, which excludes photographs and artwork created by animals or by machines without human intervention.”

The need for human authorship also explains why the Office refused to register a claim of copyright in a monkey selfie. David Slater, a British nature photographer, went to a wildlife park in Celebes and set up his camera in a way that enabled a crested macaque (known to the world as Naruto) to take photos of himself smiling. Slater claimed copyright in the Naruto photos because of his creative staging of the camera and settings. When some copies of the photos appeared on Internet sites, Slater claimed this was infringement. Techdirt picked up on the dispute and questioned Slater’s ownership rights, claiming the photos were in the public domain or its posting of the monkey selfies was fair use.

An interesting twist in the monkey selfie case was a lawsuit that the People for the Ethical Treatment of Animals (PETA) brought against Slater, claiming it was Naruto’s guardian and therefore entitled to claim copyright in the photos on Naruto’s behalf. The trial judge ruled there was no human author of the photos, and so the photos were in the public domain. The appeals court affirmed dismissal of PETA’s lawsuit.

Automatic Writing Cases

One set of relatively close precedents to the AI authorship issue are those rendered in the U.S. and U.K. involving claims of copyright in texts ostensibly created by supernatural beings.¹

One such case was *Cummins v. Bond*, which a U.K. court decided in 1927. In justifying his copying of some parts of the text at issue, Bond relied on Cummins’ statements that he wrote the text in a trance and was channeling messages from the spirit world. In Bond’s view, if the spirit was the author, then no human author could claim copyright. The court decided that Cummins was the author of the text because he had “translated” the spirit’s message into English.

Penguin Books v. New Christian Church was a similar case decided in

the U.S. in 2000. The Copyright Office initially refused to register the work at issue, *A Course in Miracles*, because the application identified Jesus as its author. A second attempt at registration was more successful because “Anonymous” was now identified as its author. When the New Christian Church made copies of the text, thinking it was in the public domain, Penguin (to whom the copyright had been assigned) sued for infringement. The court held that there was sufficient creativity in the editorial selection and arrangement of these materials to support a copyright.

Software Output Cases

Two U.S. cases have ruled that certain outputs of computer programs were not infringing derivative works. The first was *Design Data v. Unigate Enterprises* in 2017. Unigate hired a Chinese company to use Design Data’s CAD software to generate drawings, data, and models for structural steel components for buildings. Unigate sold these outputs to its clients.

Design Data claimed the Chinese company used an infringing copy of its software to generate these outputs, and the outputs were thus infringing

derivative works of the infringed program. An appellate court ruled that Unigate's importation and sale of the CAD outputs were not infringements of Design Data's derivative work right. To be a derivative work, some expression from the underlying program would have to have been appropriated.

Rearden v. Walt Disney Co. in 2018 involved a similar claim. Rearden owned copyright in MOVA software, which created wire-frame models of live-action filmed performances onto which other images, such as animation, could then be superimposed for the movie. Rearden claimed that *Beauty and the Beast*, among others, infringed the MOVA copyright because the company Disney hired to generate models for this movie had used an infringing copy of the MOVA program.

Although the court allowed Rearden to proceed with its claim that Disney might be vicariously liable for its contractor's infringement, it rejected Rearden's claim that movies whose CGI effects were generated in part by an infringing program was a derivative work of the program. The court reasoned the "lion's share" of the creative expression in the movies was attributable to Disney, not the MOVA software.

Chinese Precedents

Chinese courts have recently decided two cases on AI authorship. The first was *Feilin v. Baidu*, which involved Baidu's republication of parts of "Analytic Report on the Judicial Big Data in the Film and Entertainment Industry in Beijing," which had been generated by AI software.

The court ruled that no copyright could exist in AI-generated outputs. They were not "works" protected by copyright, for there was no human author eligible to claim rights in them. The court directed that any text generated by an AI program must be identified as AI-generated.

Feilin's claim of infringement, however, was upheld because he had modified the AI outputs and had manually colored certain drawings. Under this ruling, human tinkering with AI-generated documents might qualify for copyright.

The second such case was *Shenzhen Tencent v. Yinxin*. Yinxin copied an ar-

The pragmatic answer to the AI authorship puzzle is the user who is responsible for generating the outputs.

ticle about stock market activity that was automatically generated by Dreamwriter, an intelligent writing assistance program developed by Shenzen. The court ruled that the article was copyrightable and Shenzen was its author. Yinxin's copying of the article was held to be an infringement, a ruling that is seemingly inconsistent with *Feilin*.

Commentator Views

A few dozen articles have been written over the years, speculating about the copyrightability of computer-generated works and the AI authorship issue.^{1-3,6} No consensus has emerged from this commentary.

Some say AI-generated works are in the public domain, like the monkey selfie. Some say the person or firm that wrote the AI program should get copyright in any copyrightable outputs. Others suggest the person who actually generates the output should be the rights-holder, if anyone is.

Some propose that both the programmer and the user should be co-owners of any copyrights in AI-generated outputs. Some would adapt the U.S. work-made-for-hire rules, under which employers or entities that specially commission certain works are authors-in-law, even if not authors-in-fact, to enable copyright ownership rights to be decided.

One problem with these proposed solutions is the AI outputs having some commercial value are products of highly collaborative processes, as *The Next Rembrandt* video demonstrates. AI software is not, as some commentators seem to believe, a black-box into which data is input at one end and the output spit out at the end. AI software has numerous component parts, not all of which may come from the same entity:

training data, weights to be given to various criteria, models for generating outputs or certain parts, algorithms used to analyze the data, and software that executes instructions. Also important is the know-how of AI programmers who fine-tune these component elements to yield the desired results.

Conclusion

The pragmatic answer to the AI authorship puzzle, as I have argued elsewhere,⁴ is the user who is responsible for generating the outputs. If anyone needs to be designated as owner of rights in the outputs, it should be the user. That person possesses the outputs, discovered that the potential commercial value of the outputs, and is generally best situated to assess and exploit that value.

Moreover, as in the automated writing and *Feilin* cases, the user will often have adapted, rearranged, edited, or otherwise tinkered with the outputs to make them suitable for commercialization. If anyone needs copyright incentives to take the raw outputs and adapt them for commercial dissemination, it is that user. Besides, the user will also have already paid the owner of the AI software components for the right to use them to generate outputs.

It is, moreover, unlikely the Copyright Office or judges in litigation will generally be able to tell the difference between outputs that have been created by AI and those created by humans. Only time will tell what definitive answer that legislators and courts decide upon to resolve this long-standing puzzle. ■

References

1. Bridy, A. Coding creativity: Copyright and the artificially intelligent author. *Stanford Tech. Law Journal* 5, 1 (2012).
2. Ginsburg, J.C. and Budiarto, L.A. Authors and machines. *Berkeley Technology Law Journal*, 34, 343 (2019)
3. Grimmelmann, J. There's no such thing as a computer-authored work—And it's a good thing, too. *Columbia J. Law & Arts* 39, 403 (2016).
4. National Commission on New Technological Uses of Copyrighted Works, Final Report (1979).
5. Office of Technology Assessment. *Intellectual Property Rights in an Age of Electronics and Information* (1986).
6. Samuelson, P. Allocating ownership rights in computer-generated works. *U. Pittsburgh Law Review* 47, 1185 (1986).
7. The next Rembrandt: Can the great master be brought back to create one more painting?; <https://www.nextrembrandt.com/>

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

Copyright held by author.

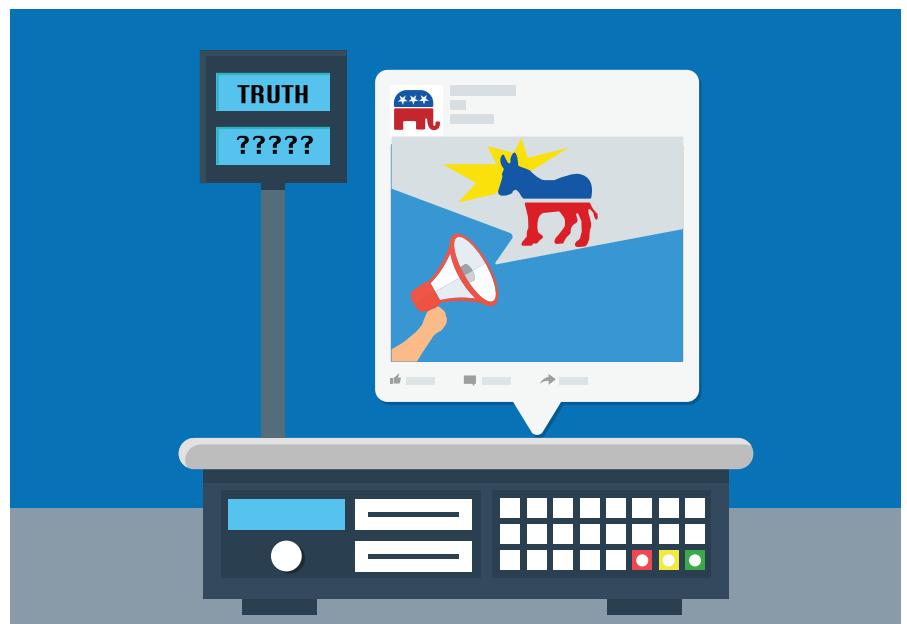
Economic and Business Dimensions Proposal: A Market for Truth to Address False Ads on Social Media

Guaranteeing truth in advertising.

WHEN IT COMES to political ads on Facebook, anything goes. On Twitter, nothing does. In a speech at Georgetown University last October, Mark Zuckerberg, Facebook’s chief executive, defended the company’s decision not to fact-check political ads on the site. Shortly after, Jack Dorsey, Twitter’s chief executive, tweeted that his company had decided to reject all political advertising. In January of this year, Facebook doubled down on its original decision to accept all political ads no matter how egregious the lies an ad buyer wishes us to believe.

They are both wrong. Facebook pollutes our political discourse. Twitter impoverishes it. Between promoting false ads and rejecting truthful ones, here’s a better way: create a “market for truth.” It requires neither machine algorithms to discern truth nor judgments by a potentially self-interested company. Instead, it discourages liars from lying.

First, ask political advertisers to guarantee their truth. Each politician or PAC that places an ad would put a large sum of money in escrow as an “honest ad pledge” that their claims are true. Second, if anyone disputes the ad, an independent fact-checker would judge the ad’s truthfulness. This role could



fall to any one of a number of organizations that routinely make such judgments: FactCheck.org, Politifact, HoaxSlayer, or Snopes. It could even be a panel sampled randomly from Fox and CNN viewers. The watchword is independence. It cannot be Facebook’s self-appointed Oversight Board and it most emphatically cannot be government.

To dispute an ad, an aggrieved party must issue a challenge by paying a non-refundable fee. This challenge price should cover the cost of independent

fact checking. Unchallenged ads wouldn’t require fact checks. If an ad is challenged, the independent fact checker then judges the disputed ad’s accuracy.

If the ad proves false, the injured party receives the advertiser’s pledge, which they can spend as they wish to undo the damage the false ad has caused. If the ad proves true, however, the pledge reverts to the ad buyer, and the challenger forfeits the cost of the fact check. Challengers would have no incentive to pay for fact checking that

Results of ACM's 2020 General Election

President:

Gabriele Kotsis
(July 1, 2020–June 30, 2022)

Vice President:

Joan Feigenbaum
(July 1, 2020–June 30, 2022)

Secretary/Treasurer:

Elisa Bertino
(July 1, 2020–June 30, 2022)

Members at Large:

Nancy M. Amato
(July 1, 2020–June 30, 2024)

Tom Crick
(July 1, 2020–June 30, 2024)

Susan Dumais
(July 1, 2020–June 30, 2024)

Mehran Sahami
(July 1, 2020–June 30, 2024)

Alejandro Saucedo
(July 1, 2020–June 30, 2024)

proves their opponents are right. If an ad goes unchallenged after, say, 30 days, the honest ad pledge reverts to the ad buyer.

In all cases, the cost of guaranteeing the truth of an honest ad is zero. The false advertiser, however, has paid for the ad, paid the pledge penalty, and paid in reputation. Simply put, the forfeited pledge is the price of a lie. It is paid only by liars. A politician who still wishes to lie may certainly do so. But lying becomes expensive.

What about the slippery middle ground between truth and falsehood—the innuendo and half-truths that infect so much political advertising? Imagine a photo of Joe Biden and his son looking shifty, accompanied by the tagline: “Hunter Biden served on the board of Ukraine’s most corrupt company while his father, as Vice President, did all he could to fire a powerful Ukrainian prosecutor.” None of that is exactly false. But it implies the senior Biden tried to prevent the prosecutor from going after the company, when in fact he sought the opposite: he wanted the prosecutor fired for failing to pursue corruption.

How should an honest ads market handle an ad like this? It refunds half the pledge for an ad that’s half a truth. Based on the egregiousness of the lie, the amount of a refund can correspond to one of the sliding scales fact checkers already use. Indeed, Politifact did rate an ad like the one here as half-true on a scale that ranges from: true, mostly-true, half-true, mostly-false, false, and pants-on-fire. Other fact checkers use similar scales. A market for truth need not be perfect. It just needs to be credible and unbiased. By asking PACs and politicians to warrant their claims, it changes the balance of power, favoring truth over lies in our political discourse.

**A market for truth
need not be perfect.
It just needs
to be credible
and unbiased.**

A politician of greater integrity could make bigger promises and voluntarily escrow 2X or 3X the normal pledge. Or, a politician of little means could pre-check a message with the fact checker, ensure an honest message and have bonds markets underwrite the pledge without risk. By contrast, a politician of low integrity could afford only smaller promises, namely the minimum lie price that the media platform requires as its honest ad pledge. And, low integrity politicians would keep losing their pledges. For a dishonest politician, the costs mount with each additional lie.

How might this work? To operate a market for truth, we can rely on established administrative practices that we already use for trust and legitimacy. Taking our own government as precedent, we split oversight into legislative, judicial, and executive branches. A legislative body gets to define “fake ads.” Despite their differences, even Fox News, CNN, and the *New York Times* might be able to agree on a working definition of fake ads independent of specific use cases and their own ads and news stories. A judicial body gets to decide whether a specific case represents an instance of fake advertising according to this definition. Again, Snopes, Hoax Slayer, Politifact, or a jury of peers might play this role only now they must judge according to the definition provided by the legislative body. Jurors do not get to use their own individual definitions. Finally, the executive branch enforces these definitions and decisions. It collects the honest ad pledges and disburses them to ad buyers or ad challengers based on rulings by the judicial body. Social media platforms like Facebook and Twitter can play this role but they decide neither the definitions nor the outcomes of challenges. By dividing the branches of fake ads governance, we recreate an institution where no branch judges truth as applied to itself and no branch has an economic incentive to bias its behavior to get rich.

Why does this work? A truth market for trading honest ads works for exactly the same reason as a carbon market based on cap and trade. It solves the problem of pricing externalities and markets for trade in externalities already exist. Carbon dioxide is pollution. It is a negative externality that harms others. An entity that is causing damage needs

to pay for that damage by buying pollution credits that put a price on the harm done. Fake news is pollution. It is a negative externality that harms others. The size of the honest ads pledge, that is, the lie price, could be any escrow amount set by the social media platform but really should be the expected size of the harm done. This negative externality is the “social cost” of the damage done by lying. The crowdsource identification of harm is the market that “trades” the externality. The harmed parties claim the lie price and get paid for the damage they experience. Carbon trading markets work so we can expect markets for truth will also work.

Importantly, a market for truth works even when the amount of damage, the lie price, is not known in advance. Imagine Exxon Mobile today taking out an ad that human activity does not cause global warming. The lie price for political ads in the U.S. alone is too small for the lie price of global warming policy ads internationally. You can quickly see that, if a firm repeatedly pays the lie price, then their willingness to keep lying is too small relative to the true social cost. Then the lie price should rise until they stop the lies that harm people. In other words, we have an “efficient search” process that can force firms and super PACs to internalize the true social cost of their negative externalities even when that cost is initially unknown.

And what about free speech? In the U.S., skeptics might object that an honest ads pledge would not withstand First Amendment scrutiny if the pledge were mandatory. U.S. courts view impediments to speech as violations of free speech. Although this is a uniquely U.S. problem, the system still works even when a pledge is voluntary. If the market for truth is fully functioning, then unwillingness to pledge an honest ad is itself a signal that the author is likely lying because honest ads incur no added cost. The 2001 Nobel Prize in Economics acknowledged the tenets of information economics precisely due to the power of “signals” to separate truth from lies. Informative signals are potentially expensive actions taken by knowledgeable parties that back up their claims. A product sold with a guarantee, for example, is almost always more reliable than a product sold “as is” or “buyer

Good sellers, knowing their claims are true, can offer guarantees that bad sellers, knowing their claims are false, cannot afford to offer.

beware.” Good sellers, knowing their claims are true, can offer guarantees that bad sellers, knowing their claims are false, cannot afford to offer. The voluntary signal separates good from bad, and fact from fiction. The proposed mechanism is very powerful.

An honest ad pledge discourages political advertisers from placing false ads. The pledge need not be mandatory—advertisers’ failure to pledge signals they do not believe their own claims. A fair challenge price discourages political adversaries from launching false challenges. The mechanism also provides revenue to pay for independent fact checking via issued challenges. Fact checkers have no financial incentive to bias their decisions and not every ad would need checking—only those that are challenged. Governance can proceed using models we already use in other contexts.

As with so many other aspects of social media, platforms like Facebook and Twitter have made the spread of false content what the tech world proudly calls “frictionless.” It is time to judiciously put some friction back. A truth market would do just that. A society that values unfettered expression over truth can set the price of lying low. A society that values greater integrity can set the price of lying higher. Currently, the price of lying in political ads is zero. □

Marshall Van Alstyne (mva@bu.edu) is a Questrom Chair Professor at Boston University where he teaches information economics. He is also a Digital Fellow at the MIT Initiative on the Digital Economy and co-author of the international best-seller *Platform Revolution* (W.W. Norton).

Copyright held by author.

Calendar of Events

At press time, scheduled conferences were significantly impacted by COVID-19, often requiring cancellation or postponement. The following conferences are scheduled to be held virtually. Please check conference websites for updates on programs and event dates.

July 6–10
DIS '20: Designing Interactive Systems Conference 2020,
Sponsored: ACM/SIG,
Contact: Kristina Anderson,
Email: h.k.g.andersen@tue.nl

July 6–10
WebSci '20: 12th ACM Conference on Web Science,
Sponsored: ACM/SIG,
Contact: Dame Wendy Hall,
Email: wh@ecs.soton.ac.uk

July 8–12
GECCO '20: Genetic and Evolutionary Computation Conference,
Sponsored: ACM/SIG,
Contact: Carlos A. Coello,
Email: ccoello@cs.cinvestav.mx

July 8–12
LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science,
Contact: Lijun Zhang,
Email: zhanglj@ios.ac.cn

July 13–17
ACM EC '20: ACM Conference on Economics and Computation,
Budapest, Hungary,
Sponsored: ACM/SIG,
Contact: Peter Biro,
Email: peter.biro@krtk.mta.hu

July 13–17
DEBS '20: The 14th ACM International Conference on Distributed and Event-based System,
Sponsored: ACM/SIG,
Contact: Kaiwen Zhang,
Email: kaiwen.zhang@etsmtl.ca

July 13–17
PEARC '20: Practice and Experience in Advanced Research Computing,
Sponsored: ACM/SIG,
Contact: Gwen Jacobs,
Email: gwenj@hawaii.edu

► Susan J. Winter, Column Editor

Computing Ethics For Impactful Community Engagement: Check Your Role

*Toward a more equitable distribution
of the benefits of technological change.*

CHECKS ARE NEEDED to guide the development of guardrails for ethical and responsible community-engaged computing research. The era of “move fast and break things” can produce false starts, injured communities, and widespread techlash. The tech sector can be more socially conscious and focus on community engagement using research from universities, computing researchers, and professionals. For example, smart cities might increase efficiency and improve quality of life, but for whom?¹⁰ Research shows how smart city initiatives can harm certain groups through, for example, facial recognition technologies that misidentify, produce ethnic bias and discrimination, or create opportunities for abuse.⁵ Technology benefits do not always accrue evenly across community members.

Ethics rarely keeps pace with technological innovation. Computing research in community-engaged projects all too often lacks tools for planning, engaging in, reflecting on, and evaluating whether the projects may bring unintended negative impacts. Computing professionals and researchers must navigate this complex ethical landscape as they bring value to communities. The key lesson, learned repeatedly but often for-



gotten, is from the parable of the tortoise and the hare. Proceeding with care produces lasting results. We explore the value of continuous checking for community-engaged research that provides guidance, so that innovation makes a positive contribution to societal well-being.

Common Problems in Community-Engaged Research

Civic and community-engaged research is often lauded, but conducting research in a community does not mean that a researcher follows ethical

practices aligned with local social norms, is responsive to local communities’ needs, or has adequately considered the potential unintended consequences. Harm is often caused even when no willful violation of ethics exists. For this reason, a deeper awareness of the responsibilities attendant to doing community-engaged research is urgently needed.

Universities and corporations have professed a focus on ethical practices of inclusion and advancement of research with public value, but often struggle to realize them in practice.

Some common mistakes that community-engaged researchers make are:

- ▶ Conflating any community participation at any stage of the research as sufficient
- ▶ Not letting a group self-define as a “community”
- ▶ Not fully understanding the problem or its context
- ▶ Not grasping the existing community power dynamics
- ▶ Not planning mindful strategies for entrance into and exit out of communities
- ▶ Reinforcing existing biases and power imbalances
- ▶ Not being aware of the historical relationship between a community and academic institutions
- ▶ Placing essential stakeholders in roles of nonparticipation or tokenism

Whether initiated by municipal officials, private corporations, or university researchers, individual and community stakeholders are often largely missing from all stages of the design and decision-making processes in community-engaged research projects, particularly in marginalized communities.¹⁰

To provide societal value through community-based efforts, we must develop ethical approaches, best practices, and institutional safeguards for community-engaged research. There are no specific remedies that can be applied to avoid each of these common mistakes, but our recommendations reduce the likelihood of making any of them. Though far from comprehensive, the recommendations discussed next highlight an existing research lens, a methodology, and four best practices for inclusive community-engaged research.

Existing Approaches to Inclusive Community Engagement

Existing approaches can help computing professionals develop best practices for civic and community-engaged work. Researchers can adopt *postcolonial computing* as a way of thinking about community-engaged computing research and development.^{6,7} Postcolonial computing was inspired by the ethical challenges of transferring technological knowledge between disparate cultures and poses a new vision for computing research and practice. Whereas colonialism extends and dominates one

The partner readiness quiz acts as a type of *humility audit*, encouraging reflection and empathy before engagement.

entity’s authority, values, or ideas onto another, postcolonial computing encourages us to reflect on the ways in which we may become a colonizing force, pushing a technology or practice onto a local community. Such interactions are fraught with potential for asymmetries of power between researchers and communities and may lead to a destructive logic of “us” knowing better and helping “them.” A postcolonial computing lens requires computer science professionals to abandon the notion that their projects are about making things better for “other” cultures or communities. Rather, the focus becomes understanding the complex dynamics between researchers and community members, including local traditions and perceptions of power. Memorial University of Newfoundland’s CLEAR Lab^a is an excellent example of a postcolonial approach to community-engaged research.

Community Based Participatory Research (CBPR) offers concrete principles and procedures aligned with the postcolonial computing lens. In CBPR, the goal of research is to exercise “power with” individuals in a community, rather than exert “power over.” CBPR envisions communities as entities defined from within, who already possess power and resources⁹ while simultaneously attending to social inequities and structural repression of power for certain individuals and communities.⁸ The focus is on communities engaging in research to address

problems identified directly by the community and using the community’s knowledge and resources to produce solutions within existing practices and local knowledge. CBPR recognizes that all members of a community have the right to be involved in decisions that affect their lives.² It produces direct research outcomes and community capacity by involving people who can take actions to improve their own conditions³ while balancing research and action for the benefit of all.⁸ As part of CBPR, researchers, designers, and communities undertake ongoing conversation and reflection about which methods to use, what kinds of knowledge and pathways to action are being created, and who is served. Adopting postcolonial computing and CBPR approaches to community-engaged research reduces the likelihood of making the eight common mistakes described here; four concrete techniques that put these approaches into practice and can guide researcher activities.

An Updated Toolbox for Ethical Community-Engaged Research

The postcolonial computing lens and the CBPR methodology enable computing researchers and professionals to support community efforts while engaging in research with communities that provide real societal value. The authors’ collective research experiences inform four additional key practices that are useful for computing research and design with communities.

First, project leads should conduct “readiness for partnership” checks to determine if computing professionals are ready to collaborate as equals with community members and vice versa. One of our co-authors (Lewis) had a wealth of experience on both sides of community engagement, both with researchers who sought to work with her as an influential community member, as well as being the PI of a grant herself. Her experiences managing the tensions between these roles as an expert patient, activist, and inventor led to the development of parallel patient and partner readiness quizzes.^b The partner readiness quiz acts as a type of *humility audit*,

a See <https://www.theatlantic.com/video/index/591640/recycling-plastics/>

b See <https://partner.openingpathways.org/> and <https://patient.openingpathways.org/>

encouraging reflection and empathy before engagement. More tools like these should be developed, widely disseminated, and used to assess academics' readiness to collaborate with non-academic communities.

Second, we must work to *remove distinctions* of status between community participants, academics, and practitioners while maintaining awareness of power dynamics between participants and their contexts. Through creating an atmosphere of mutual respect and establishing 'psychological safety,' groups can continually reduce distinctions in status and promote co-production of knowledge that draws on diverse forms of expertise.⁴ In framing a problem or designing solution(s), many individuals and groups may hold different viewpoints of the situation. At the beginning of a project, it is helpful to create a comprehensive understanding of all stakeholders who are directly or indirectly affected and to strive to include them. As solutions are proposed, it is important to revisit stakeholder perspectives and include additionally affected individuals, groups, and communities.

Third, *value expertise* by creating paid community researcher positions, or engage in exchanges of goods and/or services for labor. Avoid community volunteer work. Providing fair and equitable compensation to community collaborators acknowledges their expertise and legitimizes their contributions. Providing compensation often requires that we navigate and even agitate within our organizations and to external funding bodies to include compensation for community members as well as the time of researchers and designers to manage relationships with these community positions. However, this is important, and initiating this conversation will help shift our institutions to make this easier to do in the future.


Fourth, Institutional Review Boards (IRB) are designed to protect human participants in research but can be slow to adapt, and committee members may not have expertise in community-engaged research. Universities should establish separate *Social Embeddedness IRB review committees* just like they have different IRB review committees for biomedical and social/behavioral research. Socially

Responsible design requires mutual care, respect, and equality.

embedded IRB committees would review research intended to create direct impact on communities through the research process. They would articulate a set of rights that communities could exercise to ensure the research furthers the goals and honors the values of the community, that participants in community-engaged research are treated ethically, and that their rights, interests, and welfare are protected. A Social Embeddedness IRB would support the interests of both the researcher and communities, share best practices between projects, and make visible the distribution of benefits and accountability among various stakeholders.

Conclusion

As Avle, Li, and Lindtner proposed in their essay on "Responsible IoT after Techno-solutionism," changing the way we work when conducting computing research and design across cultures requires that we move toward mutual accountability.¹ Responsible design requires mutual care, respect, and equality. Rather than ill intent, the eight common mistakes of community-engaged research we noted often stem from a lack of awareness that can be remedied by having a framework (postcolonial), an approach (CBPR), and an ethos to continue to do better as demonstrated by the four key practices outlined here. Through collectively reflecting on our experiences in community-engaged research and entrenched cultural assumptions and biases, we can undo incomplete models and practices that pervade computing research and design. We propose drawing on the wealth of guideposts that exist within and outside of computing to create an ever-expanding toolbox for community-engaged research that leads with an ethos of humility and care, and includes practices of self-disclosure and

reflection to improve our ability to have a positive impact with communities. This may be the best way to ensure the benefits and harms of technological change are distributed equitably and forestall any coming "techlash." 

References

1. Avle, S., Li, D., and Lindtner, S. Responsible IoT after techno-solutionism. *Medium*. (Aug. 27, 2018); <https://medium.com/the-state-of-responsible-iot-2018/responsible-iot-after-techno-solutionism-cf583e5f9b9a>
2. Barber, B.R. *Strong Democracy: Participatory Politics for a New Age*. University of California Press, 1984.
3. Baum, F., MacDougall, C., and Smith, D. Participatory action research. *Journal of Epidemiology and Community Health* 60, 10 (Oct. 2006), 854–857; <https://doi.org/10.1136/jech.2004.028662>
4. Edmondson, A. Psychological safety and learning behavior in work teams. *Administrative Science Quarterly* 44, 2 (Feb. 1999), 350–383; <https://doi.org/10.2307/2666999>
5. Garvie, C., Bedoya, A., and Frankle, J. *The perpetual line-up: Unregulated police face recognition in America*. Center on Privacy & Technology at Georgetown Law (2016); <https://www.perpetualineup.org/>
6. Harrington, C., Erete, S., and Piper, A.M. Deconstructing community-based collaborative design: Towards more equitable participatory design engagements. In *Proceedings of the ACM on Human-Computer Interaction* 3 (CSCW), (2019) 216:1–216:25; <https://doi.org/10.1145/3359318>
7. Irani, L. et al. Postcolonial computing: A lens on design and development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (2010), 1311–1320; <https://doi.org/10.1145/1753326.1753522>
8. Israel, B.A. et al. Critical issues in developing and following CBPR principles. In *Community-based Participatory Research for Health: Advancing Social and Health Equity* (3rd ed). Jossey-Bass & Pfeiffer Imprints, Wiley, 2017, 31–46.
9. McKnight, J.L. and Kretzmann, J.P. Mapping community capacity. In *Community Organizing and Community Building for Health and Welfare* (3rd ed), Rutgers University Press, 2012, 171–186.
10. Winter, S.J. Who benefits? Considering the case of smart cities. *Commun. ACM* 62, 7 (July 2019), 23–25; <https://doi.org/10.1145/3332807>

Kathleen H. Pine (khpine@asu.edu) is an assistant professor at the College of Health Solutions, Arizona State University, Phoenix, AZ, USA.

Margaret M. Hinrichs (mhinrich@asu.edu) is a postdoctoral research associate at the School for the Future of Innovation in Society, Arizona State University, Tempe, AZ, USA.

Jieshu Wang (jwang490@asu.edu) is a Ph.D. student in Human and Social Dimensions of Science and Technology at the School for the Future of Innovation in Society, Arizona State University, Tempe, AZ, USA.

Dana Lewis (Dana@OpenAPS.org) is the founder of OpenAPS, Seattle, WA, USA.

Erik Johnston (Erik.Johnston@asu.edu) is a professor at the School for the Future of Innovation in Society, Arizona State University, Tempe, AZ, USA.

The ASU Knowledge Exchange for Resilience is supported by Virginia G. Piper Charitable Trust. Piper Trust supports organizations that enrich health, well-being, and opportunity for the people of Maricopa County, AZ, USA. The conclusions, views and opinions expressed in this column are those of the authors and do not necessarily reflect the official policy or position of the Virginia G. Piper Charitable Trust. This material is based upon work by the National Science Foundation under Grant No. 1816080. Any opinions, findings, and conclusions or recommendations expressed in the material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Copyright held by authors.

Viewpoint

Consumers vs. Citizens in Democracy's Public Sphere

Attempting to balance the challenging trade-offs between individual rights and our obligations to one another.

FROM FOREIGN INTERVENTION in free elections to the rise of the American surveillance state, the Internet has transformed the relationship between the public and private sectors, especially democracy's public sphere. The global pandemic only further highlights the extent to which technological innovation is changing how we live, work, and play. What has too often gone unacknowledged is that the same revolution has produced a series of conflicts between our desires as consumers and our duties as citizens. Left unaddressed, the consequence is a moral vacuum that has become a threat to liberal democracy and human values.

Surveillance in the Internet Age, whether by governments or companies, often relies on algorithmic searches of big data. Statistical machine learning algorithms are group-based. Liberal democracy, in contrast, is individual-based, in that it is individuals whose rights are the chief focus of constitutional protection. Algorithmic opacity, which can be the product of trade secrets, expert specialization or probabilistic design,³ poses additional challenges for self-government because it by definition abstracts away the individual on which a rights-based regime depends. Even with attentiveness to constitutional constraints, NSA surveillance, as Edward Snowden revealed, violated the right to privacy that all American



citizens have, leading to revision of the Patriot Act. Europe's privacy protection standards are even higher, restricting second- and third-hand use of customer data.

In contrast, in illiberal regimes, the distinction between the public and private spheres is not drawn in the same way, and the individual is not the regime's point of departure. Privacy is routinely sacrificed at the altar of national security and societal goals. For example, China's Google, Baidu, has partnered with the military in the China Brain Project. It involves running deep-learning algorithms over the data Baidu collects about its users. According to an account in *Sci-*

entific American, every Chinese citizen will receive a so-called 'Citizen Score,' which will be used to determine who gets scarce resources such as jobs, loans, or travel visas.^a China also uses facial recognition software to monitor its Uighur Moslem minority for law enforcement purposes.^b China's Social Credit System, designed to reward "pro-social" and punish "anti-social" behavior, is becoming operational.^c

Since capitalism and democracy

a See <https://bit.ly/3er2Fxx>

b See <https://nyti.ms/3dcE11U>

c See <https://wapo.st/2zHrXIG>; and <https://bit.ly/3esPZpC>



developed contemporaneously and symbiotically, the divergence between technological advances and human values has been all too easy to overlook. In the Chinese context, deploying citizen scores and racial profiling in such utilitarian fashion may be legitimate, but in a rights-based democracy, such algorithmic discrimination must be illegitimate. Bell curves do not matter for human rights. Individuals do.

In *The Human Condition*, Hannah Arendt lamented “the absurd idea of establishing morals as an exact science” by focusing on things that are easily measurable or quantifiable.¹ From her perspective, what was most significant about modern theories of behaviorism is “not that they are wrong but that they could become true ... It is quite conceivable that the modern age—which began with such an unprecedented and promising outburst of human activity—may end in the deadliest, most sterile passivity history has ever known.”¹ The question is not “whether we are the masters or the slaves of our machines, but whether machines still serve the world and its things, or if, on the contrary, they and the automatic motion of their processes have begun to rule and even destroy world and things.”¹

The widening digital age conflict between producers/consumers and citizens reflects a heightened tension between market and liberal democratic/republican values. To see this more clearly, it is helpful to think about two models of man: bourgeois man and citizen man. For Marx and

his heirs, the class struggle between bourgeois man and working man, between the oppressor and the oppressed, is the central dynamic. Montesquieu, Machiavelli, Montaigne, and the American Founders, however, “ransacked the archives of antiquity,” as Arendt puts it, to imagine a different model of man for the new republic. The model man of this new system, which built on the Roman conception of the public sphere, was the citizen of the Athenian polis.² The American Constitution’s architects thus drew on both Greece and Rome in imagining the new republic of the United States. Inclusive citizen engagement in American political life is thus essential for both self-government and human flourishing. While it is a fact that the original vision excluded blacks and women from citizenship, it is equally true the same values evolved over time to include all humans of voting age.

It is easy to see how contemporary free market fundamentalism—the idea that free markets are the solution to all challenges of public life—is a logical consequence of trends Arendt astutely identified over a half-century ago. Whenever the people allow companies to pursue profit maximization relentlessly in a global market without attention to consequences, producers are unwittingly elevated over citizens. Along parallel lines, whenever citizens allow their personal data to be harvested in exchange for a better deal, consumers are inadvertently elevated over citizens. Progressive Supreme Court Jus-

tice Louis Brandeis foresaw this accelerated challenge to the health of democracy’s public sphere, when he described the Gilded Age consumer as “servile, self-indulgent, indolent, ignorant” and thereby easily manipulated by advertising, the opposite of the engaged citizen.⁴

The March 2016 standoff between Apple and the FBI illustrates the new potential for conflict between business and government interests that technological change has wrought. Apple refused to help the government unlock the iPhone of Syed Farook, who was charged with killing 14 in the December 2, 2015 San Bernardino terrorist attack. Since Apple’s market is global, it had no interest in complying with the FBI’s request, as its foreign customers are unlikely to pay a premium for a smart phone that the U.S. government can access. Yet Apple is also a company headquartered in the U.S., and American citizens have an obvious interest in preventing future terrorist attacks. The same friction between the profit motive and public interest was present in the decisions of Facebook’s senior leadership to downplay Russian interference in the 2016 elections until a free press forced them to own their self-interested choices.

The Internet thus has had at least two major consequences for American constitutional democracy. First, as our public conversations move online, disparate virtual spaces are replacing the public square, undermining democratic deliberation. As we saw with the 2016 elections, Facebook first looked the other way when its platform was manipulated by the Russians and others to increase polarization and help elect Donald Trump. To get a better sense of the magnitude of the problem, Facebook announced in May 2019 that it had deleted more than three billion fake accounts, a number approximately comparable to the combined 2018 populations of the U.S., China, and India.^d

The global nature of the ad market for Google and Facebook represents the greatest challenge. Facebook profited when Russian troll farms bought ads in the run-up to the 2016

d See <https://bbc.in/2zts01h>

election, but the American public sphere was simultaneously diminished. Looking to the future, the prospect of an alliance between authoritarian states and large IT monopolies that would effectively merge corporate and state surveillance, as George Soros has warned, could facilitate totalitarian control unlike anything the world has previously seen.^e

The move to cloud computing has also had important implications for privacy rights. The Fourth Amendment requires the government to justify to a court why it has a compelling interest in your personal information, protecting the contents of your laptop and desk from illegal search and seizure. What most Americans do not understand is that once you upload material to the Cloud, you trade that Constitutional protection for a corporate guarantee, yet the Fourth Amendment is mute on corporate violations of privacy.

When they sell themselves to the public as promoters of ideals rather than as profit-seeking companies, Silicon Valley firms have a vested interest in obscuring this simple fact. Until very recently, Google's mantra was "Don't Be Evil," and Facebook still defines its mission as "to make the world more open and connected." Apple recently rebranded its retail outlets as "town squares." This sincere Newspeak made it easier for consumers unthinkingly to trade their personal data for continued free use of the relevant platform.

Engineers and senior corporate leadership alike must be mindful of this decoupling of profit margins from the common good. All Western computer scientists should care about the consumers vs. citizens tension, not only because as citizens they value liberal democracy, but also because the long-term sustainability of the companies for which they work depends on it. When platforms or products appear to undermine human values, brands are tarnished in the free world, sometimes irreparably. The Google/Apple collaborative effort using Bluetooth to support contact tracing apps appears to have learned this lesson.

e See <https://bit.ly/2XBO7UR>

To be fully human in a liberal democracy is to be a citizen first and a consumer second.

The challenge will be to reclaim the public sphere for the people and democratic deliberation rather than as a locus for self-promotion and manipulation. A necessary condition for meeting that challenge will be to reintroduce practical ethics to scientific knowledge. When technological innovation outstrips the capacity of existing norms and laws, it will take more than science to re-harness science to the public interest.

Yet while Arendt and Brandeis discerned the general trajectory, in some ways, we are in uncharted territory. Scientists at the dawn of the nuclear age made possible weapons of mass destruction that still today could wreak total destruction. Scientists in the internet age are developing intelligent machines to do what was previously the work of humans. In the information age, scientists are seemingly on the brink of rendering large segments of society utterly superfluous.

One thing is certain: Silicon Valley will not be capable of safeguarding human values without public pressure and thoughtful regulation. The conflict of interest is too stark, since the core dilemma often embodies a clash between higher short-term profit margins and doing the right thing for equality before the law. There is strong sentiment, a lingering effect of decades of considering government as the problem rather than a solution, that tech companies can simply engineer processes that used to be the preserve of courts (such as Facebook's recent move to create its own oversight board to judge what content or accounts are approved or removed

from the platform^f), but it is important to remember what government is for and that there are some things that only government can do well. Simply put, nobody elected or appointed Silicon Valley.

In navigating these challenges, we can start with things we know to be true. Since both algorithmic design and data categorization can be amplifiers of prejudice, the perfect algorithm will be no silver bullet for protecting individual rights. An algorithm cannot fathom the human experience. An algorithm cannot understand the requirements of the democratic system itself.

Put another way, to be fully human in a liberal democracy is to be a citizen first and consumer second. Politics has no place in scientific research. At the same time, scientists are also citizens who are ideally positioned to evaluate both the perils of AI systems and their potential to better the human condition. Scientists who understand the inherent trade-offs between what we want as consumers or producers and what we need as citizens can be critical allies rather than enemies of pluralism and the freedom of the individual.⁵

To sustain democracy in the Coronavirus era, Americans need to be citizens first. That is a choice we all must make. Human beings in a free society cannot be reduced to data or algorithms unless we allow ourselves to be. ■

f See <https://wapo.st/2ZRNDwm>; and <https://n.pr/3dbUVPE>

References

1. Arendt, H. *The Human Condition*. University of Chicago Press, Chicago, IL, 1958, 311.
2. Arendt, H. *Thinking Without a Bannister*. Schocken Books, New York, 2018, 467–468.
3. Dourish, P. Algorithms and their others: Algorithmic culture in context. *Big Data & Society* 33, 2 (July–Dec. 2016), 6–7.
4. Rosen, J. Louis D. *Brandeis: American Prophet*. Yale University Press, New Haven, CT, 2016, 76.
5. Vienna Manifesto on Digital Humanism; <https://www.informatik.tuwien.ac.at/dighum/manifesto/>.

Allison Stanger (stanger@middlebury.edu) is the Russell Leng '60 Professor of International Politics and Economics at Middlebury College, Cary and Ann Maguire Chair in Ethics and American History at the Library of Congress, Center for Advanced Study in the Behavioral Sciences Fellow at Stanford University, and an External Professor at the Santa Fe Institute.

Viewpoint

Call For A Wake Standard for Artificial Intelligence

Suggesting a Voice Name System (VNS) to talk to any object in the world.

APPLE PIONEERED THE voice revolution in 2011 with the introduction of Siri in its iPhone 4s. Today, you tell your iPhone 11, “Hey Siri, Play Bruce Springsteen by Spotify,” and it responds, “I can’t talk to Spotify, but you can use Apple music instead,” politely displaying options on the screen^a as shown in the figure here. Or, you tell one of your five Amazon Echo devices at home, “Alexa, add pumpkin pie to my Target shopping list,”^b then “order AA Duracell batteries,” and it adds pumpkin pie and Amazon Basics batteries to your Amazon shopping cart, ignoring your request to shop at Target and be loyal to Duracell. You are the consumer, but your choices have been ignored.

Or, consider you are a brand manager. You want to customize the voice of Echo to match your brand persona, but it is not an option offered. Amazon only lets users change the default female voice to “male” and to a few other lonely options. Instead you decide to create a personalized assistant. However, unlike the leading technology companies (the so-called FAANGs—Facebook, Apple, Amazon, Netflix, and



Google), your company does not have the thousands of engineers and vast advertising budgets needed to develop it.

These examples suggest artificial intelligence (AI) environments are evolving toward more limited choice. However, we believe an open and standardized approach would be beneficial to most, including the FAANGs themselves. In fact, this approach may be urgent, given the rapid growth of adoption and increasing level of complexity for users and skill suppliers.

Large companies are rushing to the rapidly growing voice opportunity,

introducing incoherent smart speakers spanning many markets and languages. Each has its own “wake construct,” which we define as the utterance that activates a skill. Typically, it consists of a “Wake Word” followed by some skill name and associated parameters. It may also include input from face recognition or other sensors, such as what happens when you activate Siri on your Apple watch by waving your wrist. Incompatibility among devices from different suppliers is pervasive and may come from any of the wake construct’s components.

a Apple changed this response right after the EU filed an investigation into the matter. *The Financial Times*, May 5, 2019

b We developed a skill in our lab and in July 2017 Alexa’s parsing surprisingly changed. The phrase “Alexa, Target shopping list add soap” went from adding it to the skill’s list to directly adding it into Amazon’s shopping list.

For example, if you say “Alexa, Bedtime” to your Echo, it starts a skill operated by Johnson and Johnson, while if you tell Google Home, “OK, Google, Bedtime,” the device initiates a different routine that is owned by Google itself. Sometimes the behavior is similar in both devices but the service operator is different—for example, saying the words “sleep sounds” to your Echo or Google Home will initiate two different skills related to sleep and relaxation sounds. There is no central and standard repository of words to avoid this sort of inconsistency, such as one where a shop owner could reserve an action name across devices and languages.

For skill developers, devices are also incompatible and may have language-specific skill programming options. This means that porting P&G’s Tide Alexa Skill to all combinations of devices and languages would require maintaining hundreds and eventually thousands of different versions. Even then, the front-facing experience will not be consistent across devices and languages, which could confuse customers. Creating a somewhat consistent user experience across this myriad of options would require the sort of budget that is unjustifiable for small businesses. Even P&G’s Tide skill, introduced several years ago, is still available only in English.

Wake Neutrality

Amazon and Apple’s business aggressiveness—their choice to rout product requests to their services, while developing incompatible closed-garden solutions—makes perfect sense for large companies trying to establish dominance in this competitive space. However, any advantages from this aggressive incompatible offerings may be offset by raised entry costs for skill suppliers such as P&G, antitrust violations affecting retailers large and small,⁵ or unattended consumer preferences.^c

Incompatibility in the voice space contrasts with the situation in Wake Neutrality markets. In these markets, such as the Web, the phone network, barcodes, or even WiFi networking, there are standard ways of activating

services.⁹ This gives consumers a consistent experience across more choices and at lower entry costs. On the Web, one can type “www.kohls.com,” and trust reaching the same retailer, no matter the browser type, WiFi network, or device OS. Toscanini’s Ice Cream phone number is the same, no matter the phone maker, calling app, or network provider. The Internet has the Domain Name System (DNS),^d which ensures a unique name in the DNS, and phone networks have the North American Numbering Plan (NANP), which ensures a unique number in NANP.

Similarly, we believe the voice space, and AI in general, would greatly benefit from a standard “Voice Name System” (VNS) enabling unique skill names across devices. We suggest the VNS incorporates three architectural components: “Common Wake Constructs” (CWC), “Secure Voice Channels” (SVC), and “Conversational Privacy Firewalls” (CPF); each are now reviewed in turn.

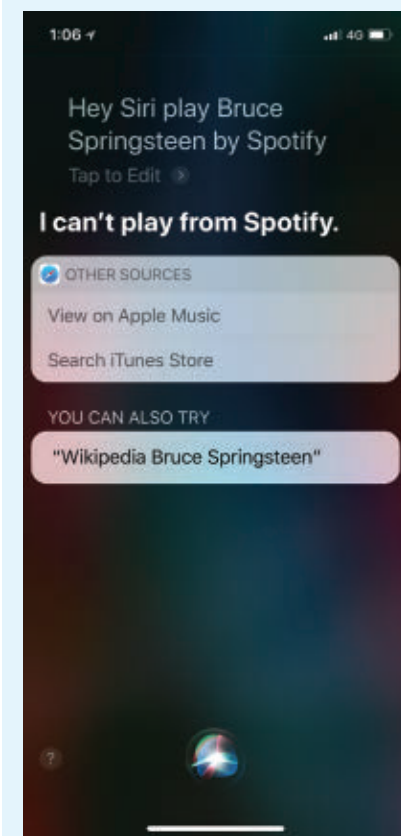
Common Wake Constructs (CWC)

Our first architectural suggestion is to implement Common Wake Constructs to standardize voice request routing. For example, the analogue of typing “*http://www.lidl.com/shopping-list*” could be to say “*OK Google, Lidl Open Shopping list.*”^e The DNS may be a starting point, so one cannot reserve a word on the VNS without having the corresponding DNS word, but it needs one extra step because voice is ambiguous. The identical pronunciation of store brands “Coles” and “Kohls” requires disambiguation. This could be done geographically, by adding a prompt, such as “OK, Coles” versus “Coles”, or by a wrist movement like in the Siri example here. Deciding whether two names are similar may be tricky requiring arbitration. Domain name registrars have performed such arbitration functions within the DNS. For example, they prevent COLES.org—with a zero “0”

^d <https://tools.ietf.org/html/rfc882>

^e “OK Google” would be like the first part of the url, that is, *http://*, “Lidl” like “*www.lidl.com*” and “Open Shopping list” like that of “*shopping-list.*” Just like there is “*http*”, “*ftp*”, one could imagine different voice services. These could include a “local skill mode” and a “single-shot” transfer mode that sends a command, but does not transfer subsequent speech commands to “Lidl.” It could also include a “multiple-shot” transfer mode that keeps you in the skill for multiple interactions.

The iPhone indicated various options to listen to Bruce Springsteen, but not the one requested.



instead of the letter “O”— from being registered to prevent phishing.

Disambiguating voice is more complex than disambiguating text and it may lead to errors because computer speech recognition is ambiguous and not 100% accurate. The VNS could address error correction by using other sensors, requiring a user to spell out a word, or verification on a different device. To decide whether two speech utterances resemble each other too closely, phoneme matches could determine whether the probability of a phishing attack is above a threshold, before a voice domain is granted. To automatically establish this probability using deep learning algorithms one can use one of the public speech sample repositories using an open source speech-to-text solution such as Baidu’s, which calculates probabilities in its last layer.

A CWC as proposed here is feasible using today’s technology. So is a standard that also includes as CWC basic command phrases such as “*<raise volume>*” or “*<play> Bruce Springsteen <in> Spotify.*” Together with one of my

^c For example, some consumers reject the courtesy behavior exhibited by dominant devices: <https://bit.ly/36yYEEi>

Digital Creativity Support for Original Journalism

Why Computing Belongs in the Social Sciences

Understanding Computer Science Participation in North Carolina

Threats of a Replication Crisis in Empirical Computer Science

Case Study: The Record-and-Replay Approach to Debugging

Q&A with Elisa Bertino

Magellan: Building Management Systems

The Revolution Will Be Digitized

Thorny Problems in Data-Intensive Science

Plus the latest news about neuromorphic chips, digital effects in movies, and technology addiction.

students we have created Huey, a CWC programming language based on human natural language.^{11,13}

Secure Voice Channels (SVC)

Devices currently avoid security issues by only allowing simple use cases, such as ordering a cab or pizza, in private environments like the car, home or office, because voice can be very insecure. In fact, researchers have shown existing wake algorithms can be fooled by sound sequences inaudible to the human ear, even to the point of forcing transcription to potentially any desired phrase.²

To expand security options, we propose designing Secure Voice Channels by adding a security layer to smart speakers, analogous to “secure http” (https), so that selected CWC require a more secure process to be activated. In public spaces, it could be based on responding to a “trick question” or in pressing a specific “secure” button associated to the device, much as we do today when using a “car key.” The VNS could also blacklist spaces where harmful voice intrusions are known to occur.

Conversational Privacy Firewalls (CPF)

A Burger King TV ad woke nearby Google devices when it stated, “OK, Google, What’s a Whopper?”. The Cannes Lions International Festival of Creativity singled out this Burger King ad as the most intrusive advertisement ever because it triggered a follow-on ad skill in each home and it informed Google which homes were watching. Hackers changed the skill to be harmful, and Google had to pull it out immediately.^f The potential for privacy violations is unprecedented because short audio segments recorded by smart speakers can have Private Identifiable Information (PII) including genre, race, mood, alcohol intake, and even personality disorders.⁶

Automated analysis of speech has been shown to detect onset of psychosis, in young adults, even before human experts. In my lab, we recently used AI to diagnose Alzheimer’s with only 20 seconds of speech and identified longitudinal biomarkers to track disease progression, achieving a spontaneous-

^f With key-activated SVC none of this would have occurred.

Disambiguating voice is more complex than disambiguating text and it may lead to errors because computer speech recognition is ambiguous and not 100% accurate.

speech detection rate of 93.8%, the highest reported so far.⁷ Similarly, we used forced cough recordings to identify COVID-19 subjects¹² and demonstrated that a single cough can reveal cultural and biological information.⁴

Our third suggestion, Conversational Privacy Firewalls (CPF), is an architectural block that filters input-output voice signals limiting the amount of PII available to intervening players. Depending on what type of PII one wants to protect, a different CPF filter mode may be appropriate. Here are some examples:

- ▶ “Speech Incognito Mode”: This converts speech to text, so that skill suppliers don’t receive any information contained in the soundwave other than text.

- ▶ “Vision Incognito Mode”: For devices such as the Echo Look, the filter could transform images to prevent proper face recognition, while blocking PII about gender or race.

- ▶ “Alexander Mode”: This mode takes speech and converts it into commands using a synthesized robot voice, spacing requests when possible. This ensures that neither the voice, the location, nor the sequence of commands is shared.

- ▶ “Strong Incognito Mode”: Evidence shows our choice of words conveys a lot of PII.³ This mode would convert “I desperately need two tickets for Sunday’s Baseball Match and would pay any amount”, to neutral text, ensuring all a service provider receives is a request in “neutral English” with limited location and user sentiment information, such as, “Are there any tickets available for Sunday’s match?” This type of incognito mode could result eventually in a new form of “Esperanto” for AI devices.

Call to Action Toward A Voice Name System (VNS)

Two suggestions for short-term actions to begin establishing the VNS standard include:⁵

1. **Defining Roles:** Implement the first version of the VNS, which would include CWC, SVC, and CPF, by using existing open source software and hosting it with an existing non-profit organization. Non-open source development choices are also possible and may co-exist. For example, large players could continue to grow closed-garden solutions, while sharing key ingredients of CWC, SVC, and CPF. The competitive landscape, including the role of each constituent, must be clarified, and existing standard bodies, such as W3C, IEEE or GS1, may need to get involved.

2. **Setting Community Objectives:** For a first version of the VNS to be used widely, agreement is needed on which application area is first. This will help set technical choices such as file formats, routing protocols and command syntax. We suggest working toward a first version of the VNS that allows you to talk to any Web page, phone app, or smart speaker. Subsequent work would pursue a new version that allows you to converse with any object in the world so that you can ask the tomato sauce you are holding in your hand “Am I allergic to you?”. This would require combining Natural Language Processing with other AI modalities such as high-level computer vision, gaze detection, SSVEP brain sensing or gesture recognition, and could imply interfacing with the Internet of Things (IoT).⁶ In addition to the VNS, we may need an Artificial Intelligence Name System (AINS).

^g There are a number of initiatives already under way but none yet with the scope we suggest. Amazon, for example, enabled Echo to wake Microsoft Cortana (*The Washington Post*, August 16th, 2018) and later announced a voice effort with a few more partners (The Verge, September 16th, 2019). A few retailers are behind the Open Voice Network (www.openvoicenet.org) a Linux initiative to standardize voice based on the MIT research of my lab, which has long-term objectives similar to the ones here suggested. W3C has a few groups interested on the topic too. Apple introduced last September “voice over”, a way for users to control via speech any application in IOS screens. Inspired in the MIT Center for Brain Man and Machine’s four module model of the human brain, MIT is introducing reference architectures for the VNS through the MIT Auto-ID Lab Open Voice initiative.

Eventually, more difficult choices will have to be made, such as determining how to manage the capturing, storing and sharing of sensor samples to improve AI device communication abilities using legal programming.⁸ For instance, when should devices be allowed to listen and talk “intelligently” to each other? When should we selectively process video footage from the home and from public spaces? Can the intelligence gathered then be used to customize AI personalities that induce you to consume more? If what we say stored at scale is gold, then, who owns our voice samples? And what about safety? Should AI agents, for example, be allowed to prevent you from driving if they hear you sound intoxicated, and should they warn you if they detect an increased risk of depression? May devices disclose your whereabouts if there is an active search warrant for you? An open discussion of these questions could enlarge the VNS standardization effort for the benefit of all. □

References

1. Bedi et al. Automated analysis of free speech predicts psychosis onset in high-risk youths. *Nature npj Schizophrenia*, 1, (2015). 15030.
2. Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C. & Zhou, W. (2016, August). Hidden Voice Commands. In *USENIX Security Symposium* (pp. 513-530).
3. Eichstaedt J.C. et al. Facebook language predicts depression in medical records. In *Proceedings of the Natl Acad Sci USA*, 115, 44 (2018); 11203–11208. doi:10.1073/pnas.1802331115
4. Huetto, F. et al. *Discovery RNNs, explainable RNN saliency visualization, and its application to unsupervised segmentation of COVID-19 forced coughs*. To appear 2020.
5. Khan, L.M. Amazon’s antitrust paradox. *Yale Law Journal* 126 (2016), 710.
6. Kraepelin, E. Manic depressive insanity and paranoia. *The Journal of Nervous and Mental Disease* 53, 4 (1921), 350.
7. Laguarda, J. et al. *Explainable Audio Proc. with the MIT CBMM Open Voice Brain Model: Multi-modal personalized diagnosis of Alzheimer using COVID-19 Coughs*. To appear 2020.
8. Sarma, S., Brock, D.L., and Ashton, K. The networked physical world. Auto-ID Center White Paper MIT-AUTOID-WH-001 (2000).
9. Subirana B., and Bain, M. Legal programming: Privacy and security in highly dynamic systems. *Commun. ACM* 49, 9 (Sept. 2006), 57–62; DOI: 10.1145/1151030.1151056
10. Subirana, B., Bivings, R., and Sarma, S. Wake neutrality of AI devices. Chapter 9, *Algorithms & Law*, M. Ebers and S. Navas, Eds., Cambridge University Press, 2020 ISBN 9781108347846
11. Subirana, B. and Levinson, H. *Conversational Computer Programming: Creating Software with Huey Conversational Commands*. 2020.
12. Subirana, B. et al. Hi Sigma, do I have the Coronavirus?: Call for a New Artificial Intelligence Approach to Support Health Care Professionals Dealing With The COVID-19 Pandemic. arXiv preprint arXiv:2004.06510 (2020).
13. Subirana et al. The MIT Voice Name System (VNS) and the Huey Natural Language. White Paper, MIT Auto-ID Laboratory. arXiv preprint (2020).

Brian Subirana (subirana@mit.edu) is Director of the MIT Auto-ID Laboratory and member of the Faculty of Harvard University, Cambridge, MA, USA.

Copyright held by author.



Advertise with ACM!

Reach the innovators
and thought leaders
working at the
cutting edge
of computing
and information
technology through
ACM’s magazines,
websites
and newsletters.



Request a media kit
with specifications
and pricing:

Ilia Rodriguez
+1 212-626-0686
acmm mediasales@acm.org



Q Article development led by [acmqueue](https://queue.acm.org)
queue.acm.org

Building projects despite (and because of) existing complex systems.

BY PAT HELLAND

The Best Place to Build a Subway

MANY ENGINEERING PROJECTS are big and complex. They require integrating into the existing environment to tie into stuff that precedes the new, big, complex thing. It is common to bemoan the challenges of dealing with the preexisting stuff. Many times, engineers don't realize their projects (and their paychecks) exist only *because of* the preexisting and complex systems that impose constraints on the new work.

This column looks at some sophisticated urban redevelopment projects that are very much part of daily life in San Francisco and compares them with the challenges inherent in building software.

The best place to build a subway is in the open cornfields of Nebraska.

It's pretty darned flat. You can space the subway stations at regular intervals in a consistent fashion. There are no pesky historic monuments that get in the way. No cathedrals and no civic center.

Unfortunately, there are no passengers in the corn fields and no economic reason to have a subway there. This leads to a chicken-and-egg problem. You cannot easily build infrastructure where it's crowded, and you cannot afford to build infrastructure where it's not crowded.

I'm fond of saying there are two kinds of software companies: those that start out building scalable infrastructure and those that are in business.

This means successful companies will invest in evolving their infrastructure for enhanced scalability. That's difficult.

Infrastructure and Users Growing Together

Urban planning is fascinating. There are questions of financing the changes to the city and the churn of land use and infrastructure. In general, a decision is made that the old occupants must leave, and they are bought out at what is supposed to be fair market value. Significant amenities such as parks, transit centers, and more are planned for the area. With the now-vacant land and promised fancy amenities, developers can purchase parcels to develop skyscrapers for offices, hotels, and housing. The price of the property under the skyscrapers pays for the original land purchase and the amenities.

I recently spent five years living in Golden Gateway, right by the Ferry Building in the Financial District in San Francisco. The area was redeveloped in the 1970s. It involved tearing down an old, dilapidated produce market, which was moved about five miles south. Five skyscrapers of office building, a retail mall, hundreds of apartments, multiple parks, and a Hyatt Regency now occupy the space.

My job is smack in the middle of



The Transbay Tube during its construction in the 1960s.

the Transbay redevelopment, also in the Financial District. The one- to two-story rundown light manufacturing in this area south of Market Street was condemned, and the land was transferred to city ownership. Transbay has been an ongoing project since around 2000 and seems to be more than half done. As I go in and out of work, I see brand-new towers, an amazing transit center and park, as well as cranes and construction.

Other redevelopments include the

Yerba Buena district (now including the Moscone Convention Center) and the Mission Bay area (now including the new basketball arena, Chase Center).

Many of these massive projects have incurred various costs and challenges. In the 1950s and 1960s, the Fillmore redevelopment targeted a part of town whose residents were largely African Americans. A 36-block area was torn down including housing, a distinct lifestyle, and a world-famous jazz community. Most of the

previous occupants could not afford to return.

In many cases, these huge, multi-decade redevelopment projects bring new life to part of a city, but sometimes we can't foresee what we're going to lose.

Jasper O'Farrell's Risky Plan

In 1847, San Francisco was a tiny town of about 600 people, formerly known as Yerba Buena, that just the previous year had joined up with the United

States. Gold was not discovered in California until the next year, which would transform San Francisco into a major city.

A few years earlier, in 1835, William Richardson had settled in Yerba Buena, and laid out the streets for an expanded settlement on a north-south grid. This area, called Portsmouth Square, is now part of Chinatown. By 1847, numerous buildings were aligned on this north-south grid.

In the southern part of the town was Mission Dolores. Mission Street ran 4 1/2 miles from the mission to San Francisco Bay in a northeasterly direction. I work in a building on Mission Street.

In 1847, the new American military mayor of San Francisco commissioned Jasper O'Farrell to perform a land survey of San Francisco. He corrected a number of the property boundaries and street alignments in the northern

part of the city. He decided to cut the city into two grids: the northern grid running north-south and east-west, and the southern grid running north-east-southwest and southeast-northwest. The existing northern part of the city had a few hundred wooden structures. The street widths varied from 45 feet to 69 feet, some a bit larger. In the southern part of the city, where there was little or no settlement, O'Farrell decided to make the streets wider. Mission Street was laid out to be 82 1/2 feet wide.

To separate these two parts of the grid, O'Farrell created a massive 120-foot-wide Market Street paralleling Mission Street in the southern part. This was a ridiculously large waste of land for a street, especially in a town of 600 people. Remember, the transportation at the time consisted of horses and wagons. What would be the reason for such a wide street?

The locals of the town were furious. That land was valuable and was being taken from them for no value at all. Quickly, a mob formed and decided Jasper O'Farrell should hang for wasting their land. The townsfolk set out to get him. Fortunately, a friend tipped O'Farrell off and he rode a horse to North Beach, caught a boat to the North Bay, and settled in Sonoma County, where he died in 1875.¹

Serendipity Where You Least Expect It

In the 1950s, plans were laid for the Bay Area Rapid Transit (BART). This commuter train would run at high speed, connecting San Francisco, through tubes under San Francisco Bay, to the East Bay including Oakland, Berkeley, and more. To everyone's surprise, the vote for increased taxes squeaked by in 1962, and BART was funded.

The Transbay Tube connected to San Francisco at Market Street, and plans included a tunnel under Market Street for about two miles. The Market Street Subway is shared with the light-rail Muni Metro lines that run trains throughout San Francisco. Muni runs only within the city of San Francisco.

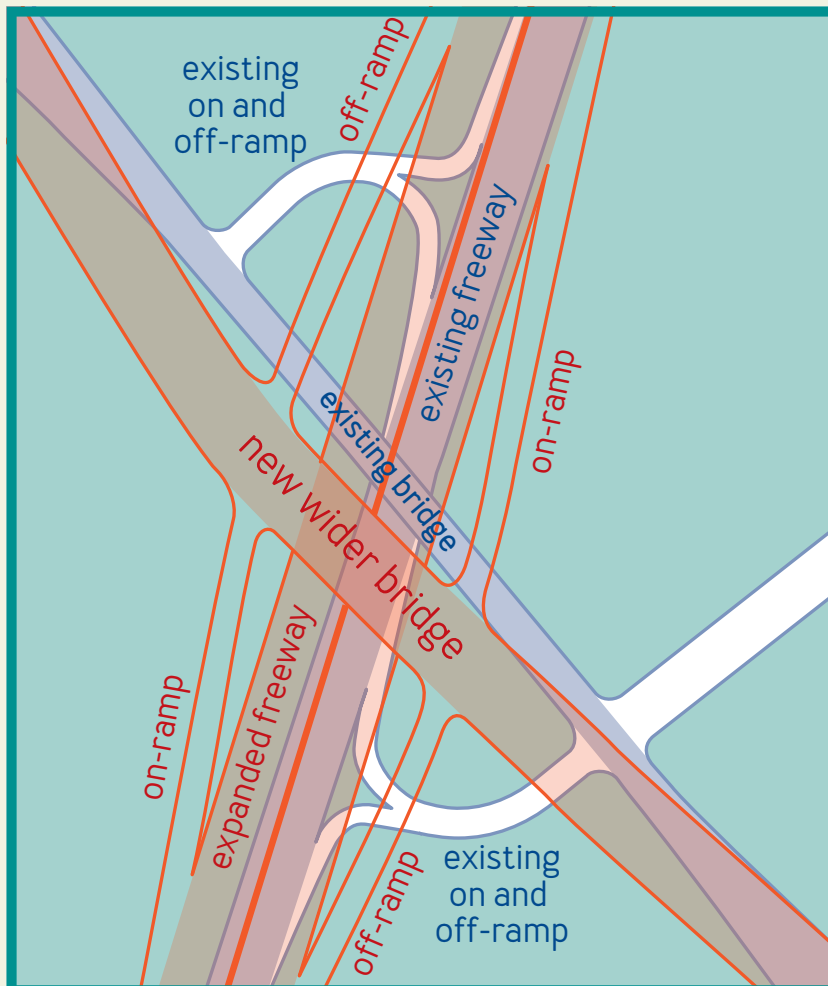
Construction on Market Street started in 1967 and continued for almost 10 years. Using the cut-and-cover approach dramatically reduced the construction cost. Cut-and-cover involves digging a huge trench and building the stations and the tracks in the open air. When done, the underground project is covered with dirt. It's unlikely that the Market Street Subway could have been funded with a less intrusive approach. Unfortunately, for 10 years Market Street was a mess, and lots of businesses went bankrupt.

Still, the 120-foot width of Market Street opened up the possibility that construction with cut-and-cover could be attempted. Without that generous width, we likely wouldn't have BART today.

Why Does That Overpass Have a Bend in It?

When I have the occasion to sit in a window seat on a plane, during take-off and landing I always look at the overpasses on the freeways. Frequently, I see a gentle bend in the road as it

An overpass schematic courtesy of CALTRANS.



crosses the freeway. This causes me to smile as I recognize the remnant of a freeway's widening.

Many times, I've lived near or commuted through a freeway-widening construction project. Each time, the process seems nonsensical. Buildings on one side of the crossover road are torn down. Then, new foundations and a bridge are built side by side with the old bridge. In general, there's no interruption of the traffic on the overpass and, even more important, no interruption on the freeway. Sometimes, with excruciating pain, a freeway will have to be shut down over a long holiday weekend.


When the construction is done, the beautiful new overpass will have been constructed right next to the old one. The new overpass connects to the side street just where the old one did. It just swerves to the side as it crosses the freeway to allow for two overpasses before tearing down the old one.

One especially challenging part of evolving a complex system is keeping it going while it's being changed. Years back, any new version of software had to be sent in a box, and after installing, it would run better on the data stored on disk. By the early 1980s, I was worried about wide area network distributed transactions and how I could compatibly evolve the protocol. It was not unusual for this to take three releases of planned messaging changes, with each release being six months apart.


Now, everyone supports cloud-based solutions. Everything runs 24/7. That is a huge value for customers and puts additional constraints on the engineers supporting the system and the application. Just like the folks widening the freeway need to keep it running 24/7, we need to plan for the evolution of the system and the detailed steps required to get from here to there.

Conclusion

I've spent almost 35 years of my 40-plus-year career working at companies with thousands of engineers. Having so many engineers means it is both easier and harder to get projects completed. With lots of resources, you have the ability to assemble a substantial team. While you have the benefit



Urban planning is fascinating. There are questions of financing the changes to the city and the churn of land use and infrastructure.



of lots of resources, however, there's a lot of interdependency and engineering nuance to consider. Even more, there's the legacy of a large code base. Legacy typically offers much more good than bad.

Cities are usually designed around the prevailing transportation. I have been privileged to visit Old Jerusalem, which was built with donkeys in mind for transportation. Most streets in the old part of the city are perhaps 20–25 feet wide. Cars cannot drive on these streets. Of course, widening them would be impossible without destroying the buildings next to them.

Applications start with communication and data or database expectations, as well as application structure expectations. Just like cities and transportation evolve, the compute infrastructure evolves.

Starting with a clean slate may seem to be more desirable. There are fewer constraints. There's also an increased chance that your software project will not take root and will not matter to anyone. The best hope is to build something that has an appropriate investment in infrastructure based on the economics. While doing that, try to have the insight to leave especially wide roads, perhaps 120 feet wide, even if they don't matter much now. Just make sure the townsfolk don't become a mob looking for vengeance! 

Related articles on queue.acm.org

Eventual Consistency Today: Limitations, Extensions, and Beyond

Peter Bailis and Ali Ghodsi

<https://queue.acm.org/detail.cfm?id=2462076>

Condos and Clouds

Pat Helland

<https://queue.acm.org/detail.cfm?id=2398392>

Sizing your System

Kode Vicious

<https://queue.acm.org/detail.cfm?id=1413256>

Reference

1. Prendergast, T. *Forgotten Pioneers*. Books for Libraries Press, San Francisco, CA, 1942, 71.

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

Copyright held by owner/author.
Publication rights licensed to ACM.

Article development led by [acmqueue](https://queue.acm.org)
queue.acm.org

Cryptography meets monetary policy.

BY JEREMY CLARK, DIDEM DEMIRAG,
AND SEYEDEHMAHSA MOOSAVI

Demystifying Stablecoins

THE FIRST WAVE of cryptocurrencies, starting in the 1980s, attempted to digitize government-issued currency (or *fiat currency*, as cryptocurrency enthusiasts say).⁸ The second wave, represented prominently by Bitcoin,⁷ provide their own separate currency—issued and operated independently of any existing currencies, governments, or financial institutions. Bitcoin’s currency (BTC) is issued in fixed quantities according to a hard-coded schedule in the protocol.

In the words of Bitcoin’s pseudonymous inventor:

“There is nobody to act as a central bank... to adjust the money supply... that would have required a trusted party to determine the value because I don’t know a way for software to know the real world value of things. If there was some clever way, or if we wanted to trust someone to actively manage the money supply to peg it to something, the rules could have been programmed for that. In this sense, it’s more typical of a precious metal. Instead of the supply changing to keep the value the same, the supply is predetermined and the value changes.”²



Without active management, the exchange rate of BTC with governmental currencies has been marked by extreme volatility. Figure 1 shows a comparison of fiat currencies and bitcoin. The values were retrieved daily between Jan. 1, 2016 and Jan. 1, 2019. (Note that 1,000 mBTC = 1 BTC). Squint at the chart to notice how the GBP (British pound) drops around June 2016: This mild-looking pinch is actually the so-called “sharp decline” and “severe swing” that followed the Brexit referendum in the U.K. It is completely overshadowed, however, when placed beside BTC’s large fluctuations.

A Third Wave?

Extreme volatility is not specific to BTC. It can also be seen in its contemporaries: ETH (ether) and XRP (Ripple). This instability is an issue of practical



IMAGE COMPOSITION BY ANDREJ BORYS ASSOCIATES, USING PHOTO BY RUDMER ZWIERVER

importance: Volatility encourages users to hoard (if the value is going up) or avoid (if it is going down) the currency rather than use it. It makes lending risky, as currency movements can exceed interest payments. A lack of lending and credit inhibits the formation of mature financial markets. In response, a flood of proposals have been made for new cryptocurrency designs that purport to provide a stable exchange rate similar to (or exactly mirroring) a government-issued currency like the U.S. dollar. These designs are called *stablecoins*.

Stablecoins have garnered a lot of attention recently, both positive and negative. According to *CoinMarketCap*, a service that provides financial metrics for cryptocurrencies, more value in tether (a cryptocurrency issued by Tether Limited) changes hands across a giv-

en day than bitcoin—despite questions about tether’s reserves and regulatory investigations into its affiliates. The announcement of Facebook’s Libra stablecoin project made international headlines and has been remarked on by the Federal Reserve Board, U.S. legislators, and even the sitting U.S. president. Another project, Basis (*née* Basecoin) raised \$133 million in venture capital but folded when it could not find a tenable path through U.S. financial regulations. Central banks, including those of Sweden and Denmark, have explored the idea of government-issued stable cryptocurrencies.

Stablecoins promise the functionality of Bitcoin without the roller-coaster ride of its exchange rate. But can this new breed of cryptocurrency really outsmart decades of central bank policy with algorithms and smart contracts?

Knowledge Gap

Understanding how stablecoins work should be easy. Most projects have white papers outlining the design, the coins are marketed to the general public, and there is no shortage of online articles surveying various designs.

Unfortunately, there are a number of pitfalls in systemizing this knowledge. Many white papers are obfuscated with jargon—terms left undefined and used inconsistently across other projects and the financial literature. In other cases, system components appear to be mislabeled. For example, a component that clearly meets the definition of a *security* or a *derivative* might instead be labeled a *bond* or a *loan*. Maybe this is a lack of precision. Maybe it is a play to make an unconventional protocol appear more conventional. Or maybe these are un-

Figure 1. Comparison among fiat currencies and Bitcoin.

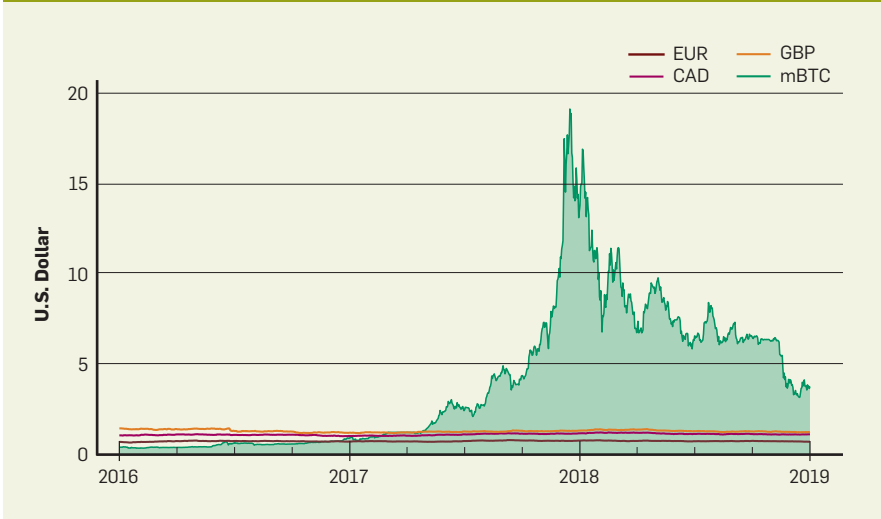


Figure 2. Stablecoin proposals as of Jan. 11, 2019.

Class	Mechanism	Resembles	Rank
Backed	Directly Backed and Redeemable	USDC	20
		TrueUSD	26
		Paxos	38
		Gemini Dollar	52
		StableUSD (USDS)	685
		Stronghold USD	891
		Petro	1210
	Directly Backed	Tether	6
		EURSToken	95
		BitCNY	304
		Terracoin	1280
		Saga	1495
		GJY, Novatti AUD, UPUUSD	⊥
Indirectly Backed	Dai	57	
	BitUSD	398	
	Nomin	⊥	
Intervention	Money Supply Adjustments	Ampleforth	⊥
		RSCoin	⊥
	Asset Transfer	NuBits	892
		CarbonUSD	1262
		Basecoin	⊥

conscious attempts at keeping any regulatory red flags at half-mast. In any case, here we make an effort to offer direct and simple explanations. In parallel to our work, other academics have produced their own taxonomies.^{6,9}

How Do Stablecoins Work?

We started by finding stablecoin projects on CoinDesk, an online news source

for cryptocurrencies, using search queries such as “stablecoins,” “stability,” and “price-stable.” This resulted in 185 articles up to Jan. 11, 2019. (Given its high profile, Facebook’s Libra coin, which was released after this date, is included.) The 25 projects for which there was sufficient documentation were classified as shown in Figure 2. Projects are classified according to

what they assert (for example, there is no warranty that projects classified as “redeemable” provide actual redemption of the assets that back their coins). Projects are sorted according to their rank on *CoinMarketCap*, which evaluates cryptocurrencies that are actively traded on an exchange service. Unlisted projects are ranked ⊥.

Next, each project was distilled into a core stability mechanism. Instead of enumerating the intricate details of how each “brand” of stablecoin works—details that could change tomorrow—we concentrated on the fundamentals. Broadly, the projects can be split into two categories: those that try to directly match the stability of a second asset such as the U.S. dollar and could not exist without this underlying asset; and those that propose independent currencies with algorithmic and/or human intervention mechanisms for providing stability.

Type 1: Backed Stablecoins

The first general type of stablecoin tries to match the stability of a second *target* asset, such as the U.S. dollar, either by making use of it (*directly backed*) or by making use of a third *reserve* asset like ETH (*indirectly backed*). These stablecoins could not exist without their underlying assets.

Directly backed and redeemable. For stablecoins in this category, the company operating the cryptocurrency obtains a reserve of some valuable asset—it might be the U.S. dollar or another sovereign currency, gold or another commodity, or a basket of multiple assets. It then issues digital tokens that represent a unit of the underlying asset, which can be exchanged online (to illustrate, assume a token is redeemable for \$1).

Working Example: Alice is a trusted third party and uses Ethereum to instantiate a DApp (decentralized application), which issues 1,000 AliceCoins as standard tokens (for example, ERC20). She asks \$1 USD for one AliceCoin and promises to redeem any AliceCoin for \$1 USD. If Bob buys 10 AliceCoins for \$10 USD, Alice deposits the \$10 USD in a bank account. Whenever Alice receives a buy order for AliceCoins and does not have any left to sell, she creates new ones. If Carol wants to redeem five AliceCoins, Alice

withdraws \$5 USD and exchanges it with Carol, taking those AliceCoins out of circulation. Alice frequently publishes bank statements showing that her account holds enough U.S. dollars to redeem all coins in circulation (the number of AliceCoins can be checked any time on Ethereum. For more information, see the sidebar “Ethereum and DApp Primer.”).

The idea of directly backed and redeemable currency predates Bitcoin: Liberty Reserve provided a similar digital currency, with some caveats about its redeemability (not to mention its legality). Liberty Reserve, e-gold, and similar pre-blockchain services, however, would maintain transaction details and account balances on a private server. Blockchain enables decentralized trust for the transactions, while the coin creation and redemption processes rely on a trustworthy firm (see sidebar “Bitcoin and Blockchain Primer.”). In short, this type of stablecoin is more centralized than Bitcoin but less than Liberty Reserve. Also consider that while decreasing centralization can be good for trust and transparency, additional measures are needed to ensure it is not harmful for privacy.

For finer-grained analysis, Figure 3 provides a comparative evaluation where a filled circle (●) indicates the properties (columns) are fulfilled by the corresponding mechanism (rows) within reason. A half circle (◐) means the property is fulfilled but the fulfillment is bounded. An open circle (○) means it is unfulfilled. A question mark (?) indicates a heuristic has been proposed for stability and the conditions under which it will work are not well enough established to evaluate. Finally, (×) indicates

the property is not applicable.

Recall the mechanism for issuing AliceCoins. If buyers are willing to pay more than \$1 USD for 1 AliceCoin, new coins can be generated for \$1 USD and sold to these buyers for a profit, ensuring bids return to \$1 USD (it corrects overvaluation). If sellers are willing to take less than \$1 USD for 1 AliceCoin, those coins can be bought and redeemed for a profit, ensuring offers return to \$1 USD (it corrects undervaluation).

In reality, transactions are not free, efficient, or entirely frictionless and some price deviation is expected. If redemption is ever in doubt, then the price can fall freely from \$1 USD (although this will not necessarily happen; as we will discuss). The trustworthiness of the operating firm and the custodian of the reserves is essential, and financial audits are an important step to establishing confidence (although many pitfalls exist when audit-

Bitcoin and Blockchain Primer

A public blockchain is a type of distributed database (or ledger) that is open to anyone who wants to maintain it, is robust against faulty and malicious participants, and runs without anyone in charge. When participants look at a local copy of the ledger, they are assured that everyone has the exact same records and that each record was validated by the majority of participants before it was written into the ledger.

Bitcoin is a digital currency that introduced the idea of a blockchain to track how much of its currency (BTC) is held by each account, and to write “smart” transactions for payments. Transactions are added to the blockchain in a batch (called a block) by a network participant (called a miner), and miners include a special transaction that pays them newly minted BTC (called a coinbase transaction). The amount of new BTC released to miners follows a schedule built into the protocol and will decrement over time, eventually reaching zero once a determined amount of BTC has been made available.

Ethereum and DApp Primer

Ethereum is a blockchain protocol with a BTC-esque cryptocurrency called ether (ETH). To a degree much greater than Bitcoin, Ethereum allows users to code verbose smart contracts or decentralized applications (DApps), which can be stored on the blockchain for a fee. Once a DApp is deployed, users can run its functions (again, for a fee). The functions are executed by the miners, and the output is written to the blockchain. Among other things, a DApp can receive and store ETH and define functions for how ETH can be transferred from the DApp. DApps can also create their own currencies and circulate them as tokens. ERC20 tokens are compliant with a widely used Ethereum standard and can interoperate with existing wallet software, Web-based exchanges, and token-tracking websites.

Figure 3. Comparative evaluation of mechanisms to design stablecoins.

Mechanism	Corrects undervaluation	Corrects overvaluation	Decentralizes issuance	Decentralizes redemption	Decentralizes transfer	No trusted oracle
	Price		Trust			
Traditional Digital Cash	●	●	○	○	○	●
Traditional Cryptocurrency	○	○	●	×	●	●
Directly Backed and Redeemable	●	●	○	○	●	●
Directly Backed	○	●	○	○	●	●
Indirectly Backed	◐	●	●	●	●	○
Money Supply Adjustments	?	◐	●	×	●	◐
Asset Transfer	?	◐	●	×	●	◐

Prices

A cryptocurrency (like any asset) has two prices: the most someone is willing to pay; and the least someone is willing to sell for. These are referred to as the best bid price and best offer (or ask) price, respectively. Note the best bid price should logically be less than the best offer price; otherwise, an exchange would happen (such prices might occasionally “cross,” but this should be temporal and quickly resolved with an exchange). Say a stablecoin is designed to ensure one unit is always priced at \$1 USD. To argue stability, one must show both that the bid price should never exceed \$1 and that the offer price should never dip below \$1. Note, conversely, that bids can dip below \$1 (everyone prefers to pay less than something is worth) and asks can exceed \$1 (everyone prefers to receive more than something is worth).

ing blockchain-based assets¹⁰).

Directly backed. What if a stablecoin operates exactly as in the previous section but does not offer a redemption process for the coin’s underlying assets? If there is no clear assertion of redemption, the project is listed as directly backed in Figure 2.

Working Example: Alice is a trusted third party that issues 1,000 AliceCoins as ERC20 tokens. She asks \$1 USD for 1 AliceCoin and promises to deposit and hold the payment in a bank account. As before, Alice creates new AliceCoins when she runs out and publishes frequent bank statements. She offers no direct redemption of AliceCoins for U.S. dollars.

Here, bids will not exceed \$1 for the same reason as mentioned previously. There is no longer a way to profit, however, if offers vary between \$0 USD and \$1 USD (that is, the mechanism does not prevent undervaluation). Generally, coins in this category are, in fact, redeemable by one user: the company operating the coin. It could purchase undervalued coins to release \$1 USD from its reserves. For this reason, stablecoins in this category are scrutinized (to the extent made possible by the operating firm) to ensure reserves are intact. If every AliceCoin is not backed by \$1 USD, Alice could overissue AliceCoins to enrich herself.

The largest coin in this category is Tether. Tether claims to be redeemable, but the redemption process is reported by users to have a lot of friction, the firm is accused of issuing coins to manipulate markets,⁵ and the firm has not always maintained full reserves of U.S. dollars to allow all Tether to be redeemed (for these reasons, we categorize it here). To many, it is a mystery

why Tether remains highly liquid with daily trading volumes exceeding all other cryptocurrencies in value (according to *CoinMarketCap* at the time of writing) including Bitcoin. One explanation is that it is too useful to fail.

A key use case, illustrated by Tether and the affiliated exchange Bitfinex, is as a temporary store of value for traders and speculators. Traders who want to divest their BTC for U.S. dollars have three options: (1) Hold the U.S. dollars in an exchange account, which can be used only on the same exchange and requires the exchange to be a trustworthy custodian; (2) withdraw the U.S. dollars from the exchange, but this requires identity verification (in most jurisdictions), a bank that will accept proceeds of cryptocurrency trading, and a substantial time delay; (3) exchange BTC into a stablecoin that can be withdrawn from the exchange (that is, moved from the exchange to Alice’s private key) with little friction, delay, or regulatory oversight. This third option is a balanced alternative—the withdrawn stablecoin can be moved onto a different exchange, transferred to other users, or used for direct purchases without involving the original exchange. In short, it offers more flexibility than leaving U.S. dollars in an exchange account and less friction than withdrawing U.S. dollars.

Indirectly backed. Both of the previous mechanisms—directly backed and redeemable, and directly backed—place heavy trust assumptions on the company operating the currency (recall Figure 3). Could a currency be managed autonomously by a DApp? The key idea of this mechanism is to offer a redeemable token that can be converted into \$1 USD worth of ETH at the going USD/ETH exchange rate. Therefore, the amount of ETH received will grow or shrink depending on the exchange rate. Because a

blockchain has no inherent knowledge of exchange rates, this mechanism still requires one trustworthy entity called an *oracle* to write the exchange rate into the blockchain (or consensus can be taken across a set of oracles).

Working Example: Alice is no longer assumed to be trustworthy. She sets up a DApp that can hold ETH and issue tokens. The DApp determines how much ETH is equivalent to \$1.50 USD using the current exchange rate, provided to the DApp by a trusted third-party oracle, and Alice deposits this amount of ETH into the DApp. The DApp issues to Alice two places in a line—each place is a transferrable token. At some future time, the holder of the first place in line can redeem up to \$1 USD worth of the deposited ETH at the going exchange rate, and the holder of the second place in line gets any remaining ETH. Alice will transfer the first place in line (as a stablecoin called AliceCoin) to Bob for \$1 USD and will hold or sell the second place in line. When Bob redeems the AliceCoin, it will be worth \$1 USD in ETH when the entire deposit of ETH is worth more than \$1 USD. If the exchange rate drops enough, the entire deposit will be worth less than \$1 USD—Bob will get all of the deposit, and the holder of the second place in line will get nothing.

Bids for an AliceCoin in excess of \$1 USD will be fulfilled as long as there are individuals like Alice willing to lock up a deposit of ETH that is 1.5 times the face value of what they receive (this is called over-collateralization). An AliceCoin offered for less than \$1 USD can be purchased and redeemed for a profit, assuming the DApp holds enough ETH. Otherwise, an AliceCoin will sell between \$0 and \$1 USD according to the value of the ETH held by the DApp.

Is it risky for Alice to offer such an AliceCoin? Holding the second place in line is more volatile than holding the ETH itself. This stability mechanism does not (and cannot) eliminate volatility; it simply pushes it from first place to second place in line. The second place in line, however, is never more than \$1 USD short of the full amount of ETH held in the DApp. By keeping the \$1 USD she received for the AliceCoin, Alice offsets any losses from the second

place in line. She has no more risk than holding ETH. The second place in line can also be sold to someone who is seeking risk: The token is a leveraged bet that ETH rises in value. Is it risky for Bob? In most conditions, holding an AliceCoin is purposefully the same as holding U.S. dollars. If the USD/ETH rate deteriorates quickly, however, the AliceCoin will use up its buffer and start to lose value (at the same rate as ETH).

Here are a few of the design decisions to consider when deploying an indirectly backed stablecoin: What should the overcollateralization ratio be (for example, 1.5x)? When can an AliceCoin be redeemed (for example, on demand, after an elapsed time, after movements in USD/ETH and so on)? How do you issue multiple AliceCoins (for example, collateral for each coin is held separately, or collateral for all coins are pooled together and coins are interchangeable)?

Type 2: Intervention-based Stablecoins

The second broad category of stablecoins encompasses those that propose independent currencies with algorithmic and/or human intervention mechanisms for providing stability.

Money supply adjustments. A trusted oracle provides the going exchange rate between the cryptocurrency and a stable-valued asset, such as the U.S. dollar. When the cryptocurrency gains value, the supply of the cryptocurrency is increased; when it loses value, the supply is decreased. This mechanism is based on how central banks have historically controlled their economies; however, the specifics of exchange-rate targeting have been abandoned by modern central banks after past failures.

That said, exchange rates are an example, and other financial indicators could be used: oracle-provided interest rates (should lending markets emerge) or purchasing power; on-blockchain metrics such as transaction volumes (should these prove robust against manipulation); or human discretion (such as central banks themselves⁴).

Allowing a cryptocurrency to expand is not difficult. Who receives the new currency is a design decision with options including: existing holders of the currency in proportion to their holdings; existing holders through a random lottery; miners; or a specific

entity such as a central bank. Determining who loses when the currency contracts is the primary challenge.

Working Example: Alice forks Bitcoin to create a new altcoin called AliceCoin. She tweaks the schedule for releasing new AliceCoins (called the *coinbase* amount) according to the rules outlined here. She sets up a trusted oracle for the latest exchange rate of AliceCoins to U.S. dollars. AliceCoin is programmed to apply an intervention when the price of an AliceCoin exceeds \$1.02 USD or dips below \$0.98 USD. If the price exceeds \$1.02 USD, the miner is allowed to increase the coinbase amount (determined by some mathematical relationship with how much the price exceeds \$1.02 USD). If the price dips under \$0.98 USD, the miner must decrease the coinbase amount based on the same relationship. The correctness of the claimed coinbase is verified by other miners in deciding to accept or reject a mined block, as per all other checked conditions in Bitcoin.

If many bids for AliceCoin exceed \$1.02 USD, some of the newly injected currency could be spent on obtaining U.S. dollars until all buyers willing to pay more than \$1.02 USD have purchased AliceCoins. This is merely a heuristical argument because there is no guarantee the recipients will spend the new currency on U.S. dollars, especially if demand for the dollar is falling. The justification for offers below \$0.98 is symmetric: The currency contractions could make holders less willing to spend it on U.S. dollars. If the price drop is caused by a lack of demand for AliceCoins rather than an oversupply, however, then removing supply will only thin out the market but not actually give traders incentive to trade and correct the undervaluation.

When the coinbase is increased or decreased dynamically (this is called an *elastic coinbase*), increases can be by any amount, but decreases cannot appear to go past zero. When the coinbase is exactly zero, miners still have incentive to mine because of the fees provided in the transactions. In fact, this is how Bitcoin will eventually (projected to happen in 2140) function once all BTC is created (how well it will work is debatable¹).

Could the coinbase go negative? Since miners are rewarded the sum of the coinbase and the transaction fees, a coinbase can indeed be moderately negative if the transaction fees are greater than the negative coinbase. Under this deployment, users are effectively burning their transaction fees to contract the money supply.

Asset transfer. The second subtype of intervention-based stability mechanism expands and contracts the supply of currency to influence its value; however, it uses a less direct contraction method, as shown in the following example.

Working Example: Alice instantiates a DApp with an ERC20 token called an AliceCoin. The DApp is programmed to apply an intervention when the price of an AliceCoin exceeds \$1.02 USD or dips below \$0.98 USD according to a trusted oracle. If the price exceeds \$1.02 USD, the DApp creates a new set of AliceCoins (as before, according to some mathematical relationship) and transfers them to users waiting in line for them. How do users wait in line? When the price dips under \$0.98 USD, the DApp creates new positions at the end of the line and auctions them off to the highest bidder. The payment for a place in line is made in AliceCoins from the bidder to the DApp, and the DApp destroys the payment. The place in line is a transferrable token. If the line is empty, AliceCoins are distributed according to a fallback policy.

If many bids in excess of \$1.02 USD remain unexecuted, the logic follows the previous section: The currency is handed out in hopes that more U.S. dollars will be purchased. Offers below \$0.98 are justified on the premise that individuals will buy places in line, and if this premise is true, the resulting contraction of the currency follows the same logic as the previous section. The purchase of a spot in line is highly speculative—the currency might not return to stability and the spot might never be reached. As the line gets longer, the price of a place in line falls, and the speculative market thins out to traders wanting a higher and higher risk/reward ratio. These trends do not guarantee, or even point toward, a recovery in price.

Discussion and Conclusion

In summary, some stablecoins tokenize a low-volatility coin and bring it onto the blockchain. Others generally play one of two tricks: The first is to expand and contract the amount of currency to stabilize the value; the second is to turn two high-volatility coins (for example, of the underlying cryptocurrency) into one stablecoin and one extremely volatile coin. This last trick is similar to other financial assets that do not reduce overall risk but instead push it from one tranche of the asset to another.

A more detailed version of this article is available as a white paper.³ It includes more details and discussion about the categories, some empirical studies of how stable these coins are, reasons stablecoins are never perfectly stable, and an evaluation of whether Ethereum's mechanism for paying for computation (gas) is stable or not (the answer: it does not seem to be, for now).

Figure 4 is taken from the white paper and shows volatility in prices for two fiat currencies (Canadian dollar [CAD] and Euro [EUR]) and two stablecoins (Tether and BitUSD) against USD and BTC (prices from January 2017 to November 2018; 1000 mBTC = 1 BTC). A vertical line segment indicates the currency correlates with USD, while horizontal correlates with BTC. While CAD and EUR are free-floating currencies, they demonstrate a degree of stability not that different from the stablecoins, which demonstrates the stability

of similar central banking operations in these economic zones. (See the sidebar entitled "Prices.")

Why are there so many stablecoin projects? The differentiation among coins is along a few parameters: the type of asset that can be redeemed for the coin: USD, EUR, gold, and so on; the underlying blockchain (for example, Bitcoin, Ethereum, among others) and the low-level technical design (updatable contracts, governance, among others); and the country it operates from, which determines the degree of regulatory compliance that is required.

What's next? Self-sovereign stablecoins are interesting and probably here to stay; however, they face numerous regulatory hurdles from banking, financial tracking, and (likely) securities laws. For stablecoins backed by a governmental currency, the ultimate expression would be a centrally banked digital currency (CBDC). Since paper currency has been in steady decline (and disproportionately for legitimate transactions¹¹), a CBDC could reintroduce cash with technological advantages and efficient settlement while minimizing user fees.

Acknowledgments

J. Clark acknowledges support for this research project from the AMF (Autorité des Marchés Financiers) and the National Sciences and Engineering Research Council (NSERC)/Raymond Chabot Grant Thornton/

Catallaxy Industrial Research Chair in Blockchain Technologies. S. Moosavi acknowledges support from Fonds de Recherche du Québec – Nature et Technologies (FRQNT). □

Related articles on queue.acm.org

Bitcoin's Academic Pedigree

Arvind Narayanan and Jeremy Clark
<https://queue.acm.org/detail.cfm?id=3136559>

Blockchain Technology: What Is It Good For?

Scott Ruoti, Ben Kaiser, Arkady Yerukhimovich, Jeremy Clark, and Robert Cunningham
<https://queue.acm.org/detail.cfm?id=3376896>

A Hitchhiker's Guide to the Blockchain Universe

Jim Waldo
<https://queue.acm.org/detail.cfm?id=3305265>

References

- Carlsten, M., Kalodner, H., Weinberg, S.M., Narayanan, A. On the instability of bitcoin without the block reward. In *Proceedings of the ACM SIGSAC Conf. Computer and Communications Security*, 2016, 154–167.
- Champagne, P. *The Book of Satoshi: The Collected Writings of Bitcoin Creator Satoshi Nakamoto*. e53 Publishing, 2014.
- Clark, J., Demirag, D., Moosavi, S. SoK: Demystifying Stablecoins. *Social Sciences Research Network*, 2019; https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3466371.
- Danezis, G., Meiklejohn, S. Centrally banked cryptocurrencies. In *Proceedings of the Network and Distributed System Security Symp*, 2016.
- Griffin, J.M., Shams, A. Is Bitcoin really un-tethered? *Social Sciences Research Network*, 2018; https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3195066.
- Moin, A., Sekniqi, K., Siler, E.G. SoK: A classification framework for stablecoin designs. *Financial Cryptography*, 2020.
- Nakamoto, S. Bitcoin: a peer-to-peer electronic cash system, 2008; <https://bitcoin.org/bitcoin.pdf>.
- Narayanan, A., et al. *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, Princeton, NJ, 2016.
- Pernice, I.G.A., et al. Monetary stabilization in cryptocurrencies: design approaches and open questions. In *Proceeding of the IEEE Crypto Valley Conf. Blockchain Technology*, 2019.
- Pimentel, E., Boulianne, E., Eskandari, S., Clark, J. Systemizing the challenges of auditing blockchain-based assets. *Social Sciences Research Network Electronic J.*, 2019.
- Rogoff, K.S. *The Curse of Cash: How Large-Denomination Bills Aid Crime and Tax Evasion and Constrain Monetary Policy*. Princeton University Press, Princeton, NJ, 2017.

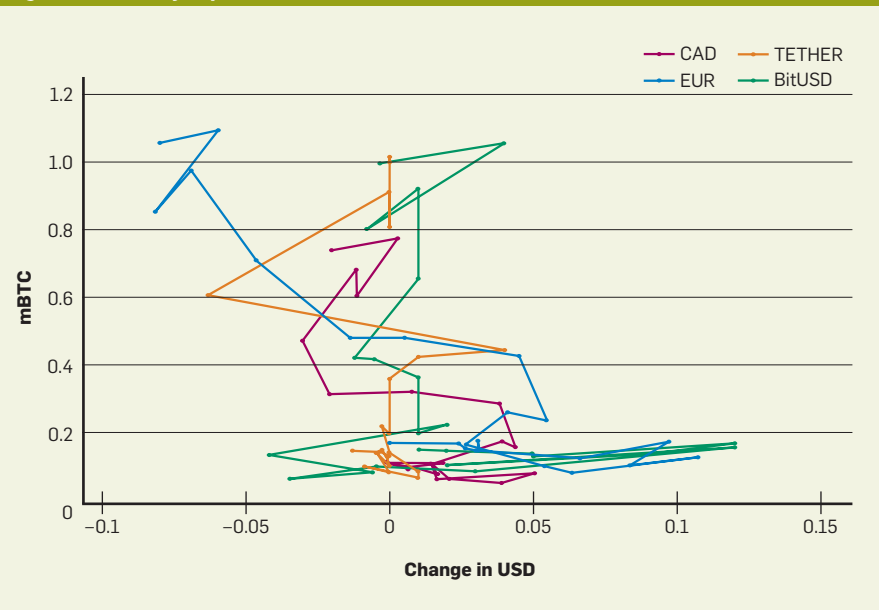
Jeremy Clark is an associate professor at the Concordia Institute for Information Systems Engineering in Montreal, Canada, where he holds the NSERC/Raymond Chabot Grant Thornton/Catallaxy Industrial Research Chair in Blockchain Technologies.

Didem Demirag is a Ph.D. student at the Concordia Institute for Information Systems Engineering in Montreal, Canada. She is working on realizing secure function evaluation using blockchain and is an intern at Autorité des Marchés Financiers, Montreal.

Seyedehmahsa Moosavi is a Ph.D. student at the Concordia Institute for Information Systems Engineering in Montreal, Canada, focusing on understanding the future of financial technologies using blockchains.

Copyright held by authors/owners.
 Publications rights licensed to ACM.

Figure 4. Volatility in prices for fiat currencies and stablecoins.



SHAPE THE FUTURE OF COMPUTING. JOIN ACM TODAY.

www.acm.org/join/CAPP

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

- Professional Membership: \$99 USD
- Professional Membership plus ACM Digital Library: \$198 USD (\$99 dues + \$99 DL)

ACM STUDENT MEMBERSHIP:

- Student Membership: \$19 USD
- Student Membership plus ACM Digital Library: \$42 USD
- Student Membership plus Print *CACM* Magazine: \$42 USD
- Student Membership with ACM Digital Library plus Print *CACM* Magazine: \$62 USD

- Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in computing. Membership in ACM-W is open to all ACM members and is free of charge.

PAYMENT INFORMATION

Name

Mailing Address

City/State/Province

ZIP/Postal Code/Country

- Please do not release my postal address to third parties

Email Address

- Yes, please send me ACM Announcements via email
- No, please do not send me ACM Announcements via email

- AMEX VISA/MasterCard Check/money order

Credit Card #

Exp. Date

Signature

Purposes of ACM

ACM is dedicated to:

- 1) Advancing the art, science, engineering, and application of information technology
- 2) Fostering the open interchange of information to serve both professionals and the public
- 3) Promoting the highest professional and ethics standards

By joining ACM, I agree to abide by ACM's Code of Ethics (www.acm.org/code-of-ethics) and ACM's Policy Against Harassment (www.acm.org/about-acm/policy-against-harassment).

I acknowledge ACM's Policy Against Harassment and agree that behavior such as the following will constitute grounds for actions against me:

- Abusive action directed at an individual, such as threats, intimidation, or bullying
- Racism, homophobia, or other behavior that discriminates against a group or class of people
- Sexual harassment of any kind, such as unwelcome sexual advances or words/actions of a sexual nature

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.



ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)
Hours: 8:30AM - 4:30PM (US EST)

Fax: 212-944-1318
acmhelp@acm.org
www.acm.org/join/CAPP

DOI:10.1145/3361682

DSAs gain efficiency from specialization and performance from parallelism.

BY WILLIAM J. DALLY, YATISH TURAKHIA, AND SONG HAN

Domain-Specific Hardware Accelerators

FROM THE SIMPLE embedded processor in your washing machine to powerful processors in data center servers, most computing today takes place on general-purpose programmable processors or CPUs. CPUs are attractive because they are easy to program and because large code bases exist for them. The programmability of CPUs stems from their execution of sequences of simple instructions, such as ADD or BRANCH; however, the energy required to fetch and interpret an instruction is 10× to 4000× more than that required to perform a simple operation such as ADD. This high overhead was acceptable when processor performance and efficiency were scaling according to Moore's Law.³² One could simply wait and an existing application would run faster and more efficiently. Our economy has become dependent on these increases in computing performance and efficiency to enable new features and new applications. Today, Moore's Law has largely ended,¹² and we must

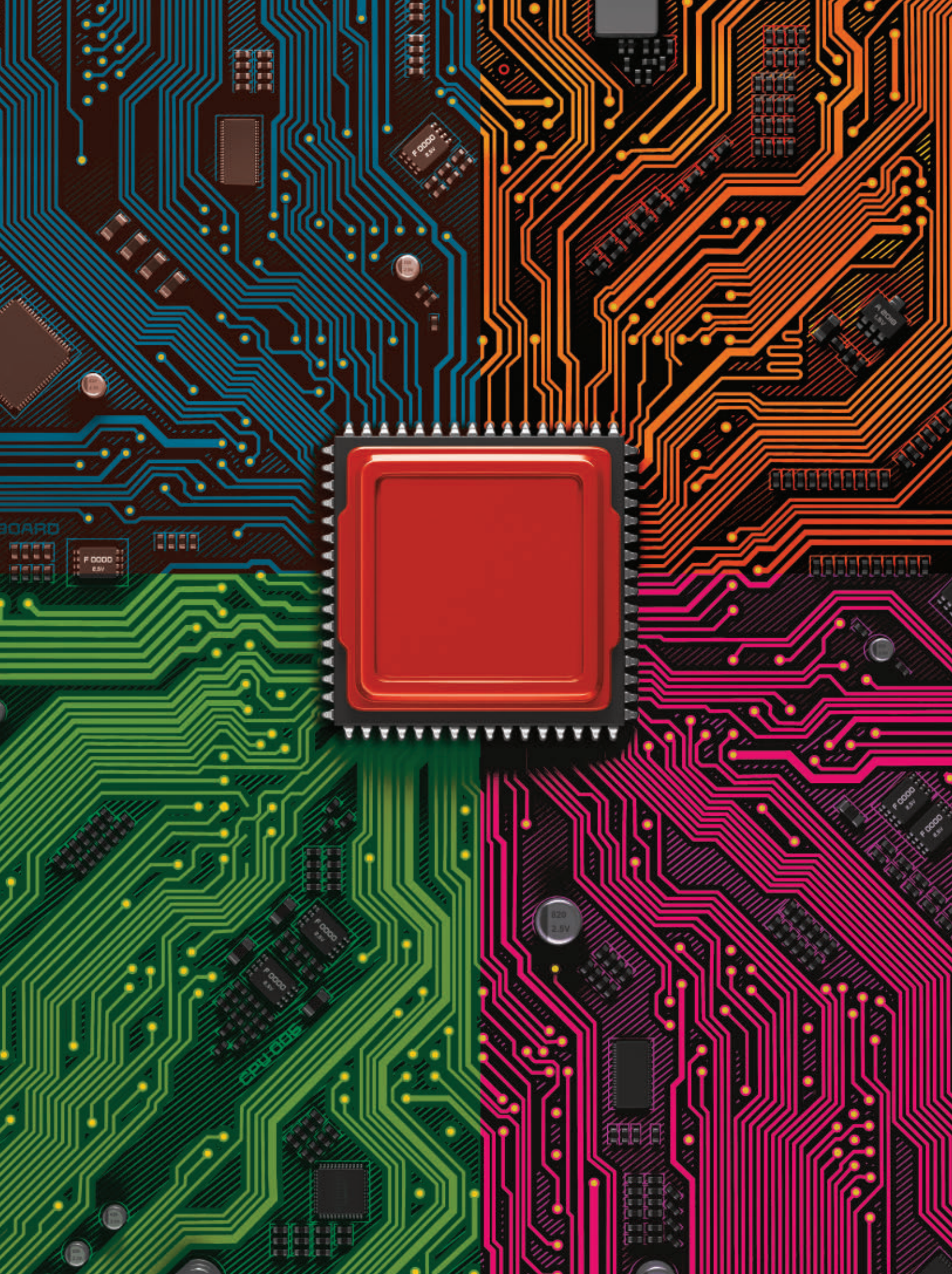
look to alternative architectures with lower overhead, such as domain-specific accelerators, to continue scaling of performance and efficiency. There are several ways to realize domain-specific accelerators as discussed in the sidebar on accelerator options.

A domain-specific accelerator is a hardware computing engine that is specialized for a particular domain of applications. Accelerators have been designed for graphics,²⁶ deep learning,¹⁶ simulation,² bioinformatics,⁴⁹ image processing,³⁸ and many other tasks. Accelerators can offer orders of magnitude improvements in performance/cost and performance/W compared to general-purpose computers. For example, our bioinformatics accelerator, Darwin,⁴⁹ is up to 15,000× faster than a CPU at reference-based, long-read assembly. The performance and efficiency of accelerators is due to a combination of specialized operations, parallelism, efficient memory systems, and reduction of overhead. Domain-specific accelerators⁷ are becoming more pervasive and more visible, because they are one of the few remaining ways to continue to improve performance and efficiency now that Moore's Law has ended.²²

Most applications require modifications to achieve high speed up on

>> key insights

- **Most speedup comes from parallelism enabled by specialization—the main source of efficiency.**
- **The underlying algorithms often have to change—trading increased hardware-friendly computation for reduced memory bandwidth demands.**
- **Accelerator design is really parallel programming guided by a cost model—arithmetic is free and global memory is expensive.**
- **Memory typically dominates both area and power of domain-specific accelerators.**
- **Specialized instructions give much of the advantage of a DSA at a fraction of the development cost and while retaining programmability.**
- **Domain-specific accelerators are one of the few ways to continue scaling the performance and efficiency of computing hardware.**



domain-specific accelerators. These applications are highly tuned to balance the performance of conventional processors with their memory systems. When specialization reduces the cost of processing to near zero, they become memory limited. The application must be reworked, codesigning the application with the accelerator, to reduce memory bandwidth and memory footprint. Even after rework, many domain-specific accelerators remain memory dominated.

A well-designed accelerator covers the broadest possible space of applications—accelerating a domain rather than a single application. Adding domain-specific instructions to a programmable processor provides the efficiency of the specialized instruction while retaining flexibility. Complex instructions give better efficiency because they amortize the high overhead of programmability. Building a parallel computer from domain-specific processing elements can also accelerate a large domain of applications with only a small loss of efficiency.

The design of a domain-specific accelerator is really a form of parallel programming, but with a cost model very different from what most programmers use. Arithmetic and logical operations are nearly free, and memory accesses have a cost that is a function of the size of the memory being accessed. Most of the effort in designing an accelerator is refactoring the application to optimize efficiency under this model. We envision future programming systems where the programmer specifies the algorithm and a mapping to hardware in space and time. From this description, the detailed design of the accelerator would be largely automated. Such tools will facilitate the rapid exploration of the accelerator design space and eliminate many of today's obstacles to accelerator design.

The remainder of this article describes the current state of the art in domain-specific accelerators. We start by discussing the four techniques accelerators employ to achieve performance and efficiency: specialization, parallelism, local and optimized memory, and reduced overhead. We then explore the process of codesigning applications and accelerators and we discuss how most accelerators are memory

dominated. The challenge of balancing specialization with generality is examined, and later we describe how accelerator design can be viewed as designing parallel programs with a set of costs reflecting modern hardware.

Sources of Acceleration

Domain-specific accelerators exploit four main techniques for performance and efficiency gains:

Data specialization: Specialized operations on domain-specific data types can do in one cycle what may take tens of cycles on a conventional computer. Specialized logic to perform an inner-loop function gains in both performance and efficiency.

Parallelism: High degrees of parallelism, often exploited at several levels, provide gains in performance. To be effective, the parallel units must exploit locality and make very few global memory references or their performance will be memory bound.

Local and optimized memory: By storing key data structures in many small, local memories, very high memory bandwidth can be achieved with low cost and energy. Access patterns to global memory are optimized to achieve the greatest possible memory bandwidth. Key data structures may be compressed to multiply bandwidth. Memory accesses are load-balanced across memory channels and carefully scheduled to maximize memory utilization.

Reduced overhead: Specializing hardware eliminates or reduces the overhead of program interpretation.

The speedup gains from specialization and parallelism are multiplicative. The dynamic programming engine described here, for example, gets a 37× speedup from specialization and an additional 4034× speedup from parallelism for a net 150,000× speedup compared to a conventional processor. Some of these factors are also dependent. Achieving high degrees of parallelism, for example, depends on locality. The 4096 processing elements in the dynamic programming engine only reference small local *traceback* memories. This degree of parallelism

would not be possible if global memory references were required. Optimizing memory may also rely on specialization. Compressing data structures may only make sense if specialized logic is available to do the compression.

Data specialization. The defining feature of many domain-specific accelerators is a set of hardware operations specialized to the application domain. The inner loops of many demanding applications perform tens to hundreds of arithmetic and logical operations with only very local memory references. In many cases, specialized logic can perform the entire inner loop in a single cycle with a small amount of area and power. This logic is fed by specialized registers and communication links that provide and consume data with very low energy. As an example, consider the Smith-Waterman algorithm⁴⁴ with affine gap penalties.¹⁴ This algorithm is widely used in genome analysis to align two gene sequences. Each iteration of the inner loop computes the following recurrence equations:

$$I(i,j) = \max \{H(i,j-1) - o, I(i,j-1) - e\} \quad (1)$$

$$D(i,j) = \max \{H(i-1,j) - o, D(i-1,j) - e\} \quad (2)$$

$$H(i,j) = \max \begin{cases} 0 \\ I(i,j) \\ D(i,j) \\ H(i-1,j-1) + W(r,q_j) \end{cases} \quad (3)$$

Here $H(i,j)$ is the maximum score for an alignment ending at (i,j) , o and e are the penalties for opening and extending an insertion or deletion, and $W(r,q)$ is the cost of substituting base r for base q . The computation is performed in 16-bit integer arithmetic.

Performing this computation on a conventional x86 processor without SIMD vectorization takes around 35 arithmetic and logical operations and 15 load/store operations. On an Intel Xeon E5-2620 4-issue, out-of-order 14nm CPU, each iteration takes about 37 cycles and consumes 81nJ. On our 40 nm Darwin accelerator, each iteration takes a single cycle, a 37× speedup, and consumes 3.1pJ, a 26,000× reduction in energy. Of the 3.1pJ, only 0.3pJ is consumed computing the recurrence equations. The balance of 2.8pJ is used for a single

memory access to store a 4-bit “traceback pointer” that identifies which preceding cell was used to compute the value.

A large fraction of the area and energy savings of specialization are due to elimination of overhead. Much of the 81nJ consumed by the x86 processor and much of its area are spent fetching, decoding, and reordering instructions. This overhead is largely eliminated by specialization. The processing element that computes the recurrence equations takes only 0.004mm² of die area in a 40nm process. Despite being three technology nodes behind the 14nm CPU, the specialized operations of the accelerator offer orders of magnitude improvement in performance, power, and area.

Specialization also enhances locality by reducing the cost of memory compression. In our EIE accelerator for sparse neural networks,¹⁶ we store 10%–30% dense networks in compressed-sparse-column (CSC) format. We further compress the row pointers to 4-bits each using run-length coding and compress the network weights using a 16-entry codebook. The compressed-sparse representation of network weights results in a 30× reduction in size allowing the weights of most networks to fit into efficient, local, on-chip memories, which takes two orders of magnitude less energy to access than off-chip memories.

On a conventional processor, the extra operations required to walk the pointers of the sparse-matrix data structure make such representations inefficient for densities above 1%. Similarly, the overhead of the run-length and codebook compression would be prohibitive on a general-purpose processor. With specialized logic, the pointer walking is done in a dedicated pipeline stage, with the pointers fetched from a dedicated, local memory. The decompression, both for the run-length pointer encoding and the codebook lookup, is also done in a dedicated pipeline stage. The area needed to support sparsity and compression with specialized logic is relatively small: the 16-entry weight decoder takes less than 1% of the die area; the pointer RAM takes about 20% of the area and power. On a general-purpose processor, the overhead is prohibitive.

Parallelism. Most domain-specific accelerators exploit parallelism at one or more levels. By specializing the par-

Acceleration Options

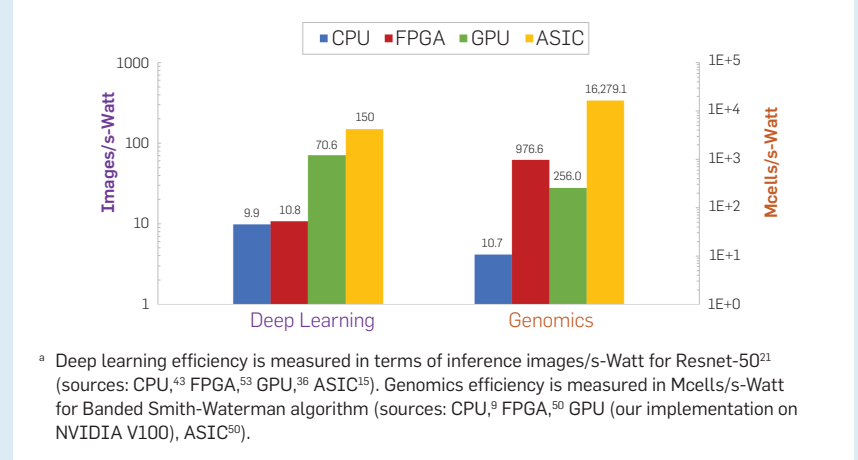
Domains of applications can be accelerated with ASICs, FPGAs, or GPUs each offering different trade-offs between development cost, programmability, and efficiency. ASICs (application-specific integrated circuits) provide the highest efficiency but have a high nonrecurring engineering (NRE) cost and poor programmability. Their logic is hardwired at design time for a single application domain. Soft logic in FPGAs lowers the efficiency for specific tasks by 10–100×²⁹ but enables the same chip to be dynamically configured for different applications, for example, for deep learning or genomics. Soft logic also allows for deeper specialization (for example, constant folding specific values of weights in a neural network⁵) and allows for an accelerator to be instantiated near the data it operates on, reducing communication costs.⁴⁷ GPUs are platforms that accelerate multiple domains by incorporating specialized operations (such as HMMA⁴) and memory optimizations (such as compressed surface storage⁵). For the applications they accelerate, they provide near-ASIC efficiency. For other applications, their SIMT execution model³³ offers order of magnitude better efficiency than CPUs at the expense of single-thread performance.

Figure 2 compares the efficiency of FPGAs, GPUs, and ASICs for two domains: deep learning and genomics. For domains where GPUs have specialized logic, such as deep learning, they provide near-ASIC efficiency.⁸ In other domains, such as genomics, GPUs provide lower efficiency than FPGAs but offer faster development time. For genomics, we coded the banded Smith-Waterman algorithm⁵⁰ in CUDA for the GPU in one day—giving 25× improvement in efficiency over the CPU. Bringing this algorithm up on an FPGA took two months of RTL design and performance tuning—to achieve four times the efficiency of the GPU. Hardening this RTL into an ASIC gives 16× the efficiency of the FPGA but with significant nonrecurring costs and lack of flexibility. Adding a dynamic-programming step (DPS) instruction to the GPU matches the efficiency of the ASIC and with no loss of efficiency.

There are architectures that provide intermediate trade-off points for programmability and efficiency between FPGAs and ASICs. CHARM⁶ uses a number of programmable domain-specific accelerator building blocks (ABBs), organized into ABB islands, to compose domain-specific acceleration. CGRAs²⁰ provide coarse grain reconfigurability, at word level or operator level instead of bit level, and incur lower overhead compared to FPGAs.

- a The NVIDIA T4 and Habana Goya have nearly identical arithmetic performance per Watt. The difference in the figure is due to the difference in memory interface, GDDR on the T4 and LPDDR on the Goya.

Figure 2. Comparison of computation efficiency (in Tasks/s-Watt) for CPU, FPGA, GPU, and ASIC for deep learning and genomics domains.^a



^a Deep learning efficiency is measured in terms of inference images/s-Watt for Resnet-50²¹ (sources: CPU,⁴³ FPGA,⁵³ GPU,³⁶ ASIC¹⁵). Genomics efficiency is measured in Mcells/s-Watt for Banded Smith-Waterman algorithm (sources: CPU,⁹ FPGA,⁵⁰ GPU (our implementation on NVIDIA V100), ASIC⁵⁰).

allelism to the application domain, the synchronization and communication between processing elements are greatly simplified. Only the communication and synchronization patterns in the application being accelerated need to be supported. By eliminating overhead, the parallel pro-

cessing elements can be made very simple and very small. As an example, the alignment portion of our Darwin accelerator exploits parallelism at two levels. At the outer-loop level, $A = 64$ systolic arrays of processing elements process 64 separate alignment problems in parallel. There is no communi-

cation between the subproblems, and the only synchronization required is upon completion of each subproblem. A typical reference-based assembly performs billions of alignments, so there is ample outer-loop parallelism.

At the inner-loop level, each array consists of $P = 64$ processing elements that compute 64 elements of the H , I , and D matrices in parallel. The computation is performed along an anti-diagonal of the matrices as originally suggested in Lipton and Lopresti.³⁰ On cycle t , processing element p computes the matrix elements at $(p, t - p)$. Matrices with more than P rows are processed in swaths P rows at a time. Because matrix element (i, j) depends only on the elements directly above $(i - 1, j)$, directly to the left $(i, j - 1)$, and above and to the left $(i - 1, j - 1)$, only systolic nearest neighbor communication between the processing elements is required. As with all systolic arrays, synchronization is implicit.

The parallelism exploited by special-purpose accelerators typically has very high utilization. Utilization at the outer-loop level is close to 100%. Until the very end of the computation, there is always another subproblem to process as soon as one finishes. With double buffering of the inputs and outputs, the arrays are working continuously. At the inner-loop level, utilization is 98.5%. Computation is performed in 512×512 tiles. At the start of each tile, only a single PE, at the upper left corner, is active. Each cycle another PE becomes active until all 64 are operating. Similarly, at the bottom right corner, the number of active PEs ramps linearly down from 64 to 0. Although it is possible to have the idle PEs start on the next alignment immediately, this is not done in Darwin. Idling of PEs at the start and end makes the average PE utilization 98.5% and the overall speedup due to parallelism 4034 \times . This speedup due to parallelism is multiplicative with the 37 \times speedup due to specialization giving an overall speedup on alignment of 150,000 \times .

In EIE, we parallelize a sparse matrix \times sparse vector multiplication by partitioning the rows of the matrix across 256 PEs. Each nonzero input activation and its column are broadcast to all PEs. Upon receipt of each activation, each PE walks the nonzero row entries for that column in its subset of rows,

accumulating row sums locally. Other than the activation broadcast, there is no communication between the PEs. A FIFO queue of pending input activations at each PE load balances work across the PEs, improving PE utilization from 50% without the FIFO to 90% with the FIFO.

Local and Optimized Memory. The gains from specialization and parallelism are dependent on keeping the computation supplied from small, local memories. Each cycle, each of the 4096 PEs in the Darwin alignment engine stores a *traceback pointer* to memory, achieving a net write bandwidth of nearly 2TBps. These pointers are used to construct the optimal alignment when the dynamic programming completes. If traceback pointers were stored to global memory, the computation would be bottlenecked by memory bandwidth. Instead, the traceback pointers are stored in 4096 small SRAMs, one associated with each PE. A conventional memory subsystem, even one with many levels of caches, is largely serial and would limit the achievable parallelism to a very small number.

In a similar manner, the filtering stage of the Darwin accelerator uses 16 dedicated SRAMs to store the bin counts, the number of seeds that match within a range of a candidate alignment. Although at most four bins are incremented each cycle, the speedup here is more than four times because the bin-count updates are random and cause interference with the sequential accesses to the seed position tables. With the bin-count updates removed from the memory stream, the sequential reads of the seed tables achieve nearly ideal memory throughput. Overall, the speedup from memory access optimization is 9 \times –24 \times —3 \times speedup from fewer accesses to DRAM (moving bin-count updates to SRAM) and 3 \times –8 \times speedup from the increased bandwidth by changing a random access pattern to mostly sequential. Four DRAM memory channels were added to the accelerator providing another four times the speed up from memory parallelism.

Data compression can be employed to both increase the effective size of a local memory and to increase the effective bandwidth of a memory interface. The NVDLA,³⁵ EIE,¹⁶ and SCNN,³⁷ for example, all store the weights of a

neural network as sparse data structures giving an average 3 \times –10 \times increase in the effective capacity of on-chip memories. The EIE and SCNN also run-length encode the pointers of the sparse data structure as 4-bit increments. This gives a density advantage of 4 \times –8 \times compared to storing these pointers in full 16- or 32-bit form. The weights in EIE are further compressed using a 16-entry codebook. Each weight is represented by a 4-bit codeword, giving an 8 \times savings compared to a 32-bit float. The savings in the number of weights and the number of bits per weight is multiplicative giving an overall compression rate of 32 \times –64 \times . Whenever weights are loaded from off-chip DRAM memory, the effective off-chip bandwidth is increased by this rate—compared to loading uncompressed data. GPUs have long stored surfaces in lossless compressed form³ to increase effective memory bandwidth.

Overhead reduction. Overhead reduction is an important aspect of specialization. Even a simple in-order processor spends over 90% of its energy on *overhead*: instruction fetch, instruction decode, data supply, and control.¹⁰ A modern out-of-order processor spends over 99.9% of its energy on overhead⁵¹ adding costs for branch prediction, speculation, register renaming, and instruction scheduling. Performing a 32-bit integer add takes only 63 fJ in 28nm CMOS.²⁴ Performing an integer add instruction on a 28nm ARM A-15 takes over 250pJ,⁵¹ about 4000 \times the energy of the add itself. Special purpose engines such as Darwin and EIE completely eliminate this overhead. Moreover, most adds do not need full 32-bit precision and just the number of bits needed are added, further saving energy. There are no instructions to be fetched and hence no instructions fetch and decode energy. There is no speculation, and hence no work lost due to mis-speculation. Most data is supplied directly from dedicated registers and thus no energy is required to read from a cache or from a large, multiported register file.

The high energy and area costs of instruction and data supply overhead motivate complex instructions. The energy of a single add operation is swamped by the instruction overhead energy. A complex instruction, such as the matrix-multiply-accumulate instruction (HMMA) of the NVIDIA Volta V100,⁴ on the other

hand, performs 128 floating-point operations in a single instruction and thus has an operation energy that is many times the instruction overhead. Using complex, specialized instructions, one can build efficient, specialized, programmable computer systems. We revisit the concept of complex, specialized instructions later.

Codesign is Needed

Achieving high speedups and gains in efficiency from specialized hardware usually requires modifying the underlying algorithm. Because existing algorithms are highly tuned for conventional general-purpose processors, they are rarely the optimal approach for a specialized solution. Instead, the algorithm and hardware must be *codesigned* to jointly optimize performance and efficiency while preserving or enhancing accuracy.

Many existing algorithms are tuned to balance the performance of conventional processors with their memory systems. When the cost of the processing is made nearly zero via specialization, they become completely memory dominated. To get significant speedup, such algorithms must be refactored to reduce the bandwidth demands on global memory. Although methods such as tiling³² and compression can be used to reduce global bandwidth to some degree, often more fundamental restructuring is required.

One approach to codesign is to trade more of an operation that is inexpensive in hardware (that is, logic limited) for less of an operation that is expensive (that is, memory limited). For example, conventional applications for long-read genomic sequence alignment such as GraphMap⁴⁵ spend most of their compute time on filtering and relatively little time on alignment. This approach makes sense for a general-purpose processor where filtering is relatively cheap and alignment is expensive. It is exactly the wrong optimization for specialized hardware where alignment can be made extremely efficient (26,000× more efficient and 150,000× faster than on a general-purpose processor) but filtering is fundamentally limited by global memory bandwidth. If we were to apply hardware specialization to the existing algorithm, we would be limited to a speedup of 4–5× due to the memory bandwidth required.

To exploit this difference in costs, Darwin spends 200× less time on filtering than GraphMap. This results in a 560× increase in candidate positions to be aligned and hence 560× more work for the alignment stage. However, because alignment is accelerated by 150,000×, the net result is a speedup of more than 200×. Darwin's parameters for filtering and alignment are adjusted so that the new alignment-heavy algorithm has equal or higher sensitivity than the filtering-heavy algorithm it replaces.

Codesign may also be used to reduce memory footprint—to make the use of small local memories feasible. The conventional Smith-Waterman algorithm for long, 10^4 base-pair, reads, for example, would require a prohibitively large, 10^8 entry, store for traceback pointers. To reduce the memory footprint to more feasible 2×10^5 entries, we developed the GACT algorithm that performs the dynamic programming in overlapping tiles.⁴⁹ Tiling reduces the memory footprint to that of a single 512×512 -entry tile. Overlapping the tiles by an amount $O = 128$ that is larger than the largest expected deviation of the optimal path from the diagonal in practice gives optimal alignments.

In other cases, codesign enables algorithms that would otherwise be inefficient on conventional hardware. For example, in Han et al.,^{17,18} we showed how neural networks could be pruned to 10%–30% density and compressed by 30×. The overhead of sparse methods on conventional hardware made these algorithms uninteresting except for memory compression. Codesigning special-purpose hardware for sparse operations enables these algorithms to be used to reduce computation as well.

As another example, software for whole genome alignment, such as LASTZ,¹⁹ uses ungapped extension to filter seed hits because gapped extension is prohibitively expensive on

conventional hardware. With specialized hardware, gapped extension becomes feasible⁵⁰ giving much better sensitivity when comparing the genomes of distantly-related species—where the alignments have frequent gaps.

Memory Dominates Accelerators

The area and power of most accelerators are dominated by memory, and their performance is often memory limited. As a result, much of the codesign described earlier is developing algorithms that have a small memory footprint. Most of their memory bandwidth requirements can be satisfied by small, local memories. They require only modest bandwidth from large, global memories.

Table 1 shows the relative area and power for memory and logic in the Darwin GACT accelerator, the Darwin D-SOFT accelerator, and the EIE sparse neural network accelerator. The EIE numbers are shown for 64 processing elements (PEs). D-SOFT and EIE, which accelerate a memory-limited application (seed filtering and matrix-vector multiplication, respectively) using large local memories, use over 90% of their die area for memory. In D-SOFT, the power component is over 90% as well, because the bin update logic is relatively simple, consisting of on-chip routing and simple arithmetic operations (add and compare). Even in the EIE, where the 16-bit multiply operations are more expensive, memory still consumes more than half of the chip power. For the GACT accelerator, which performs a compute-intensive dynamic programming operation, the memory that stores the traceback pointers consumes about 80% of the die area and over 75% of the power. The simple 16-bit additions and comparisons at the core of the dynamic programming recurrence equations take very little area and power. The low area and power of simple logic and arithmetic

Table 1. Breakdown of chip area and power into logic and memory units for the GACT and D-SOFT accelerators in Darwin⁴⁹ and for the EIE accelerator¹⁶ using TSMC 40 nm.

	Unit	Area (mm ²)	(%)	Power (W)	(%)
GACT	Logic	17.6	20.5	1.04	23.6
	Memory	68.0	79.5	3.36	76.4
D-SOFT	Logic	6.2	1.8	0.41	4.4
	Memory	320.3	98.2	8.80	95.6
EIE	Logic	2.8	6.9	0.23	40.3
	Memory	38.0	93.1	0.34	59.7

make domain-specific accelerators efficient. However, it also makes them memory limited. When logic is “free,” memory dominates.

Because the area and power of most accelerators are memory dominated, a reasonable first estimate of these costs can be made by considering only the memory. This allows rapid design space exploration.

Because global memory bandwidth is extremely expensive, many accelerators are designed to be global memory limited—to keep this expensive resource busy. In Darwin, for example, the four DRAM memory channels provide at most four seeds per cycle. We sized the D-SOFT filtering hardware with 16 bin-count banks so it can always keep up with the four seeds per cycle from external DRAM. Similarly, the 4K GACT PEs are provisioned so that alignment can keep ahead of the filtering stage.

Because external memory bandwidth is so critical, it should be optimized. Memory schedulers should be employed that maximize memory throughput⁴¹ and memory contents should be compressed where possible.

Balancing Specialization and Generality

In the design of domain-specific accelerators, there is a tension between generality and efficiency. Building an engine specialized for just one application can give the highest possible efficiency. However, its range of use may be too limited to generate enough volume to recover design costs, or a new algorithm may be developed rendering the accelerator obsolete. Building a completely general-purpose computer, on

the other hand, would result in poor efficiency. A happy medium lies in building an engine that accelerates a *domain* of applications where the breadth of applications is increased while retaining most of the efficiency of the completely specialized accelerator.

Special instructions vs. special engines. One approach to building accelerators for broad domains is to add specialized instructions to a general-purpose processor. A hardware block is built to accelerate the core operations for a domain of algorithms—matrix multiply for deep learning or dynamic programming for genomics—and the operations are made available as instructions on a general-purpose processor. This approach makes the core operation as efficient as a completely specialized accelerator but allows the use of the programmable general-purpose processor to adapt its use to different algorithms and applications.

The HMMA (half-precision matrix multiply-accumulate) instruction in the NVIDIA Volta V100 GPU⁴ is an example of adding a specialized instruction to a general-purpose processor. The instruction multiplies two 4×4 half-precision (16-bit) floating-point matrices accumulating the results in a 4×4 single-precision (32-bit) floating-point matrix. The Turing IMMA (integer matrix multiply accumulate) instruction performs this same operation on 8×8 8-bit integer matrices accumulating an 8×8 32-bit integer result matrix.²⁶ These operations accelerate the inner loops of both training and inference for convolutional, fully-connected, and recurrent layers of deep neural networks. A single HMMA instruction performs 128 floating-point

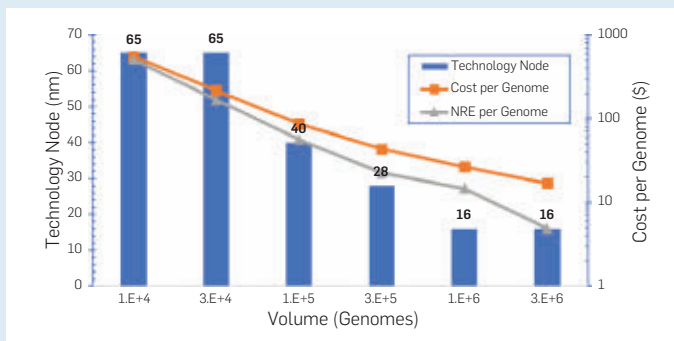
operations: 64 half-precision multiplies and 64 single-precision adds. An IMMA instruction performs 1024 integer operations. This large amount of math amortizes the overhead. Using data from Horowitz,²⁴ we estimate that when executing an HMMA (IMMA) instruction, 77% (87%) of the energy is consumed by arithmetic. The balance of the energy is consumed by instruction overhead and fetching the data operands from the large GPU register files and shared memory. A dedicated accelerator, such as the Google TPU,²⁷ could be at most 23% (13%) more efficient on half-precision (8-bit integer) matrix multiply. This bound is just for the core matrix multiply operation. The accelerator may be more efficient at staging data in on-chip memories and in optimizing data movement. Also, the GPU die will be larger and hence more expensive, because it includes area for the general-purpose functions, and for other accelerators, which are unused when doing matrix multiply. This die cost factors into the recurring portion of total cost of ownership.

The advantage of implementing the accelerator as an instruction is that the full power of the general-purpose processor is available to implement other layers of the network. Pooling, normalization, batch-normalization, sparsity-mask, and nonlinear function layers are easily implemented. As new algorithms and methods are developed, they are easily incorporated as custom layers while retaining the efficiency of the accelerator for the bulk of the operations.

In a similar manner, one could implement a special-purpose dynamic programming instruction to accelerate genomics calculations. A dynamic-programming step (DPS) instruction would take as input the values of H , I , and D (Eqs. (1)–(3)) for a portion of the current diagonal and generate corresponding values for a portion of the next diagonal along with their traceback pointers. Adding such an instruction to a general-purpose GPU or CPU would provide most of the efficiency gains of a hardwired accelerator such as Darwin.

One advantage of building a specialized instruction rather than an entire engine is that only the instruction must be developed, not the entire system. Most of the complexity of a computing

Figure 1. TCO for a genomics accelerator as a function of volume. At low volumes, older technology nodes give a lower TCO because of their lower nonrecurring costs (NRE).



engine, specialized or general purpose, is in the memory system, on-chip interconnect, I/O system, and global control. When a DSA is implemented by adding an instruction to a general-purpose GPU or CPU, it can leverage the existing system components. The complexity of the domain-specific block is 100s–1000s of times smaller than the complexity of the system (as measured by lines of code). With a dedicated engine, the entire system must be developed.

Today, architects pressed to increase efficiency are turning to complex instructions, such as HMMA, IMMA, and DPS to amortize fixed instruction overhead. Complex, domain-specific instructions enable the efficiency of domain-specific accelerators to be combined with the generality of a programmable processor and with development costs a fraction of that required to develop a dedicated accelerator.

Domain-specific parallel computers.

The ability to increase generality with little loss of efficiency is illustrated by comparing two simulation accelerators: the MSE¹¹ and MARS.² The MSE was a parallel computer where processing elements were highly specialized to different stages of the simulation pipeline. The MSE was 300× faster than a contemporary general-purpose computer (a VAX 11-780), but it could only accelerate switch-level simulation.

MARS used a single domain-specific programmable processing element that could serve as any of the pipeline stages. In a single cycle, each MARS PE could read a word from the input queue, perform an address calculation, read or write a word from external SRAM, extract a bit field from a word, perform an arithmetic or logical operation on the bit field, insert the resulting bit field into another word, and write a word to the output queue. The net result was a speedup of about 200× compared to a contemporary general-purpose computer (a Sun 3/260),² and this performance doubled over a period of years as the pipeline was refactored to eliminate bottlenecks and individual pipeline stages were tuned. MARS was nearly as fast as a hardwired engine, and because the area was dominated by memory, smaller, hardwired PEs would have made little difference to the overall area.

A key factor in the success of MARS was the low-overhead associated with horizontal microcode control. The energy overhead of programmability was largely the cost of fetching a 64-bit microinstruction from a 64-bit × 64-word microinstruction store. This energy was small compared to the access to a much larger SRAM made almost every cycle by most pipeline stages. The instruction fetch and control overhead of a conventional processor would have been prohibitive.

By defining the right domain-specific architecture, MARS was able to implement many simulation pipelines (and other tasks, such as speech recognition) with performance and efficiency approaching that of full-custom hardware. MARS proved so useful that it was reimplemented in five different generations of CMOS technology from 1.25 μm to 0.35 μm.

Total Cost of Ownership (TCO)

The technology node used to implement an accelerator should be chosen to give minimum total cost of ownership (TCO). As shown in Figure 1 for a genomics accelerator, at low volumes, the minimum TCO occurs at older (larger geometry) technology nodes. For each volume (number of de-novo, long-read genome assemblies), the bar shows the technology node that gives the minimum TCO. Nonrecurring costs are from Khazraee et al.²⁸ The lines show the nonrecurring engineering (NRE) cost per genome and TCO per genome for the minimum TCO technology. Darwin's 40nm technology gives minimum TCO at 10⁵ genomes assembled. Paying the high nonrecurring costs for a 16nm technology is not justified until a volume of 10⁶ genomes is reached. De-novo, long-read assembly of noisy reads on a CPU would cost around \$1,500 per genome,²⁵ so a custom accelerator gives lower TCO even for a volume as low as 10⁴ genomes—in a 65nm technology. At this point, the cost per genome is almost entirely NRE. Backend development costs are roughly the same for 130 nm and 65 nm (\$4.3M) and dominate mask costs, so nodes older than 65 nm do not offer a material savings in NRE.

A similar TCO calculation can be used to compare the cost of a dedi-

cated accelerator to the cost of adding specialized instructions to a CPU or GPU, or to the cost of combining several accelerators—perhaps sharing memory and I/O systems—on a single chip. Adding specialized instructions or combining accelerators gives a higher recurring cost but gives a larger volume over which to amortize the nonrecurring costs.

Accelerator Design

The design of a domain-specific accelerator is really the design of a fine-grained, memory-constrained, parallel program for a limited set of tasks. Most of the effort is in crafting an algorithm that achieves high parallelism with a small local memory footprint and low global memory bandwidth. Once a highly-parallel, highly local algorithm is developed, the design of the hardware is straightforward—and is largely dominated by memory as described previously.

The major difference between designing a DSA and writing a parallel program for a conventional parallel machine such as Summit²³ is the cost model. The cost model in turn drives differences in granularity and memory footprint. Most programmers use a cost model based loosely on the PRAM model.⁴⁰ Arithmetic functions and accesses to anywhere in a large global memory are all counted as unit-cost operations.

Even on a conventional x86 processor, the PRAM model is highly unrealistic, and on a modern GPU, even more so. Global memory operations are hundreds of times more expensive than arithmetic operations and local memory operations—those that hit in the cache. On multinode machines such as Summit, communication between nodes takes microseconds, the equivalent of thousands of operations. This leads to very coarse-grained parallelism and communication.^a Adjusting the PRAM model for the realities of conventional parallel machines has led to models such as log P⁸ that weight global accesses and communication with approximations of their actual cost.

a Many parallel machines have been built with very efficient hardware communication and synchronization.^{13,34,42} Unfortunately the need to use commodity processors for the nodes of mainstream machines has prevented most programmers from benefiting from such efficient mechanisms.

Accelerator costs. A DSA has a very different cost model than a machine such as Summit. If we use energy and area as a proxy for cost, a simple model is that arithmetic is free and accessing memory has a cost dependent on the size of memory being accessed. A more accurate cost model is as follows:

Arithmetic: In 14 nm technology, arithmetic costs range from 10fJ and $4 \mu\text{m}^2$ for an 8-bit add operation to 5pJ and $3600 \mu\text{m}^2$ for a double-precision floating-point multiply.²⁴ As described earlier, these costs are usually small compared to those of memory.

Local memory: Accessing a small (8KByte) local memory in 14nm costs 50fJ/bit and SRAM memory has an area of $0.013 \mu\text{m}^2$ per bit. The additional cost of accessing larger on-chip memories is the communication cost of getting to and from a small 8KByte subarray. This communication costs 100fJ/bit-mm, so the cost of accessing a memory of size S (in bits) is $50 + 0.022\sqrt{S}$ fJ. On-chip memories of up to several hundred megabytes can be realized in today's technology. A 100MB (800Mbit) memory has an access cost of about 0.7pJ/bit.

Global memory: Off-chip global memory is even more expensive. Accessing a relatively energy-efficient LPDDR4 memory costs about 4pJ/bit and higher-speed SDDR4 memory costs about 20pJ/bit.³¹ Global memory is also bandwidth limited. Memory bandwidth off of an accelerator chip is limited to about 400GB/s. Placing memories on interposers can give bandwidths up to 1TB/s, but at the expense of limited capacity.

Local Communication: Communication between blocks on chip has an energy cost that increases linearly with distance at a rate of 100fJ/bit-mm.

Global communication: High-speed off-chip channels use SerDes that have an energy of about 10pJ/bit.

Logic and local memory energies scale linearly with technology—as the capacitance of the devices scales down

while supply voltage is held constant.^b Communication energy remains roughly constant. This nonuniform scaling makes communication—such as nonlocal memory access—even more critical in future systems.

Programming accelerators. Each DSA requires firmware and a software development kit (SDK) to facilitate programming. Darwin-WGA,⁵⁰ for example, uses the OpenCL programming framework,⁴⁶ which provides a software API (in C/C++) for the two kernels it accelerates in hardware, Banded Smith-Waterman and GACT-X, along with a memory model API for exchanging data between the host and accelerator global memory. The application is then written in C++ with calls to this API. The API allows Darwin-WGA to be repurposed for different genomic applications, such as reference-guided assembly, *de novo* assembly, and cross-species whole genome alignments. Accelerators that support a more flexible domain-specific language (DSL), such as Halide,³⁹ or a broad software library, such as Tensorflow,¹ require adding a back-end to the domain-specific compiler to map the compiler IR to the accelerator. Back-end optimizations, particularly those that minimize off-chip data transfers, significantly impact accelerator performance.

Creating accelerators with programs. Although the accelerators we have built to date have been designed by directly writing Verilog RTL,⁴⁸ we envision a future in which an accelerator is designed by writing a parallel program describing the function of the accelerator along with mapping directives that specify how the computation and state of the program is mapped to hardware in space and time.

For example, for our dynamic programming accelerator, the program is largely Eqs. (1) through (3), along with a write to a traceback memory. We describe all possible parallelism and rely on dependence analysis to serialize the computation as required:

Algorithm 1: GACT

```
tb ← GACT(r, q)
```

^b For recent technology nodes, scaling linear dimensions by $0.7\times$ has given only a $0.8\text{--}0.9\times$ reduction in logic energy due to the complexity and overhead of current multi-patterned design rules.

```
input : r[TS], q[TS]
output : tb[TS,TS]
for i = 0..TS-1 do
    for j = 0..TS-1 do
        in(i,j) ← Max(h(i,j-1) - O, in(i,j-1) - E)
        del(i,j) ← Max(h(i-1,j) - O, del(i-1,j) - E)
        h(i,j) ← Max(0, in(i,j), del(i,j), h(i-1,j-1) + W(r[i],q[j]))
        tb[i,j] ← ComputeTb(h(i,j), in(i,j), del(i,j))
    end
end
```

In this pseudocode, the curved brackets (for example, $in(i, j)$) specify abstract indices. The square brackets indicate memory (for example, $tb[i, j]$). The recurrence matrices, in , del , and h are never fully materialized. Only the diagonal of indices needed for the current computation is held in storage at any given time. The input strings r and q are vectors of size TS (tile size) and the resulting traceback array tb is an array of size $TS \times TS$.

To map this computation to a processor array, we first declare the array and then specify the mapping. A straightforward mapping is described here. We declare an array of AS (array size) processing elements and an array of AS memory arrays each of the size $STRIPES \times TS$. We then map $h(i, j)$ to processing elements by row i and specify the time t each element is computed according to the diagonal wavefront. The in and del matrices (not shown) are mapped identically. The traceback matrix is mapped across the traceback memories by row.

Algorithm 2: Mapping

```
STRIPES ← TS / AS
processor_array p(AS)
memory_array tbm(AS)[STRIPES, TS]
Map h(i,j) → p(i % AS)
    at t = (i % AS) · TS + j - i / AS
Map tb[i,j] → tbm(i % AS)[i / AS, j]
```

We expect that having a programming system for accelerators of this type will facilitate the rapid exploration of alternative algorithms and mappings. A compilation tool can quickly determine the execution time and energy associated with a particular mapping. Once an efficient algorithm and mapping are settled on, the tool can generate the

RTL. More advanced tools could automate the generation of the mapping given constraints on time and space.

Conclusion

With the end of Moore's Law, domain-specific accelerators (DSAs) remain one of the few paths to continuing to increase the performance and efficiency of computing hardware. This paper has explored how DSAs achieve performance and efficiency drawing on the authors' designs of DSAs for genomics, deep learning, simulation, and graphics spanning four decades. DSAs gain much of their efficiency from specialization and elimination of overhead. This efficiency, in turn, enables parallelism, which accounts for much of the performance of DSAs. Most accelerators are memory dominated with much of their die area and power dissipation dominated by local memories. To benefit from specialization, many existing applications must be refactored to reduce their bandwidth demands on global memory.

A successful DSA accelerates a broad domain of applications. It may achieve such flexibility by adding specialized instructions to a programmable processor such as a GPU or CPU. Breadth can also be achieved by building a domain-specific parallel computer where domain-specific programmable processing elements carry out the processing of each pipeline stage in place of specialized logic. Such processing elements must be very lean to avoid losing much of the advantage of specialization to overhead.

Although DSAs today are designed at the RTL level, we envision a future where DSAs are designed by writing a parallel program and specifying the mapping of this program to hardware resources in time and space. Most of the intellectual effort in designing a DSA is a programming task: developing algorithms that give good performance and efficiency with the DSA cost model. Lowering this program to detailed hardware can be largely automated.

In the future, we expect many programmers will become designers of DSAs. An ecosystem will emerge to support these programmers with better tools to describe and evaluate their programs. Ultimately, we expect that computer science curricula will evolve to teach algo-

rithms and complexity with a cost model that more accurately reflects the reality of modern computing hardware. □

References

- Abadi, M., et al. Tensorflow: A system for large-scale machine learning. In *OSDI* (2016), 265–283.
- Agrawal, P., Dally, W.J. A hardware logic simulation system. *IEEE TCAD* 9, 1 (1990), 19–29.
- Beers, A.C., Agrawala, M., Chaddha, N. Rendering from compressed textures. *ACM Trans. Graph. (SIGGRAPH)* 96 (1996), 373–378.
- Choquette, J., Giroux, O., Foley, D. Volta: Performance and programmability. *IEEE Micro* 38, 2 (2018), 42–52.
- Chung, E., Fowers, J., et al. Serving DNNs in real time at datacenter scale with project brainwave. *IEEE Micro* 38, 2 (2018), 8–20.
- Cong, J., et al. Charm: A composable heterogeneous accelerator-rich microprocessor. In *ISLPED* (2012). ACM, 379–384.
- Cong, J., Sarkar, V., Reinman, G., Bui, A. Customizable domain-specific computing. *IEEE Des. Test Comput.* 28, 2 (2010), 6–15.
- Culler, D., Karp, R., et al. LogP: Towards a realistic model of parallel computation. In *ACM Sigplan Notices*, Vol. 28 (1993). ACM, 1–12.
- Dally, J. Parasail: SIMD C library for global, semi-global, and local pairwise sequence alignments. *BMC Bioinform.* 17, 1 (2016), 81.
- Dally, W.J., Balfour, J., Black-Shaffer, D., Chen, J., Harting, R.C., Parikh, V., Park, J., Sheffield, D. Efficient embedded computing. *Computer* 41, 7 (2008), 27–32.
- Dally, W.J., Bryant, R.E. A hardware architecture for switch-level simulation. *IEEE TCAD* 4, 3 (1985), 239–250.
- Esmailzadeh, H., Blem, E., Amant, R.S., et al. Dark silicon and the end of multicore scaling. In *ISCA* (2011). IEEE, 365–376.
- Fillo, M., Keckler, S.W., Dally, W.J., Carter, N.P., Chang, A., Gurevich, Y., Lee, W.S. The M-machine multicompiler. *Int. J. Parallel Program.* 25, 3 (1997), 183–212.
- Gotoh, O. An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 3 (1982), 705–708.
- Habana Labs. Goya Inference Platform White Paper v1.7, 2019. <https://tinyurl.com/yxlcfx54>
- Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M.A., Dally, W.J. EIE: Efficient inference engine on compressed deep neural network. In *ISCA* (2016). IEEE, 243–254.
- Han, S., Mao, H., Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *ICLR* (2016).
- Han, S., Pool, J., Tran, J., Dally, W. Learning both weights and connections for efficient neural network. In *NIPS* (2015), 1135–1143.
- Harris, R.S. Improved pairwise alignment of genomic DNA. PhD thesis, The Pennsylvania State University (2007).
- Hartenstein, R. Coarse grain re-configurable architecture (Embedded tutorial). In *ASPAC* (2001), ACM, 564–570.
- He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. In *CVPR* (2016), 770–778.
- Hennessy, J.L., Patterson, D.A. A new golden age for computer architecture. *Commun. ACM* 62, 2 (2019), 48–60.
- Hines, J. Stepping up to summit. *Comput. Sci. Eng.* 20, 2 (2018), 78–82.
- Horowitz, M. Computing's energy problem (and what we can do about it). In *ISSCC* (2014), IEEE, 10–14.
- Jain, M., Koren, S., Miga, K.H., Quick, J., Rand, A.C., Sasani, T.A., Tyson, J.R., Beggs, A.D., Dilthey, A.T., Fiddes, I.T., et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* 36, 4 (2018), 338.
- Jia, Z., Maggioni, M., Smith, J., Scarpazza, D.P. Dissecting the NVIDIA Turing T4 GPU via microbenchmarking. arXiv:1903.07486 (2019).
- Jouppi, N.P., Young, C., Patil, N., Patterson, D. A domain-specific architecture for deep neural networks. *Commun. ACM* 61, 9 (2018), 50–59.
- Khazraee, M., et al. Moonwalk: NRE optimization in ASIC clouds. In *Computer Architecture News*, Vol. 45 (2017). ACM, 511–526.
- Kuon, I., Rose, J. Measuring the gap between FPGAs and ASICs. *IEEE TCAD* 26, 2 (2007), 203–215.
- Lipton, R.J., Lopresti, D.P. Comparing Long Strings on a Short Systolic Array. Princeton University, Department of Computer Science, 1986.
- MICRON. System power calculators, 2019. <https://tinyurl.com/y5cvl857>
- Moore, G.E., et al. Craming more components onto integrated circuits. 1965
- Nickolls, J., Dally, W.J. The GPU computing era. *IEEE Micro* 30, 2 (2010), 56–69.
- Noakes, M.D., Wallach, D.A., Dally, W.J. The J-machine multicompiler: An architectural evaluation. *Comput. Arch. News* 21, 2 (1993), 224–235.
- NVIDIA. NVIDIA deep learning accelerator (NVDLA), 2017. <http://nvidia.org>
- NVIDIA. NVIDIA Tesla deep learning product performance, 2019. <https://tinyurl.com/y9u9amxh>
- Parashar, A., et al. SCNN: An accelerator for compressed-sparse convolutional neural networks. In *ISCA* (2017). IEEE, 27–40.
- Qadeer, W., et al. Convolution engine: Balancing efficiency & flexibility in specialized computing. In *Computer Architecture News*, Vol. 41 (2013). ACM, 24–35.
- Ragan-Kelley, J., et al. Halide: A language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. In *ACM Sigplan Notices*, Vol. 48 (2013). ACM, 519–530.
- Karpand, R.M., Ramachandran, V., Karpand, V., Karp, R.M. A survey of parallel algorithms for shared-memory machines. In *Handbook of Theoretical Computer Science*. North-Holland, 1988
- Rixner, S., Dally, W.J., Kapasi, U.J., Mattson, P., Owens, J.D. Memory access scheduling. In *Computer Architecture News*, Vol. 28 (2000). ACM, 128–138.
- Scott, S.L. Synchronization and communication in the T3E multiprocessor. In *ACM SIGPLAN Notices*, Vol. 31 (1996). ACM, 26–36.
- Shen, H., et al. Intel CPU outperforms NVIDIA GPU on ResNet-50 deep learning inference, 2019. <https://tinyurl.com/y6xewz8r>
- Smith, T.F., Waterman, M.S. Identification of common molecular subsequences. *J. Mol. Biol.* 147, 1 (1981), 195–197.
- Sović, I., Šikić, M., Wilm, A., Fenlon, S.N., Chen, S., Nagarajan, N. Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nat. Commun.* 7 (2016).
- Stone, J.E., et al. OpenCL: A parallel programming standard for heterogeneous computing systems. *Comput. Sci. Eng.* 12, 3 (2010), 66.
- Tan, T., Nurvitadhi, E., Chiou, D. Dark wires and the opportunities for reconfigurable logic. *IEEE Comput. Architect. Lett.* (2019).
- Thomas, D., Moorby, P. The Verilog® Hardware Description Language. Springer Science & Business Media, 2008.
- Turakhia, Y., Bejerano, G., Dally, W.J. Darwin: A genomics co-processor provides up to 15,000× acceleration on long read assembly. In *ASPLoS* (2018). ACM, 199–213.
- Turakhia, Y., Goenka, S.D., Bejerano, G., Dally, W.J. Darwin-WGA: A co-processor provides increased sensitivity in whole genome alignments with high speedup. In *HPCA* (2019). IEEE, 359–372.
- Vasilakis, E. An instruction level energy characterization of arm processors. Tech. Rep. FORTH-ICS/TR-450. Foundation of Research and Technology Hellas, Institute of Computer Science, 2015.
- Wolfe, M. Iteration space tiling for memory hierarchies. In *Proceedings of the Third SIAM Conference on Parallel Processing for Scientific Computing* (1987). Society for Industrial and Applied Mathematics, 357–361.
- Xilinx. Xilinx Imagenet Benchmarks, 2019. <https://tinyurl.com/y5l4ajff>

William J. Dally (bdally@nvidia.com), NVIDIA, Stanford University, CA, USA.

Yatish Turakhia (yturakhi@ucsc.edu), University of California, Santa Cruz, CA, USA.

Song Han (songhan@mit.edu), Massachusetts Institute of Technology, Cambridge, MA, USA.

Copyright held by authors/owners.
Publication rights licensed to ACM.



Watch the authors discuss this work in this exclusive *Communications* video.
<https://cacm.acm.org/videos/domain-specific-accelerators>

DOI:10.1145/3360646

A cycle that traces ways to define the landscape of data science.

BY VICTORIA STODDEN

The Data Science Life Cycle: A Disciplined Approach to Advancing Data Science as a Science

THE EDUCATION AND research enterprise is leveraging opportunities to accelerate science and discovery offered by computational and data-enabled technologies, often broadly referred to as data science. Ten years ago, we wrote that an “accurate image [of a scientific researcher] depicts a computer jockey working at all hours to launch experiments on computer servers.”⁸ Since then, the use of data and computation has exploded in academic and industry research, and interest in data science is widespread in universities and institutions. Two key questions emerge for the research enterprise: How to train

the next generation of researchers and scientists in the deeply computational and data-driven research methods and processes they will need and use? and How to support the use of these methods and processes to advance research and discovery across disparate disciplines and, in turn, define data science as a scientific discipline in its own right? An identifiable discipline of data science would encourage and reward research that fosters the continued development of computational and data-enabled methods and their successful integration into research and dissemination pipelines, as well as accelerating the generation of reliable knowledge from data science.

This article offers an intellectual framing to address these two key questions—called the Data Science Life Cycle—intended to aid decision makers in institutions, policy makers and funding agency leadership, as well as data science researchers and curriculum developers. The Data Science Life Cycle introduced here can be used as a framing principle to guide decision making in a variety of educational settings, pointing the way on topics such as: whether to develop new data science courses (and which ones) or rely on existing course offerings or a mix of both; whether to design data science curricula across existing degree granting units or work within them; how to relate new degrees and programmatic initiatives to ongoing research in data science and encourage the development of a recognized research area in data science itself; and

» key insights

- **For Data Science to emerge as a fully fledged science, it is essential to establish intellectual content, ensure knowledge organization, and incorporate external tests of validity for findings.**
- **The Data Science Life Cycle provides a flexible framework that knits stakeholder efforts together to advance Data Science as a science; providing a principled way to include topics such as ethics, reproducibility, and cyberinfrastructure for Data Science, as well as methodological, computational, and domain-specific subjects.**



IMAGE BY ANATOLI STOIKO

how to prioritize support for data science research across a variety of disciplinary domains. These can be difficult questions from an implementation point of view since university governance structures typically separate disciplines into effective siloes, with self-contained evaluation, degree-granting, and decision-making authority. Data science presents as a cross cutting methodological effort with the needs of a full-fledged science including: communities for idea sharing, review, and assessment; standards for reproducibility and replicability; journals and/or conferences; vehicles for disciplinary leadership and advancement; an un-

derstanding of its scope; and, broadly agreed-upon core curricula and subjects for training the next generation of researchers and educators.

After motivating the key data science challenges of interdisciplinarity and scope, this article presents the Data Science Life Cycle as a tool to enable the development of data science as a rigorous scientific discipline flexible enough to capitalize on unique institutional strengths and adapt to the needs of different research domains. Examples are given in curriculum development and steps to defining data science as a science.

Current Approaches to Data Science

There are currently four main approaches taken toward data science at post-secondary institutions and universities in the U.S., with some institutions opting to take more than one approach. The first model involves issuing data science degrees from an existing department or school, such as the computer science department (for example, University of Southern California, Carnegie Mellon University, University of Illinois at Urbana-Champaign), the statistics department (for example, Stanford University), a pro-

fessional studies or extension school (Northwestern University, Harvard University), engineering (Johns Hopkins University), or the School of Information (UC Berkeley). This approach can include innovative steps such as online course offerings or collaborative degrees that approximate data science. An example of the latter is the undergraduate CS+X degree pioneered by the computer science department at the University of Illinois at Urbana-Champaign, where CS refers to computer science and X refers to a domain specific discipline such as economics, anthropology, or linguistics. For a CS+X degree students receive a degree in discipline X with half their courses comprising a common core of computer science classes and half their courses from their disciplinary area X. Stanford University has a CS+X program for undergraduates designed as a joint major between computer science and the humanities. Data science itself has not been established as a sub-discipline in computer science or any other discipline to the best of my knowledge, nor is there an ACM Special Interest Group on Data Science.

The second approach to data science extends or transforms an existing department to explicitly include a home for all of data science, not just the data science degree programs. For example, the statistics department may be renamed Statistics and Data Science (for example, Yale University) or a School of Information Science or Informatics renamed to include the Data Science moniker (Drexel University). The third approach is to create a coordinating mechanism such as a Data Science institute or center at the university (Columbia University, University of Virginia, University of Delaware, University of Chicago, UC Berkeley). Such an institute tends not to have faculty lines, but affiliates faculty who have an appointment elsewhere on campus. It may grant certificates and/or degrees in coordination with affiliated faculty and units, and often began with a focus on professionals and executive education. The University of Washington, for example, extended an existing institute on campus, the eScience Institute, to house its cross-disciplinary Data Science initiative. The final approach is to bring the institute's major



The Data Science Life Cycle explicitly recognizes the need for data, software, and other artifacts, along with the research findings, to be made available to the community and enables recognition of the need for dedicated research on how this sharing is accomplished.



data science disciplinary units (for example, statistics, computer science and engineering, information science) together under one organizational umbrella to determine degree programs, grant degrees, and house faculty lines and data science research. This is the most recent approach, currently undertaken for example at UC Berkeley (to my knowledge Berkeley is also the only institution to explicitly articulate a Data Science Life Cycle when describing one of its data science degrees).

In some institutions, the trappings of data science have emerged organically within departments themselves without the data science label. For example, offering more classes in statistics and computational methods, or creating data facilities to manage the increasing volumes of data used in departmental research such as the Brain Imaging Data Structure (BIDS) in the Department of Psychology at Stanford University or the Data Analytics and Biostatistics Core in the Emory University School of Medicine. Established domain specific data repositories such as the Protein Data Bank can be central to established research and have long histories of knowledge and expertise development. As data science progresses, we would be remiss not to take the broad advances made by these efforts into account.

It is clear the potential of data science has captured the imagination of students and the broader society.¹⁸ In my experience, however, students can perceive a gap in our pedagogical offerings when it comes to supporting their interest in data science. For a student seeking to do advanced coursework in data science it can appear that statistics is not computational enough, computer science isn't data inference focused enough, information science is too broad, and the domain sciences do not provide a sufficiently deep pedagogical agenda in data science. The research context today is markedly different to even a decade ago in the use of computational and data-enabled methods in a wide range of long-established disciplines from biology (bioinformatics²³) to physics (computational physics²²) to mathematics (computer-enabled mathematical proofs¹²) to English

(quantitative analyses of literary texts¹³) to sociology (digital social science¹⁷), and students are asking the right questions about where data science fits in their education. Not only has it increased the types, scales, and sources of data-accelerated discovery,²⁵ data has opened new vistas of scientific investigation, methodological advances, and innovation through the creation of novel comprehensive datasets available to communities.^{5,16} Data science is inherently interdisciplinary, yet must have a coherent scope in order to develop as a discipline.

Defining Data Science as a Discipline: The Challenges of Interdisciplinarity and Scope

In what institutional unit or entity should a data science program reside, and what subject matter is considered within the scope of data science? These questions belie the two principal challenges to the advancement of data science as a discipline: its inherently interdisciplinary nature, and the lack of a well-defined scope.

Challenge 1. Data science is inherently interdisciplinary. Data science is emergent from a plurality of disciplines, a fact that has been widely noted.²⁸ These disciplines often exist in different parts of the institution, potentially posing coordination and implementation challenges both within the institution and for data science as an emerging field of research. Few would dispute the central role of data inference methods or software development in data science, yet even those two examples have different loci within the institutional structure: the former typically in a Department of Statistics (often situated in the Faculty of Arts and Sciences) and the latter in computer science departments (often located in the School of Engineering). In addition, schools of information science contribute expertise in data discovery, storage and retrieval, stewardship, archiving, and artifact reuse; engineering and the physical sciences disciplines perform deeply computational simulation-based research; and business schools advance business intelligence and carry out data analytics. The list of examples goes on. These disciplines contribute different but

necessary aspects of a data science discipline and many of the skills used in data science already exist in established departments.

Challenge 2: Data science must have a well-defined scope. Many definitions of data science have been put forward, indeed this publication presented its own in 2013: “Data science [involves] data and, by extension, statistics, or the systematic study of the organization, properties, and analysis of data and its role in inference, including our confidence in the inference” or, “Data science is the study of the generalizable extraction of knowledge from data.”⁶ Through conversations in 2013, the following definition was developed by Iain Johnstone, Peter Bickel, Bin Yu, and myself: “Data Science is the science of (collaboratively) generating, acquiring, managing, analyzing, carrying out inference, and reporting on data.” This broad scope means that data science covers a large proportion of the research carried out in institutions today, and implementations of data science programs can be markedly different at different institutions.²⁰

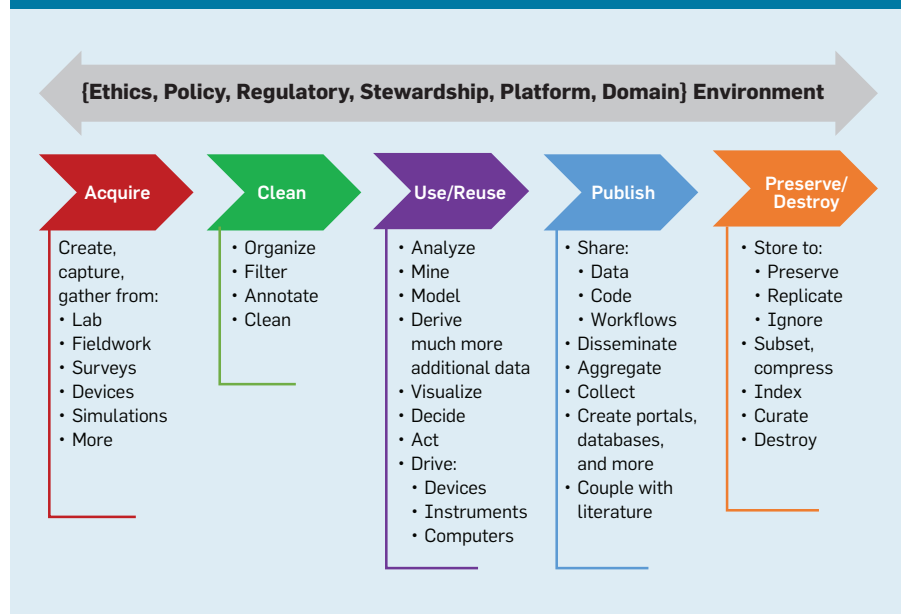
A Framing for Data Science: The Data Science Life Cycle

Although the Data Science Life Cycle is a new concept, it is an extension of “the Data Life Cycle,” which has a long history in the information sciences and many domain sciences.¹ The Data

Life Cycle describes the various stages a dataset traverses as it undergoes scientific collection and investigation and is typically used to guide data management decisions and practices. I extend this idea beyond its focus on data to describe the complete process of data science with the Data Science Life Cycle. This work extends research in the Data Life Cycle by focusing on the generation of scientific findings, and thereby including computational components, inferential methodology, and articulating a clear role for ethics and meta research within the scope of data science. It can also provide a foundational grounding for data science pedagogical program design.

Extending the concept of the data life cycle. Figure 1 shows a depiction of a Data Life Cycle, following a dataset from acquisition, through cleaning, use, publication of the resulting dataset, and then through to an eventual preserve/destroy decision for the dataset. It is important to note that there is no single fixed definition of a Data Life Cycle, rather it’s a thematic abstraction whose manifestation may change depending on the specific dataset or collection of datasets to which it is applied and the purpose of the data collection. A Data Science Life Cycle expands the area of focus beyond the dataset, to the complete bundle of artifacts (for example, data, code, workflow and computational environment information)

Figure 1. Example of a data life cycle and surrounding data ecosystem (reprinted with permission).¹



and knowledge (scientific results) produced in the course of data science research results.

Figure 2 shows a depiction of a Data Science Life Cycle describing stages of data science research, extending the Data Life Cycle reprinted in Figure 1. As in Figure 1, Figure 2 depicts an abstraction, intended to be customized to particular data science projects.

The act of scientific discovery in data science produces findings just like any area of research, and typically creates or leverages other artifacts as well, for example, the data used to support the findings and the code that produces the findings from the data (it may even produce other artifacts as well, for example, curriculum materials, software tools, and hardware prototypes). Research findings and artifacts are viewed with dissemination to

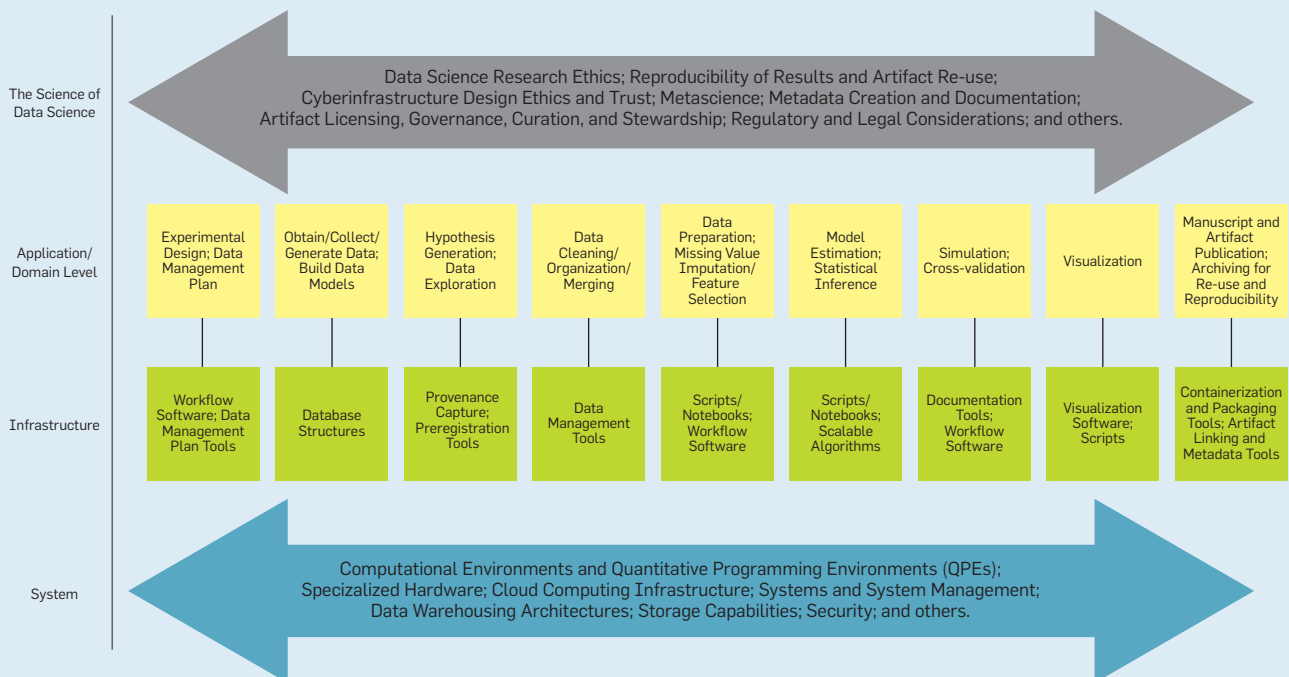
the research community at the point of publication when created. This is what is meant by the term “life cycle”—an explicit recognition that artifacts pass to the community at the point of publication, readied to begin the life cycle again in a new research effort, as inputs. The Data Science Life Cycle explicitly recognizes the need for data, software, and other artifacts, along with the research findings, to be made available to the community and enables recognition of the need for dedicated research on how this sharing is accomplished.

“Reproducibility of Results and Artifact Reuse” is listed as a topic in the overarching grey arrow in Figure 2. The life cycle approach allows a principled incorporation of the notion of computational reproducibility—the practice of ensuring artifacts and

computational information needed to regenerate computational results is openly available post-publication.^{4,15,28} Figure 2 emphasizes that artifact preservation activities occur both before and during computation, for the duration of the discovery process. An attempt to recreate computational and data manipulation steps for preservation purposes after publication can be difficult and time consuming, if not impossible. The Data Management Plan, required by the National Science Foundation and other science funders, is therefore included at the beginning of the Data Science Life Cycle, to emphasize the importance of early planning for the artifact preservation that will occur at the point of eventual publication (of the results as well as the supporting artifacts). The need for improved tools for document-

Figure 2. An example of a Data Science Life Cycle.


The light-yellow layer is the Application/Domain Science level. This level portrays the steps of a research project described at the domain level. The green layer beneath is the Infrastructure level, describing the computational infrastructure that enables the Application level. The blue arrow underlying both the application and infrastructure levels is the System level, describing system elements upon which the data science discovery process depends. Similarly, the overarching grey arrow is the meta-scientific level “The Science of Data Science.” The arrows are intended to depict the life cycle: that the output of data science discovery (the findings and the artifacts that enable reproducibility such as code, data, and workflow and computational environment information for example) are published for verification and reuse by others. The figure provides a way to consider tool use explicitly for each discovery step. Some tools may support more than one discovery step, for example, notebooks, and some application level steps may use more than one tool. Different steps in the discovery pipeline, whether at the domain, infrastructure, system, or science of data science level, may be carried out by different people.




tation and recording of the steps in the data science discovery process becomes evident with this approach as does greater recognition that the production of reusable research artifacts (for example, data, software that support a published scientific finding) is a valuable researcher activity.

Computational and meta-scientific aspects of data science must be explicitly considered. Crucially, the Data Science Life Cycle adds an additional dimension to the Data Life Cycle: the computational layer that enables data science research. A data scientist may proceed through the steps depicted in the Data Science Life Cycle in Figure 2: experimental design; obtaining/generating/collecting data; data exploration and hypothesis generation; data cleaning, merging, and organization; feature selection and data preparation; model estimation and statistical inference; simulation and cross-validation, visualization; publication and artifact preservation/archiving. This series of steps is called the “Application Level” (depicted in pale yellow in Figure 2), referring to the scientific application or domain of research. As noted, the Data Science Life Cycle is an abstraction and any particular research project may include a subset of these steps.

There are additional components beyond the Application Level in every data science project, depicted by the grey arrow across the top of Figure 2 mentioned earlier, including data science ethics; documentation of the research and meta data creation; reproducibility; and policy and legal aspects including governance, privacy, and intellectual property considerations.²⁶ This is the “Science of Data Science Level.” In addition, data science projects encompass computational skills and technologies (for example, interpreted languages such as R and Python, data querying languages, distributed computing resources) represented in the green, lower layer, called the “Infrastructure Level” of the Data Science Life Cycle. None of the technologies listed in Figure 2 are prescriptive but they support the steps in the Data Science Life Cycle, in particular the Application Level. Importantly, each are research areas of research and development in their own right, including



A life cycle approach encourages and enables a unification of views regarding data science and gives us a footing from which to adapt and evolve the practice and teaching of data science to research projects and to institutional strengths.



notebooks and workflow software; visualization tools; statistical inference languages; data management tools; and archiving and artifact linking tools. Running across the entire Data Science Life Cycle, and depicted in the blue arrow at the bottom of Figure 2, are the hardware and other technological structures on which the data science experiment is carried out, including compute infrastructure, cloud computing systems, data structures, storage capabilities, and quantitative programming environments (QPEs).⁹ This is called the System Level. Computational reproducibility is an important factor when deciding which artifacts and details in the discovery process to preserve and share. For example, information on how and why parameters were selected in model selection could be included in the documentation and workflow information. The Data Science Life Cycle highlights the various contributions made to the research by different people and could help indicate ways to give appropriate credit by including information on who has contributed what to the discovery process.

Two simplified examples of the Data Science Life Cycle in research settings. Here, I present two applications of the Data Science Life Cycle to simplified but representative descriptions of research that illustrate how this approach can surface nuanced and important aspects of data science in different settings. In the first example researchers wish to classify two types of cancer using gene expression data.¹⁰ ¹¹ The steps the authors describe for an experiment are as follows:

1. Obtain gene expression data (the data are already split into train/test subsets based on clinical conditions).
2. Normalize the data (including both train/test subsets).
3. Apply Recursive Feature Elimination:
 - a. Train classifier using Support Vector Machines (SVMs).
 - b. Compute a ranking criterion for each feature.
 - c. Remove features with the smallest ranking criteria.
 - d. Iterate until a tolerance threshold is reached.
4. Perform cross-tests with the baseline method from Golub et al.¹⁰ to compare gene sets and classifiers.

Mapping this experimental description to the Application Layer of the Data Science Life Cycle could proceed as follows: Obtain Data → Data Preparation → Feature Selection/Model Estimation → Cross-tests and Validation → Publication and Archiving. Information regarding the tools and software used for each step is then mapped to the Infrastructure Layer and overarching issues, such as data governance and sharing policies, detailed in the Science of Data Science Level. Notice this data science pipeline incorporates a cyclical loop in the pipeline when Recursive Feature Elimination is employed.

The second example gives a stylized description of hypothesis-driven research experiment to test whether a journal's impact factor is related to the existence of a data or code sharing author policy.²⁷ The steps are as follows:

1. Determine the hypothesis to test.
2. Design an appropriate experiment to test the hypothesis.
3. Collected data on journal impact factors and artifact policies as well as other descriptive information.
4. Test the hypothesis.
5. Report the results.

We map the steps to the Data Science Life Cycle as follows: Determine Hypothesis → Experimental Design → Collect Data → Statistical Inference → Publication. Computational tools used in each step can be detailed in the Infrastructure Level description, and issues that apply to the entire life cycle considered in the Science of Data Science Level, such as data and code availability, preregistration of hypothesis tests, Institutional Review Board (IRB) information, if relevant. Although simplified, these two examples represent different research questions and two different instantiations of the Data Science Life Cycle, but both show how the Data Science Life Cycle framework allows important aspects of the research, such as computational implementations and data ethics, to be cogently and deliberately incorporated as part of the research and publication process.

These examples also illustrate how the Data Science Life Cycle tests whether a particular research effort fits under the rubric of data science. Gaps at the Infrastructure or System Levels can be



Data science is benefitting from close association with industry as computer science did at its inception.



more easily detected and recognized as part of a comprehensive Data Science research agenda, including for example algorithms; containerization technologies; abstractions of data manipulations; data structures; distributed computing; parallel, cloud or edge computing; hardware design (for example, application specific integrated circuits and their development such as TPUs, or networking capabilities for data distribution).

Considering the Data Science Life Cycle as a life cycle enables a natural consideration of crucial overarching factors such as reproducibility, documentation and meta data, ethics, and archiving of research artifacts such as data and code. The Data Science Life Cycle provides guidance on the multifaceted set of skills and personnel needed for data science, for example “skills for dealing with organizational artifacts of large-scale cluster computing. The new skills cope with severe new constraints on algorithms posed by the multiprocessor/networked world.”⁷ Workforce development is therefore incorporated into the life cycle approach, which is especially germane to data science as “enthusiasm feeds on the notable successes scored in the last decade by brand-name global information technology (IT) enterprises, such as Google and Amazon.”⁷

The Data Science Life Cycle engages relevant stakeholders in the larger research community in a systematic way, including not only data science researchers but others such as archivists, libraries and librarians, legal experts, publishers, funding agencies, and scientific societies. It gives a framework to clarify how different contributions knit together to support each other to advance data science.

Leveraging the Data Science Life Cycle

A life cycle approach encourages and enables a unification of views regarding data science and gives us a footing from which to adapt and evolve the practice and teaching of data science to research projects and to institutional strengths. There are commonalities to nearly all data science efforts, for example, data wrangling, data inference, code writing, artifact creation and sharing. A common intellectual framework

can facilitate knowledge sharing about data science as a discipline across different the fields and domains using data science methods in their research.

A data science curriculum. Conceptualizing data science as a life cycle also gives a way to position classes and sequences to teach core and elective data science skills, indicating where existing courses may fit and where new courses may need to be developed. It helps define a curriculum by using the steps of the Data Science Life Cycle as a pedagogical sequence and provides for the inclusion of overarching topics such as data science ethics, and intellectual property, reproducibility, or data governance considerations.²⁴ Perhaps most importantly the Data Science Life Cycle can indicate courses that may be out of scope and new course topics essential to data science.

The accompanying table shows how several commonly offered courses could be matched to the steps described by the Data Science Life Cycle described in Figure 2. Although not included in the table, each step can be augmented by the creation of new targeted classes if needed, such as Data Policy, Reproducibility in Data Science, Data Science Ethics, Circuit Design for Deep Learning, Software Engineering Principles for Data Science, Mathematics for Data Science, Interoperability and Integration of Different Data Sources, Data Science with Streaming Data, Software Preservation and Archiving, Workflow Tools for Data Science, Intellectual Property for Scientific Code and Data. The list goes on. The addition of domain specific optional courses could define tracks or specializations within a data science curriculum (for example, Earth sciences, bioinformatics, sociology; cyberinfrastructure for data science) to create a potential DS+X degree in the spirit of the CS+X degrees discussed previously.

The emergence of a discipline of data science is necessary to advance data science as well as encourage reliable and reproducible discoveries, elevating the endeavor to a branch of the scientific method. Data science may eventually develop as a set of discipline-adapted discovery techniques and practices, perhaps including a cross-disciplinary core. Data science is benefitting from close association with industry as

computer science did at its inception, for example, IBM’s creation of the Watson Scientific Computing Laboratory at Columbia University in 1945.¹⁴ Analysis of consumer data by Google, Facebook, and Amazon is generating prominent successes in image identification and voice transcription among other areas. Opportunities for industry employment and workforce development create an attractive feature of data science at the institutional level.

Elevating the practice of data science to a science. The Data Science Life Cycle framework is an essential conceptualization in the development of data science as a science. A recent National Academies of Sciences, Engineering, and Medicine consensus report on “Reproducibility and Replication in Science” spotlights the need to better develop scientific underpinnings for computationally and data-enabled research investigations²¹ and a March

An example mapping from some routinely offered courses to the steps of the Data Science Life Cycle.

The table is not intended as a complete and comprehensive description of all skills required to be an effective data scientist, but an illustration of how current courses could be incorporated into a data science training curriculum, within which students may pursue pathways of interest. Possible new courses to be developed can be gleaned from such a presentation. Some courses are listed in more than one step to illustrate various ways they might be included in curriculum design.

Data Science Life Cycle Step	Possible (Existing) Courses
Experimental design	<ul style="list-style-type: none"> ▶ Introduction to Probability ▶ Introduction to Statistics ▶ Design of Experiments (including Human Subjects and Informed Consent)
Obtaining data	<ul style="list-style-type: none"> ▶ Experimental Methodology ▶ Introduction to Databases ▶ Introduction to SQL, noSQL ▶ Sensor Integration and Control
Data exploration	<ul style="list-style-type: none"> ▶ Introduction to R ▶ Introduction to python ▶ Graphics and Data Visualization ▶ Introduction to Statistics
Databases and data structures including cleaning/organizing	<ul style="list-style-type: none"> ▶ Introduction to Database Systems ▶ Introduction to SQL, noSQL ▶ Natural Language Processing (NLP)
Software engineering	<ul style="list-style-type: none"> ▶ Python, R, C, C++, Julia ▶ Distributed Systems, MapReduce ▶ Software Testing
Feature selection	<ul style="list-style-type: none"> ▶ Statistical Learning ▶ Domain-specific courses, for example, Bioinformatics for Transcriptomics; Brain Imaging in Cognitive Neuroscience Research
Model estimation	<ul style="list-style-type: none"> ▶ Mathematics (Probability, Linear Algebra, Calculus, Real Analysis) ▶ Applied Statistics ▶ Machine Learning ▶ Data Mining ▶ Deep Learning ▶ Scalable Algorithms ▶ Statistical Decision Theory
Simulation and cross-validation	<ul style="list-style-type: none"> ▶ Fundamentals of Numerical Methods ▶ Introduction to Computer Modeling and Simulation ▶ Statistical Learning
Visualization	<ul style="list-style-type: none"> ▶ Information Visualization ▶ Scientific Visualization and Graphics ▶ [Domain specific courses such as Learning ArcGIS; Spatial Data Visualization]
Publication/Archiving	<ul style="list-style-type: none"> ▶ Introduction to Information ▶ Data Archiving and FAIR Data ▶ Scientific Report Writing ▶ Research Data Management ▶ Open Access and Scholarly Communication ▶ Digital Libraries and Preservation
Overarching topics	<ul style="list-style-type: none"> ▶ Ethics for Scientists ▶ Data Privacy ▶ National and International Regulatory Trends in Data Protection

2019 National Academy of Sciences Colloquium entitled “The Science of Deep Learning” aimed to bring scientific foundations to the fore of the deep learning research agenda.¹⁹ The discussion regarding the scientific underpinnings of data analysis began in 1962, when John Tukey presented three criteria a discipline ought to meet in order to be considered a science:³⁰

1. Intellectual content.
2. Organization into an understandable form.
3. Reliance upon the test of experience as the ultimate standard of validity.

If one accepts these criteria, the Data Science Life Cycle can be leveraged to demonstrate intellectual content, promote its organization (see Figure 2), and incorporate external tests of the validity of findings. On this last point, the structure of the Data Science Life Cycle builds in reproducibility, reuse, and verification of results with its embedded notion that artifacts supporting the claims (such as data, code, workflow information) be made available as part of the publication (life cycle) process. Research on platforms and infrastructure for data science facilitates Tukey’s second criterion by advancing organizational topics such as artifact meta data; containerization, packaging and dissemination standards; and community expectations regarding FAIR (findability, accessibility, interoperability, and reusability), archiving, and persistence of the artifacts produced by data science. These efforts also help enable comparisons of data science pipelines to increase understanding of any differences in outcomes of “tests of experience.”²⁹ The Data Science Life Cycle exposes these topics as areas for research within the discipline of data science.² Several conferences and journals have begun to require artifact availability and infrastructure projects are emerging to support reproducibility across the data science discovery pipeline.³ Considering these issues through a Data Science Life Cycle gives a frame for their inclusion as research areas integral to the discipline of Data Science. Data science without a unifying framework risks being a set of disparate computational activities in various scientific domains, rather than a coherent field of inquiry producing reliable reproducible knowledge.

Conclusion

Without a flexible yet unified overarching framework we risk missing opportunities for discovering and addressing research issues within data science and training students in effective scientific methodologies for reliable and transparent data-enabled discovery. Data science brings new research topics, for example, computational reproducibility; ethics in data science; cyberinfrastructure and tools for data science. Without the Data Science Life Cycle approach, we risk an implementation of data science that too closely hews to a view that reflects the perspective of a particular discipline and could miss opportunities to share knowledge on data science research and teaching broadly across disciplines. In addition, a Data Science Life Cycle approach can give university leadership a framework to leverage their existing resources on campus as they strategize support for a cross-disciplinary data science curriculum and research agenda. The life cycle approach allows data science research and curriculum efforts to support the development of a scientific discipline, enabling progress toward fulfilling Tukey’s three criteria for a science. **C**

References

1. Berman, F. et al. Realizing the potential of data science. *Commun. ACM* 61, 4, (Apr. 2018), 67–72; <https://cacm.acm.org/magazines/2018/4/226372-realizing-the-potential-of-data-science/fulltext>
2. Bernau, C. et al. Cross-study validation for the assessment of prediction algorithms. *Bioinformatics* 30, 12; <https://academic.oup.com/bioinformatics/article/30/12/1105/388184>
3. Brinckman, A. et al. Computing environments for reproducibility: Capturing the ‘whole tale’. *Future Generation Computer System* 94, 854–867; <https://www.sciencedirect.com/science/article/pii/S0167739X17310695>
4. Collberg C. and Proebsting, T.A. Repeatability in computer systems research. *Commun. ACM* 59, 3 (Mar. 2016), 62–69; <https://doi.org/10.1145/2812803>
5. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition*. 2009; <https://ieeexplore.ieee.org/document/5206848>
6. Dhar, V. Data science and prediction. *Commun. ACM* 56, 12 (Dec. 2013); 64–73; <https://doi.org/10.1145/2500499>
7. Donoho, D.L. 50 years of data science. *J. Computational and Graphical Statistics* 26, 4 (2017); <https://www.tandfonline.com/doi/abs/10.1080/10618600.2017.1384734>
8. Donoho, D.L., Maleki, A., Ur Rahman, I., Shahram, M. and Stodden, V. Reproducible research in computational harmonic analysis. *Computing in Science & Engineering* 11, 1, (Jan.-Feb 2009).
9. Donoho, D.L. and Stodden, V. 2015. Reproducible research in the mathematical sciences. J. Higham, ed. *The Princeton Companion to Applied Mathematics*.
10. Golub, T.R. et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 5439 (1999), 531–537.
11. Guyon, I. et al. Gene selection for cancer classification using support vector machines. *Machine Learning* 46 (Jan. 2002); <https://doi.org/10.1023/A:1012487302797>

12. Hales, T. Mathematics in the age of the Turing machine. *Turing’s Legacy Developments from Turing’s Ideas in Logic*. R. Downey, ed., 2014; <https://www.cambridge.org/core/books/turings-legacy/mathematics-in-the-age-of-the-turing-machine/376464C81D16F932EEFB2A2924D2F4>
13. Hoover, H. Quantitative analysis and literary studies. *A Companion to Digital Literary Studies*. S. Schreibman and R. Siemens, eds. Blackwell, Oxford, U.K., 2008.
14. IBM. The Origins of Computer Science; <https://www.ibm.com/ibm/history/ibm100/us/en/icons/compsci/>
15. Ivie, P. and Thain, D. Reproducibility in scientific computing. *ACM Comput. Surv.* 51, 3 (2018), Art. 63; <https://doi.org/10.1145/3186266>
16. Krizhevsky, A., Sutskever, I. and Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25, 2012. F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, eds; <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
17. Lazer, D. et al. Computational social science. *Science* 323, 5915 (2009); <http://science.sciencemag.org/content/323/5915/721>
18. Manyika, J. et al. Big Data: The Next Frontier for Innovation, Competition and Productivity. McKinsey Global Institute, 2011; <http://www.mckinsey.com/business-functions/business-technology/our-insights/big-data-the-next-frontier-for-innovation>
19. NAS Sackler Colloquium. The Science of Deep Learning, 2019; <http://www.cvent.com/events/the-science-of-deep-learning/event-summary-a96a8734fa841ea8d5439e081b50f54.aspx>
20. National Academies of Sciences, Engineering, and Medicine. *Data Science for Undergraduates: Opportunities and Options*. The National Academies Press, Washington, D.C.; <https://doi.org/10.17226/25104>.
21. National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. The National Academies Press, Washington, D.C., 2019; <https://doi.org/10.17226/25303>
22. Steering Committee on Computational Physics. *Computation as a Tool for Discovery in Physics*. Report to the National Science Foundation, 2002; <https://www.nsf.gov/pubs/2002/nsf02176/nsf02176.pdf>
23. Ouzounis, C.A. Rise and demise of bioinformatics? Promise and progress. *PLoS Comput Biol* 8, 4 (2012), e1002487; <https://doi.org/10.1371/journal.pcbi.1002487>
24. Saltz, J.S., Dewar, N.I., Heckman and R. Key concepts for a data science ethics curriculum. In *Proceedings of the 49th ACM Technical Symp. Computer Science Education*. ACM, New York, NY, 952–957; <https://doi.org/10.1145/3159450.3159483>
25. Siewert, S. Big data in the cloud: Data velocity, volume, variety, veracity. *IBM Developer*, July 9, 2013; <https://www.ibm.com/developerworks/library/bd-bigdatacloud/index.html>
26. Stodden, V. The legal framework for reproducible research in the sciences: Licensing and copyright. *Computing in Science and Engineering* 11, 1 (2009), 35–40.
27. Stodden, V., Guo, P. and Ma, Z. Toward reproducible computational research: An empirical analysis of data and code policy adoption by journals. *PLoS ONE* 8, 6 (2013), e67111; <https://doi.org/10.1371/journal.pone.0067111>
28. Stodden, V., McNutt, M., Bailey, D.H., Deelman, E., Gil, Y., Hanson, B., Heroux, M.A., Ioannidis, J.P.A., Tauber, M. Enhancing Reproducibility for Computational Methods. *Science* 354, 6317 (Dec. 9, 2016).
29. Stodden, V., Wu, X. and Sochat, V. AIM: An abstraction for improving machine learning prediction. In *Proceedings of the IEEE Data Science Workshop*. (Lausanne, Switzerland, 2018), 1–5.
30. Tukey, J.W. The Future of Data Analysis. *Ann. Math. Statist.* 33, 1 (1962), 1–67.

Victoria Stodden (vcs@stodden.net) is a statistician and associate professor at the University of Illinois at Urbana-Champaign, IL, USA.

This material is based upon work supported by National Science Foundation Award #1941443.

Copyright held by author/owner.

Google's TPU supercomputers train deep neural networks 50x faster than general-purpose supercomputers running a high-performance computing benchmark.

BY NORMAN P. JOUPPI, DOE HYUN YOON, GEORGE KURIAN, SHENG LI, NISHANT PATIL, JAMES LAUDON, CLIFF YOUNG, AND DAVID PATTERSON

A Domain-Specific Supercomputer for Training Deep Neural Networks

THE RECENT SUCCESS of deep neural networks (DNNs) has inspired a resurgence in domain specific architectures (DSAs) to run them, partially as a result of the deceleration of microprocessor performance improvement due to the slowing of Moore's Law.¹⁷ DNNs have two phases: *training*, which constructs

accurate models, and *inference*, which serves those models. Google's Tensor Processing Unit (TPU) offered 50x improvement in performance per watt over conventional architectures for inference.^{19,20} We naturally asked whether a successor could do the same for training. This article explores how Google built the first production DSA for the much harder training problem, first deployed in 2017.

Computer architects try to create designs that maximize performance on a set of benchmarks while minimizing

costs, such as fabrication or operating cost.¹⁶ In the case of DSAs like Google's TPUs, many of the principles and experiences from decades of building general-purpose CPUs change or do not apply. For example, here are features of the inference TPU (TPUv1) and the training TPU (TPUv2) share but are uncommon in CPUs:

- ▶ 1–2 large cores versus 32–64 small cores in server CPUs.
- ▶ The computational heavy lifting is handled by two-dimensional (2D)

128x128- or 256x256-element systolic arrays of multipliers per core, versus either a few scalar multipliers or SIMD (one-dimensional, 16–32-element) multipliers per core in CPUs.


- Using narrower data (8–16 bits) to improve efficiency of computation and memory versus 32–64 bits in CPUs.

- Dropping general-purpose features irrelevant for DNNs but critical for CPUs such as caches and branch predictors.


The most effective DNN training is supervised learning, where we start with a huge (sometimes billion-example) training dataset of known-correct (`input`, `result`) pairs. Pairs might be an image and what it depicts or an audio waveform and the phoneme it represents. We also start with a neural network model, which transforms the input into the result through an intensive calculation of weights (also called parameters); the weights are random initially. Models are typically defined as a graph of layers, where a layer contains a linear algebra part (often a matrix multiplication or convolution using the weights) followed by a nonlinear activation function (often a scalar function, applied elementwise; we call the results *activations*). Training “learns” weights that raise the likelihood of correctly mapping from input to result.

For some kinds of input data, an embedding at the start of the model transforms from sparse representations into a dense representation suitable for linear algebra; embeddings also contain weights.^{27,29} Embeddings might use vectors where features can be represented by notions of distance between vectors. Embeddings involve table lookups, link traversal, and variable length data fields, so they are irregular and memory intensive.

How do we get from random initial weights to trained weights? Current best practices use variants of *stochastic gradient descent* (SGD).³¹ SGD consists of many iterations of three steps: forward propagation, backpropagation, and weight update. Forward propagation takes a randomly chosen training example, applies its inputs to the model, and runs the calculation through the layers to produce a result (which with the random initial weights, is garbage the first time). Forward propagation is functionally similar to DNN inference,



DNN (Deep Neural Network) wisdom is that bigger machines lead to bigger breakthroughs.



and if we were building an inference accelerator, we could stop there. For training, this is less than a third of the story. SGD next measures the difference or error between the model’s result and the known good result from the training set using a loss function. Then back-propagation runs the model in reverse, layer-by-layer, to produce a set of error/loss values for each layer’s output. These losses measure the deviation from the desired output. Last, weight update combines the input of each layer with the loss value to calculate a set of deltas—changes to weights—which, when added to the weights, would have resulted in nearly zero loss. Updates can have small magnitude. Shrinking further, updates are scaled down by the learning rate to keep SGD numerically stable. Moreover, a suite of algorithmic refinements—including momentum,³⁰ batch normalization,¹⁸ and optimizers such as Adaptive Gradient (AdaGrad)¹⁴—require their own state and alter the SGD algorithm to reduce the number of steps to achieve desired accuracy.

Each SGD step makes a tiny adjustment to the weights that improves the model with respect to a single (`input`, `result`) pair. Each pass through the entire dataset is an *epoch*; DNNs typically take tens to hundreds of epochs to train. SGD gradually transforms the random initial weights into a trained model, sometimes capable of superhuman accuracy.

Given this background, we can compare inference and training. Both share some computational elements including matrix multiplications, convolutions, and activation functions, so inference and training DSAs might have similar functional units. Key architectural aspects where the requirements differ include:

- *Harder parallelization*: Each inference is independent, so a simple cluster of servers with DSA chips can scale up inference. A training run iterates over millions of examples, coordinating across parallel resources because it must produce a single consistent set of weights for the model. The number of examples processed in parallel, and the time to evaluate that multiple-example *minibatch*—often shortened to *batch*—directly affect total end-to-end training time. A *step* is the computation to process one minibatch.

► *More computation:* Back-propagation requires derivatives for every computation in a model. It includes activation functions (some of which are transcendental), and multiplication by transposed weight matrices.

► *More memory:* Weight update accesses intermediate values from forward and back propagation, vastly upping storage requirements; temporary storage can be 10x weight storage. For inference, a small activation working set can usually be kept on chip.

► *More programmability:* Training algorithms and models are continually changing, so a machine restricted to current best-practice algorithms during design could rapidly become obsolete.

► *Wider data:* Quantized arithmetic—8-bit integer instead of 32-bit floating point (FP)—can work for inference like in TPUv1 but reduced-precision training is an active research area.^{21,25} The challenge is sufficiently capturing the SGD sum of many small weight updates to preserve the accuracy of using 32-bit FP arithmetic to train models.

After explaining the TPUv2 architecture, we describe the domain specific language (TensorFlow) and compiler (XLA) for TPUv2 and compare the architecture and technology choices for the TPUv2 versus a GPU, the most popular computer for DNN training. Later, we compare performance per chip and full supercomputers of TPUs and GPUs using production applications and the MLPerf benchmarks.

Designing a Domain-Specific Supercomputer

In 2014, when the TPUv2 project began, the landscape for high-performance machine learning computation was very different from today. Training took place on clusters of CPUs. State-of-the-art parallel training used asynchronous SGD,¹² in part to tolerate tail latencies in shared clusters. Parallel training also divided CPUs into a bipartite graph of workers (running the SGD loop) and parameter servers (hosting weights and adding updates to them).

The DNN training computation appetite appeared unlimited. (Indeed, the computation requirements for the largest training runs grew 10x annually from 2012 to 2018.²) Thus, in 2014 we chose to build a DSA supercomputer in-

stead of clustering CPU hosts with DSA chips. The first reason is that training time is huge. Table 1 shows that one TPUv2 chip would take two to 16 months to train a single Google production application, so a typical application might want to use hundreds of chips. Second, DNN wisdom is that bigger datasets plus bigger machines lead to bigger breakthroughs. Moreover, results like AutoML use 50x more computation to find DNN models that achieve higher accuracy scores than the best models of human DNN experts.⁴²

Designing a DSA supercomputer interconnect. The critical architecture feature of a modern supercomputer is how its chips communicate: what is the speed of a link; what is the interconnect topology; does it have centralized versus distributed switches; and so on. This choice is much easier for a DSA supercomputer, as the communication patterns are limited and known. For training, most traffic is an all-reduce over weight updates from all nodes of the machine.

If we distribute switch functionality into each chip rather than as a stand-alone unit, the all-reduction can be built in a dimension-balanced, bandwidth-optimal way for a 2D torus topol-

» key insights

- **With the slowing of Moore's Law, ML breakthroughs require innovation in computer architecture.**
- **The increasing importance and appetite for ML training justifies its own custom supercomputer.**
- **The co-design of an ML-specific programming system (TensorFlow), compiler (XLA), architecture (TPU), floating-point arithmetic (Brain float16), interconnect (ICI), and chip (TPUv2/v3) let production ML applications scale at 96%–99% of perfect linear speedup and 10x gains in performance/Watt over the most efficient general-purpose supercomputers.**

ogy (see Figure 1). An on-device switch provides virtual-circuit, deadlock-free routing. To enable a 2D torus, the chip has four custom Inter-Core Interconnect (ICI) links, each running at 496Gbits/s per direction in TPUv2. ICI enables direct connections between chips to form a supercomputer using only 13% of each chip (see Figure 3). Direct links simplify rack-level deployment, but in a multi-rack system the racks must be adjacent.

One measure of an interconnect is its *bisection bandwidth*—the bandwidth

Table 1. Days to train production programs on one TPUv2 chip.

MLP0	MLP1	CNNO	CNN1	RNNO	RNN1
475	117	63	115	77	147

Figure 1. A 2D-torus topology. TPUv2 uses a 16x16 2D torus.

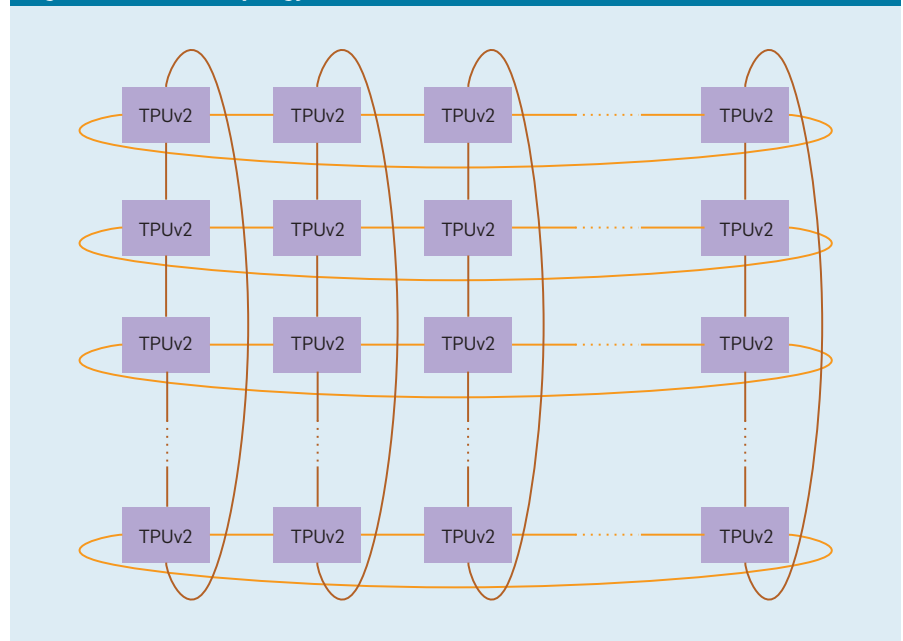
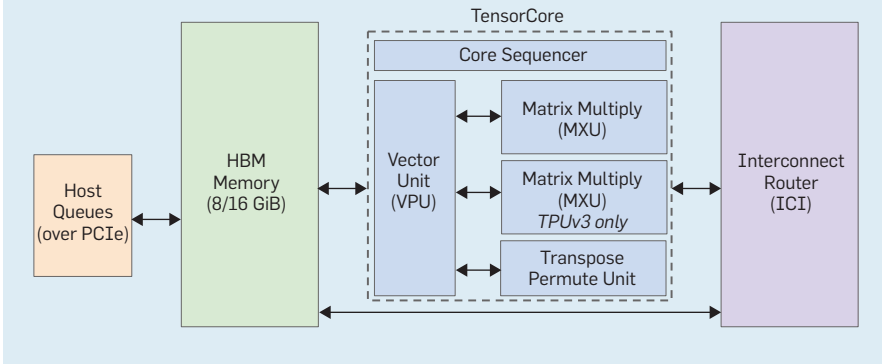


Table 2. Batch sizes for the three regions of Shallue.³² LM1B, Fashion MNIST, and Imagenet are standard DNN datasets.

Model	Perfect	Diminishing	Maximum
Transformer on LM1B	≤256	256–4096	≥4096
Simple CNN on Fashion MNIST	≤512	512–2048	≥2048
ResNet-50 on Imagenet	≤8192	8192–65536	≥65536

Figure 2. Block diagram of a TensorCore (our internal development name for a TPU core, and not related to the Tensor Cores of NVIDIA GPUs).



available between two halves of a network of the worst-case split. The TPUv2 supercomputer uses a 16x16 2D torus (256 chips), which is 32 links x 496Gbits/s = 15.9Terabits/s of bisection bandwidth. As a comparison, a separate Infiniband network (used in CPU clusters) that connected 64 hosts (each with, say, four DSA chips) has 64 ports using “only” 100Gbit/s links and a bisection bandwidth of at most 6.4Terabits/s. Our TPUv2 supercomputer provides 2.5x the bisection bandwidth over conventional cluster switches while skipping the cost of the Infiniband network cards, Infiniband switch, and the communication delays of going through the CPU hosts of clusters.

Fortuitously, building a fast interconnect inspired algorithmic advances. With dedicated hardware, and sharding the examples of a minibatch over nodes of the machine, there is little tail latency, and synchronous parallel training becomes possible. Internal studies⁵ suggested that synchronous training could beat asynchronous SGD with equivalent resources. Asynchronous training introduces heterogeneity plus parameter servers that eventually limit parallelization, as the weights get sharded and the bandwidth from parameter servers to workers becomes a bottleneck. Synchronous training eliminated the parameter servers allowing

peer-to-peer among workers, using the all-reduce to ensure workers begin and end each parallel step with consistent copies of weights.

Synchronous training has two phases in the critical path—a compute phase and a communication phase that reconciles the weights across learners. The slowest learners and slowest messages through the network limit performance of such a synchronous system. Since the communication phase is in the critical path, a fast interconnect that quickly reconciles weights across learners with well-controlled tail latencies is critical for fast training. The ICI network is key to the excellent TPU supercomputer scaling results; later we show 96%–99% of perfect linear scaleup.

Designing a DSA supercomputer node. The TPUv2 node of the supercomputer followed the main ideas of TPUv1: A large two-dimensional matrix multiply unit (MXU) using a systolic array to reduce area and energy plus large, software-controlled on-chip memories instead of caches. The large MXUs of the TPUs rely on large batch sizes, which amortize memory accesses for weights—performance often increases when memory traffic reduces.

Shallue et al.³² examined the effect of increasing batch size on training time, and found three regions for all

models (as seen in Table 2):

1. *Perfect scaling region:* Each doubling of batch size halves the number of training steps.

2. *Diminishing returns region:* Increasing batch size still reduces the number of steps, but more slowly.

3. *Maximum data parallelism region:* Increasing batch size provides no benefits whatsoever.

Such scaling while preserving accuracy required tuning the learning rate, batch size, and other hyperparameters.

Fortunately for TPUs, these recent results show that batch sizes of 256–8,192 scale perfectly without losing accuracy, which makes large MXUs an attractive option for high performance.

Unlike TPUv1, TPUv2 uses two cores per chip. Global wires on a chip don’t scale with shrinking feature size, so their relative delay increases. Given that training can use many processors, two smaller TensorCores per chip prevented the excessive latencies of a single large full-chip core. We stopped at two because it is easier to efficiently generate programs for two brawny cores per chip than numerous wimpy cores.

Figure 2 shows the six major blocks of a TensorCore and Figure 3 shows their placement in the TPUv2 chip:

1. *Inter-Core Interconnect (ICI).* Explained earlier.

2. *High Bandwidth Memory (HBM).* TPUv1 was memory bound for most of its applications.²⁰ We solved its memory bottleneck by using High Bandwidth Memory (HBM) DRAM in TPUv2. It offers 20 times the bandwidth of TPUv1 by using an interposer substrate that connects the TPUv2 chip via thirty-two 128-bit buses to four short stacks of DRAM chips. Conventional servers support many more DRAM chips, but at a much lower bandwidth of at most eight 64-bit busses.

3. The *Core Sequencer* fetches VLIW (*Very Long Instruction Word*) instructions from the core’s on-chip, software-managed Instruction Memory (Imem), executes scalar operations using a 4K 32-bit scalar data memory (Smem) and 32 32-bit scalar registers (Sregs), and forwards vector instructions to the VPU. The 322-bit VLIW instruction can launch eight operations: two scalar, two vector ALU, vector load and store, and a pair of slots that queue data to and from the matrix

multiply and transpose units. The XLA compiler schedules loading Imem via independent overlays of code, as unlike conventional CPUs, there is no instruction cache.

4. The *Vector Processing Unit (VPU)* performs vector operations using a large on-chip *vector memory (Vmem)* with 32K 128x32-bit elements (16MiB), and 32 2D *vector registers (Vregs)* that each contain 128 x 8 32-bit elements (4 KiB). The VPU streams data to and from the MXU through decoupling FIFOs. The VPU collects and distributes data to Vmem via *data-level parallelism* (2D matrix and vector functional units) and *instruction-level parallelism* (8 operations per instruction).

Your beautiful DSA can fail if best-practice algorithms change, rendering

it prematurely obsolete. We handled such a crisis in 2015 during our design in supporting batch normalization.¹⁸ Briefly, *batch normalization* subtracts out the mean and divides by the standard deviation of a batch, making the values look like samples from the normal distribution. In practice, it both improves prediction accuracy and reduces time-to-train up to 14x! Batch normalization emerged early in 2015, and the results made it a must-do for us. We divided it into vector additions and multiplications over the batch, plus one inverse-square-root calculation. However, the vector operation count was high. We thus added a second SIMD dimension to our vector unit, making its registers and ALUs 128x8 (rather than just 1D 128-wide) and add-

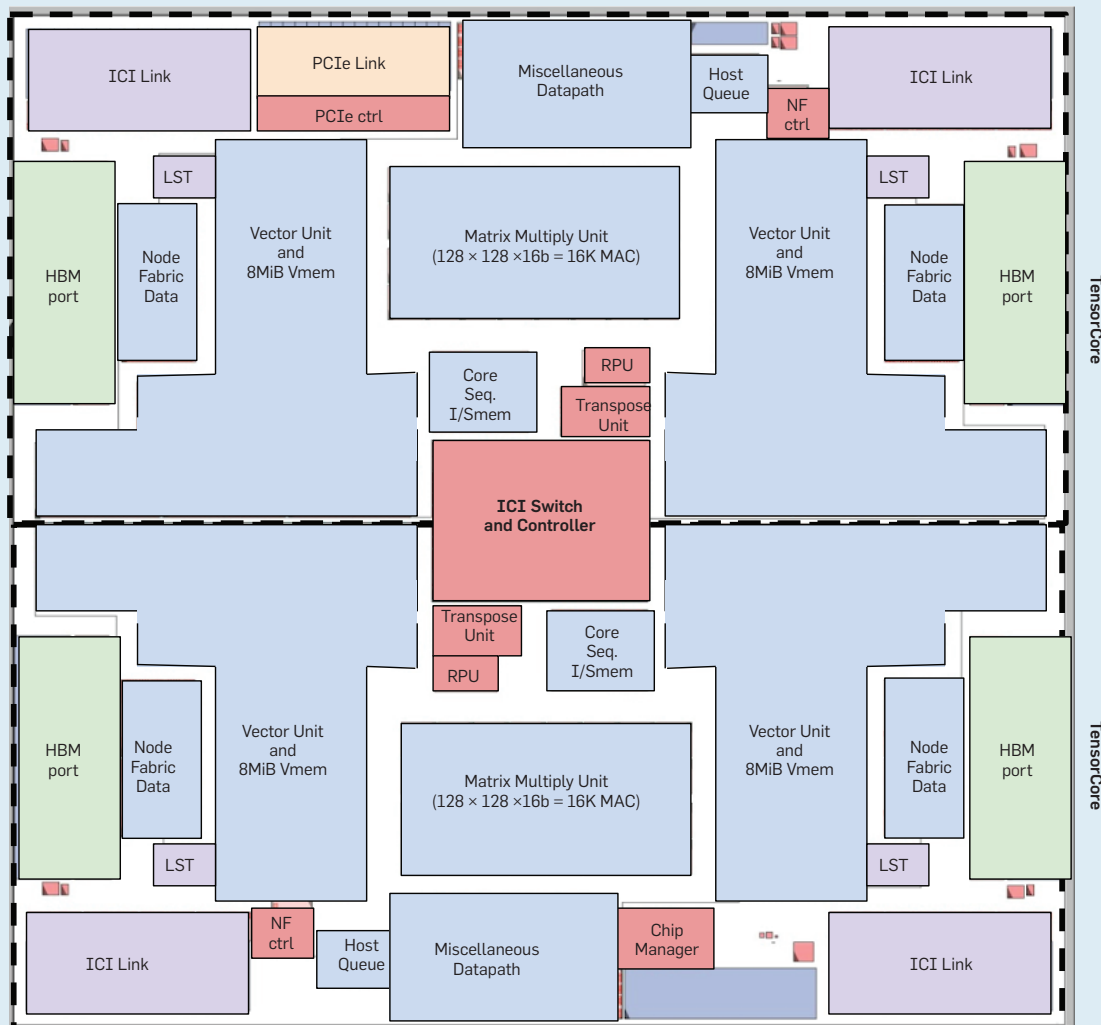
ing an inverse square root operation to the transcendental unit.

5. The MXU produces 32-bit FP products from 16-bit FP inputs that accumulate in 32 bits. All other computations are in 32-bit FP except for results going directly to an MXU input, which are converted to 16-bit FP.

The MXUs are large, but we reduced their size from 256x256 in TPUv1 to 128x128 and have multiple MXUs per chip. The bandwidth required to feed and obtain results from an MXU is proportional to its perimeter, while the computation it provides is proportional to its area. Larger arrays provide more compute per byte of interface bandwidth, but larger arrays can be inefficient. Simulations show that convolutional model utilization of

Figure 3. TPUv2 chip floor plan.

It has two TensorCores: Node fabric data and NF controller move on-chip data.



four 128x128 MXUs is 37%–48%, which is 1.6x of a single 256x256 MXU (22%–30%) yet take about the same die area. The reason is that some convolutions are naturally smaller than 256x256, so sections of the MXU would be idle. Sixteen 64x64 MXUs would have a little higher utilization (38%–52%) but would need more area. The reason is the MXU area is determined either by the logic for the multipliers or by the

wires on its perimeter for the inputs, outputs, and control. In our technology, for 128x128 and larger the MXU’s area is limited by the multipliers but area for 64x64 and smaller MXUs is limited by the I/O and control wires.

6. The *Transpose Reduction Permute Unit* does 128x128 matrix transposes, reductions, and permutations of the VPU lanes.

Alternative DSA supercomputer

node designs. The TPUv1 article evaluated hypothetical alternatives that examined the changes in performance while varying the MXU size, the clock rate, and the memory bandwidth.²⁰ We need not hypothesize here, as we implemented and deployed two versions of the training architecture: TPUv2 and TPUv3. TPUv3 has $\approx 1.35x$ the clock rate, ICI bandwidth, and memory bandwidth plus twice the number of MXUs, so peak performance rises 2.7x. Liquid cools the chip to allow 1.6x more power. We also expanded the TPUv3 supercomputer to 1024 chips (see Figure 4). Table 3 lists key features of the three TPU generations along with a contemporary GPU (NVIDIA Volta) that we’ll compare to below.

The TPUv3 die size is only 6% larger than TPUv2 in the same technology despite having twice as many MXUs per TensorCore simply because the engineers had a better idea beforehand of the layout challenges of the major blocks in TPUv2, which led to a more efficient floor plan for TPUv3.

Designing DSA supercomputer arithmetic. Peak performance is $\geq 8x$ higher when using 16-bit FP instead of 32-bit FP for matrix multiply (see Table 3), so it’s vital to use 16-bit to get highest performance. While we could have built an MXU using standard IEEE fp16 and fp32 floating point formats (see Figure 5), we first checked the accuracy of 16-bit operations for DNNs. We found that:

- ▶ Matrix multiplication outputs and internal sums must remain in fp32.
- ▶ The 5-bit exponent of fp16 matrix multiplication inputs leads to failure

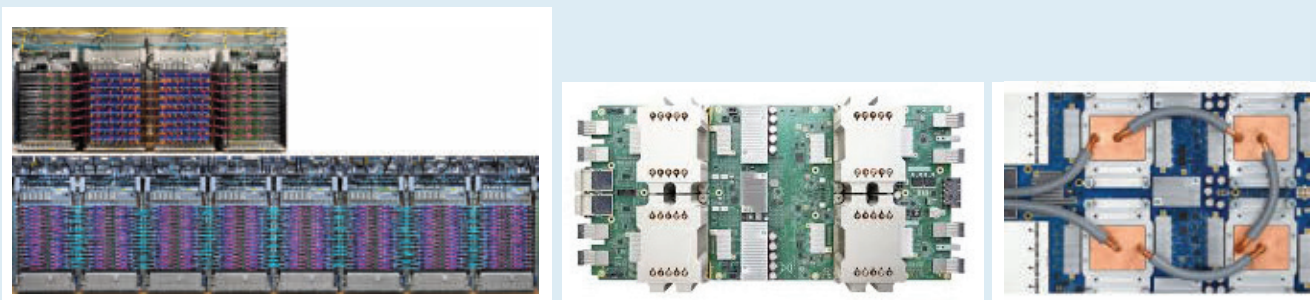
Table 3. Key processor features.

We cannot reveal technology details of our chip partner. Although it is in a larger, older technology, the TPUv2 die size is less than 3/4s of the GPU. TPUv3 is 6% larger in that same technology. TDP stands for Thermal Design Power. The Volta has 80 symmetric multiprocessors.

Feature	TPUv1	TPUv2	TPUv3	Volta
Peak TeraFLOPS/Chip	92 (8b int)	46 (16b) 3 (32b)	123 (16b) 4 (32b)	125 (16b) 16 (32b)
Network links x Gbits/s/Chip	--	4 x 496	4 x 656	6 x 200
Max chips/supercomputer	--	256	1024	Varies
Peak PetaFLOPS/supercomputer	--	11.8	126	Varies
Bisection Terabits/supercomputer	--	15.9	42.0	Varies
Clock Rate (MHz)	700	700	940	1530
TDP (Watts)/Chip	75	280	450	450
TDP (Kwatts)/supercomputer	--	124	594	Varies
Die Size (mm ²)	<331	<611	<648	815
Chip Technology	28nm	>12nm	>12nm	12nm
Memory size (on/off-chip)	28MiB/8GiB	32MiB/16GiB	32MiB/32GiB	36MiB/32GiB
Memory GB/s/Chip	34	700	900	900
MXUs/Core, MXU Size	1 256x256	1 128x128	2 128x128	8 4x4
Cores/Chip	1	2	2	80
Chips/CPU Host	4	4	8	8 or 16

Figure 4. A TPUv2 supercomputer has up to 256 chips and is 18-ft. long (top).

A TPUv3 supercomputer consisting of up to 1,024 chips (below) is about 7-ft. tall and 36-ft. long. A TPUv2 board (center) holds four air-cooled chips and a TPUv3 board (right) also has four chips but uses liquid cooling.



of computations that go outside its narrow range, which the 8-bit exponent of fp32 avoids.

► Reducing the matrix multiplication input mantissa size from fp32’s 23 bits to 7 bits did not hurt accuracy.

The resulting *brain floating format* (bf16) in Figure 5 keeps the same 8-bit exponent as fp32. Given the same exponent size, there is no danger in losing the small update values due to FP underflow of a smaller exponent, so all programs in this article used bf16 on TPUs without much difficulty. Beyond our experience that it works for training production applications, a recent Intel study corroborated its benefits.²¹ However, fp16 requires adjustments to training software (*loss scaling*) to deliver convergence and efficiency. It preserves the effect from small gradients by scaling losses to fit the smaller exponents of fp16.²⁶

As the size of an FP multiplier scales with the square of the *mantissa* width, the bf16 multiplier is *half* the size and energy of a fp16 multiplier: $8^2 / 16^2 \approx 0.5$ (accounting for the implicit leading mantissa bit). Bf16 delivers a rare combination: reducing hardware and energy while simplifying software by making loss scaling unnecessary. Thus, ARM and Intel have revealed future chips with bf16.

Designing a DSA Supercomputer Compiler

The next step was getting software for our hardware. To program CPUs and GPUs for machine learning, a framework such as *TensorFlow* (TF)¹ specifies the model and data operations machine-independently. TF is a domain-specific library built on Python. NVIDIA GPU-dependent work is supported by a combination of the CUDA language, the CuBLAS and CuDNN libraries, and the TensorRT system. TPUv2/v3s also use TF, with the new system XLA (for accelerated linear algebra) handling the TPU-dependent mapping. XLA also targets CPUs and GPUs. Like many systems that map

from domain-specific languages to code, XLA integrates a high-level library and a compiler. A TF front end generates code in an intermediate representation for XLA.

It would seem it should be more difficult to get great performance in a programming system based on Python like TF. However, ML frameworks offer both a higher level of expressiveness and the potential for much better optimization information than lower-level languages like C++. TF programs are graphs of operations, where multi-dimensional array operations are first-class citizens:

- They operate on multi-dimensional arrays explicitly, rather than implicitly via nested loops as in C++.
- They use explicit, analyzable, and bounded data access patterns versus arbitrary access patterns like C++.
- They have known memory aliasing behavior, unlike C++.

These three factors allow the XLA compiler to safely and correctly transform programs in ways that traditional compilers rarely attain.

XLA does whole-program analysis and optimization. With 2D vector registers and compute units in TPUv2/v3, the layout of data in both compute units and memory is critical to performance, perhaps more than for a vector or SIMD processor. Building efficient code for vector machines, with 1D memory and compute units, is well understood. For the MXU, two 2D

inputs interact to produce a 2D output. Each operand has a memory layout, which gets transformed into a layout in 2D registers, which in turn must be fed at the exact moment to meet systolic array timing in the MXU. (A systolic array reduces register accesses by choreographing data flowing from different directions to regularly arrive at cross points that combine them.) Depending on layout choices, the 2D registers dimensions of 128 and 8 might not be filled, lowering ALU and memory utilization. Moreover, lacking caches, XLA manages all memory transfers, including code overlays and DMA pushes to remote nodes over ICI.

XLA exploits the huge parallelism that an input TF dataflow graph represents. Beyond the parallelism of operations (“ops”) in a graph, each op can comprise millions of multiplications and additions on data tensors of millions of elements. XLA maps this abundant parallelism across hundreds of chips in a supercomputer, a few cores per chip, multiple units per core, and thousands of multipliers and adders inside each functional unit. The domain-specific TF language and XLA representation allow precise reasoning about memory use at every point in the program. There are no “aliasing” issues where the compiler must determine whether two pointers might address the same memory—every piece of memory cor-

Figure 5. IEEE FP and Brain float formats.

All formats have an implicit leading mantissa bit in normal operation.

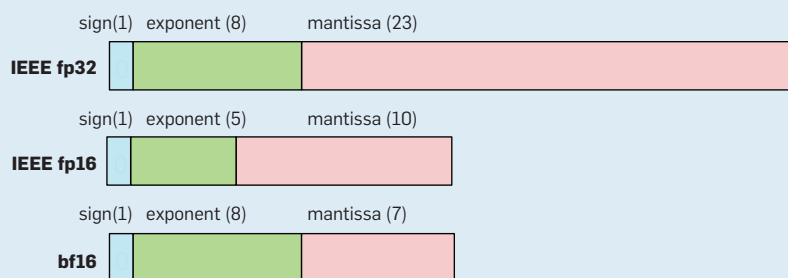


Table 4. XLA speed up on TPUv2 with fusion versus without fusion.

MLP		CNN		RNN		SSD	NMT	Mask R-CNN	Transformer	Res Net-50
0	1	0	1	0	1					
1.8	2.0	2.2	4.8	2.4	1.8	2.4	3.0	2.0	2.0	6.3

responds to a known program variable or temporary. The XLA compiler is free to slice, tile, and lay out memory and operations to best use the on-chip memory bandwidth and to reduce the memory footprint on chip or off chip.

TPUs use a VLIW architecture to express instruction-level parallelism to the many compute units of a TensorCore. XLA uses standard VLIW compilation techniques including loop unrolling, instruction scheduling, and software pipelining to keep all compute units busy and to simultaneously move data through the memory hierarchy to feed them.

Given a memory layout of data, *operator fusion* can reduce memory use and boost performance. Fusion is a traditional compiler optimization—but applied now to 2D data—that combines ops to reduce memory traffic compared to executing operators sequentially. For example, fusing a matrix multiplication with a following activation function skips writing and reading the intermediate products from memory. Table 4 shows the speedup from the fusion optimization on 2D data is from 1.8 to 6.3.

The TF intermediate form for XLA has thousands of ops. The number of ops increases when programmers cannot combine existing ops if composition is inefficient. Alas, expanding the number of ops is an engineering challenge, since software libraries need to be developed for CPUs, GPUs, and TPUs. The hope was that the XLA compiler could synthesize these thou-

sands of ops from a smaller set of primitive ops.

The XLA team needed only 96 ops as the compiler’s target to reduce work for the library/compiler by enhancing composability. For example, XLA has a single op for convolution (`kConvolution`) letting the compiler handle all the memory layout variations. The TF intermediate form has nine; for example, `Conv2D`, `Conv2dBackpropFilter`, `DepthwiseConv2dNative`, and `DepthwiseConv2dNativeBackpropFilter`. For the CNN1 program, the XLA compiler fused 63 different operations with at least one `kConvolution`.

Since ML platforms and DSAs offered a new set of compiler challenges, it was unclear how fast they would improve. Table 5 shows the median gain over only six months for MLPerf from version 0.5 to 0.6 was 1.3x for GPUs and 2.1x for TPUs! (Perhaps the younger XLA compiler has more opportunity to improve than the more mature CUDA stack.) One reason for the large gain is the focus on benchmarks, but production applications also advanced. Increasing bf16 use, optimizing model architecture, and XLA generating better code sped up CNN0 by 1.8x in 15 months and improving partitioning/placement for embeddings and XLA optimizations accelerated MLP0 by 1.65x.

Contrasting GPU and TPU Architectures

As details of TPU and GPU architectures are now public, let us compare

TPU and GPU choices before we compare performance.

Multi-chip parallelization is built into TPUs through ICI and supported through all-reduce operations plumbed through XLA to TF. Similar-sized multi-chip GPU systems use a tiered networking approach, with NVIDIA’s NVLink inside a chassis and host-controlled InfiniBand networks and switches to tie multiple chassis together.

TPUs offer bf16 FP arithmetic designed for DNNs inside 128x128 systolic arrays that halves the die area and energy versus IEEE fp16 FP multipliers. Volta GPUs have also embraced reduced-precision systolic arrays, with a finer granularity—4x4 or 16x16 depending on hardware or software descriptions—while using fp16 rather than bf16, so they may require software to perform loss scaling plus extra die area and energy.

TPUs are dual-core, in-order machines, where the XLA compiler overlaps computation, memory, and network activities. GPUs are latency-tolerant many-core machines, where each core has many threads and thus very large (20MiB) register files. Threading hardware plus CUDA coding conventions support overlapped operations.

TPUs use software controlled 32MiB scratchpad memories that the compiler schedules, while Volta hardware manages a 6MiB cache and software manages a 7.5MiB scratchpad memory. The XLA compiler directs sequential DRAM accesses typical of DNNs via direct memory access (DMA) controllers on TPUs while GPUs use multithreading plus coalescing hardware for them.

Thottethodi and Vijaykumar³⁵ concluded that when compared to TPUs:

“[GPUs] incur high overhead in performance, area, and energy due to heavy multithreading which is unnecessary for DNNs which have prefetchable, sequential memory accesses. The systolic organization [of TPUs] ... capture[s] DNNs’ data reuse while being simple by avoiding multithreading.”

In addition to the contrasting architectural choices, TPU and GPU chips use different technologies, die areas, clock rates, and power. Table 6 gives three related cost measures of these systems: approximate die size adjusted for technology; power for a 16-chip

Table 5. Speedup of MLPerf 0.6 over 0.5 in six months.

	ResNet50	SSD	MaskRCNN	NMT	Transformer	Median
Volta	1.3	1.2	1.8	1.0	2.0	1.3
TPUv3	1.4	1.4	3.5	2.1	3.0	2.1

Table 6. Adjusted comparison of GPU and TPU.

Die sizes are adjusted by the square of the technology, as the semiconductor technology for TPUs is similar but larger and older than that of the GPU. We picked 15nm for TPUs based on the information in Table 3. Thermal Design Power (TDP) is for 16-chip systems. TPUs come with a host CPU. This GPU price adds price of a n1-standard-16 CPU.

	Die size	Adjusted die size	TD (kw)	Cloud price	Relative to GPU		
					Die	TDP	Price
Volta	815	815	12.0	\$3.24	1.00	1.00	1.00
TPUv2	<611	<391	7.7	\$1.13	<0.5	0.64	0.35
TPUv3	<648	<415	9.3	\$2.00	<0.5	0.78	0.62

system; and cloud price per chip. The GPU adjusted die size is more than twice that of the TPUs, which suggests the capital costs of the chips is at least double, since there would be at least twice as many TPU dies per wafer. GPU power is 1.3x–1.6x higher, which suggests higher operating expenses, as the total cost of ownership is correlated with power.¹⁹ Finally, the hourly rental prices on Google Cloud Engine are 1.6x–2.9x higher for the GPU. These three different measures consistently suggest TPUv2 and TPUv3 are roughly half to three fourths as expensive as the Volta GPU.

Performance Evaluation

In computer architecture, we “grade on a curve” versus “grade on an absolute scale,” so we need to measure performance relative to the competition. Before showing performance of TPU supercomputers, we must establish the virtues of a single chip, for a 1024x speedup from 1,024 wimpy chips is uninteresting.

We first compare training performance for a standard set of ML benchmarks and Google production applications for TPUv2/v3 chip and the Volta GPU chip; TPUv3 and Volta are about the same speed. We then check if four MXUs per chip in TPUv3 really helped, or if other bottlenecks in the TPUv3 chip made the extra MXUs superfluous; they helped! We conclude the chip comparison looking at inference for TPUv2/v3 versus TPUv1; TPUv2/v3 are much faster.

Having established the merits of the TPU chips, we then evaluate the TPUv2/v3 supercomputer. The first step is to see how well it scales; we see 96%–99% of perfect linear speedup at 1024 chips. We then compare the fraction of peak performance and performance per Watt of TPU and traditional supercomputers; TPUs have 5x-10x better performance per Watt.

Chip performance: TPUv2/v3 versus the Volta GPU. Figure 6 shows the performance of TPUv3 and the Volta GPU over TPUv2 for two sets of programs. The first set is five programs that Google and NVIDIA both submitted to MLPerf 0.6 in May 2019, and both use 16-bit multiplication with NVIDIA software performing loss scaling. The geometric mean speedup of these programs over TPUv2 is 1.8 for TPUv3 and 1.9 for Volta.

Figure 6. Performance per chip relative to TPUv2 for five MLPerf 0.6 benchmarks and six production applications.

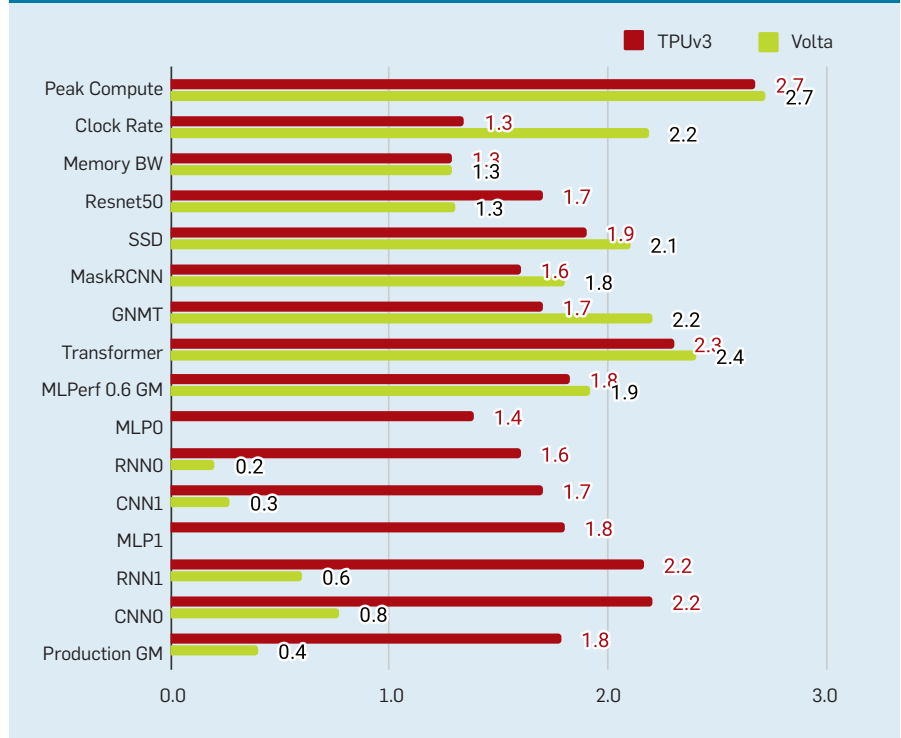


Table 7. Google’s inference (July 2016) and training (April 2019) workloads by DNN model type.

DNN Model	TPUv1 July 2016	TPUv3 April 2019
MLP	61%	27%
RNN	29%	21%
CNN	5%	24%
Transformer	–	21%

We also wanted to measure performance of production workloads. We chose six production applications similar to what we used for TPUv1 as representative of Google’s workload:

- ▶ In *MultiLayer Perceptrons* (MLP) each new layer of a model is a set of nonlinear functions of a weighted sum of all outputs (fully connected) from a prior one. This classic DNN usually has text as input. MLP0 is unpublished but MLP1 is RankBrain,⁹ which ranks search results for a Web page.

- ▶ In *Convolutional Neural Networks* (CNN), each ensuing layer is a set of nonlinear functions of weighted sums of spatially nearby subsets of outputs from the prior layer. CNNs usually have images as inputs. CNN0 is AlphaZero, a reinforcement learning algorithm with extensive use of CNNs, which mastered the games chess, Go, and shogi.³⁴ CNN1 is a Google-internal

model for image recognition.

- ▶ In *Recurrent Neural Networks* (RNN), each subsequent model layer is a collection of nonlinear functions of weighted sums of outputs *and* the previous state. Sequence prediction problems, such as language translation, use RNNs. RNN0 is RNMT+⁶ and RNN1 is Improved LAS.⁸

We recently compared the representative datacenter workloads by model type for inference on TPUv1²⁰ versus TPUv2/v3 for training. Table 7 illustrates the fast-changing nature of DNNs. We originally used the name LSTM (Long Short-Term Memory) for TPUv1 applications, a type of RNN. Although sampled three years apart—July 2016 versus April 2019—we were still surprised that CNNs were a much larger part of datacenter training, and that a new model *Transformer*³⁶—published the year that TPUv2 was de-

ployed—was as popular as RNNs. (Transformer is part of MLPerf 0.5.)

Transformer is intended for the same tasks as RNNs, such as translation, but is considerably faster since it lends itself to parallelization while RNNs have sequential dependencies. The layers of Transformer are a mix of MLPs and attention layers.⁴ Attention is the key new mechanism used in Transformer; it lets neural networks look up data associatively, in a memory-like structure whose indices themselves are learned. The components of attention resemble those of other layers, including matrix multiplications and dot products, which map well to TPU hardware. One difference is that attention matrices grow with sequence length, adding dynamic shape and memory requirements that complicate some optimizations done by XLA. The success of this recent model (see Figure 6) highlights TPU programmability.

The geometric mean speedup of the six production applications was 1.8 for TPUv3 but only 0.4 for Volta, primarily because they use 8x slower fp32 on GPUs instead of fp16 (Table 3). These are large production applications that

are continuously improved, and not simple benchmarks, so it's a lot of work to get them to run at all, and more to run well. As noted earlier, application programmers focus on TPUs, since they are in everyday use, so there is little urge to include loss scaling needed for fp16. (TF kernels for embeddings have not been developed for GPUs, so we exclude MLPs from the GPU geometric mean as they could not run.)

Is TPUv3 memory bound or compute bound? While the peak compute improvement of TPUv3 over TPUv2 is 2.7x, the improvements in memory bandwidth, ICI bandwidth, and clock rate are only $\approx 1.35x$. We wondered whether the extra MXUs in TPUv3 would be underutilized due to bottlenecks elsewhere. Figure 6 shows that one production application runs a bit higher than the memory improvement at 1.4x, but the other five and all the MLPerf 0.6 benchmarks run much faster at 1.6x to 2.3x. The large application batch sizes and sufficient on-chip storage enabled these good results. As the MXUs are not a large part of the chip (Figure 3), doubling the MXUs in TPUv3 clearly proved beneficial.

Inference on a training chip: TPUv2/v3 vs. TPUv1. What about inference speed? Running it on a training chip—which works since it is like the forward pass—could help applications that require frequent training on fresh data. TPUv2/v3 do not support 8-bit integer data types, so inference uses bf16. One upside of using the same arithmetic for training and inference is that ML experts don't need to do extra work—called *quantization*—to ensure the same accuracy of the DNN model.

One danger is the larger batch sizes needed to run efficiently on TPUv2/v3 could hurt inference latency. Fortunately, we have DNN models that can meet their latency targets with batch sizes of greater than 1,000. With billions of daily users, inferences per second across the whole data center fleet can be very high.

The LSTM0 benchmark, for instance, ran at 48 inferences per second with a response time of 122ms on TPUv1.¹⁹ TPUv2 runs it 5.6x as fast with a 2.8x lower response time (44ms) at the same batch size. The lower latency in turn allows for larger batches compared to TPUv1 to be served in production yet still meet latency targets. With larger batches, the throughput rose to 11x with a latency improvement of 2x (58ms) vs TPUv1. TPUv3 reduces latency 1.3x (45ms) versus TPUv2 at the same batch size.

DSA supercomputer scaling performance. Alas, only ResNet-50 from MLPerf 0.6 can scale beyond 1,000 TPUs and GPUs. Figure 7 shows three ResNet-50 results. Ying et al. published a ResNet-50 results on TPUv3 that delivered 77% of perfect linear scaleup at 1,024 chips,⁴¹ but the TPUv3 version for MLPerf 0.6 only runs at 52%. The difference is in MLPerf's ground rules. MLPerf requires including *evaluation* in the training time. (Evaluation runs a holdout dataset after a model training finishes to determine its accuracy.) Like Ying et al., most researchers exclude it when reporting performance. More unusually, MLPerf requires running evaluation at the end of every four epochs to deter benchmark cheating. ML developers would never evaluate that frequently. For MLPerf 0.6, NVIDIA ran ResNet-50 on a cluster of 96 DGX-2H each with 16 Voltas connected via Infiniband switches at 41% of linear scaleup for 1,536 chips.

Figure 7. Supercomputer scaling: TPUv3 and Volta.

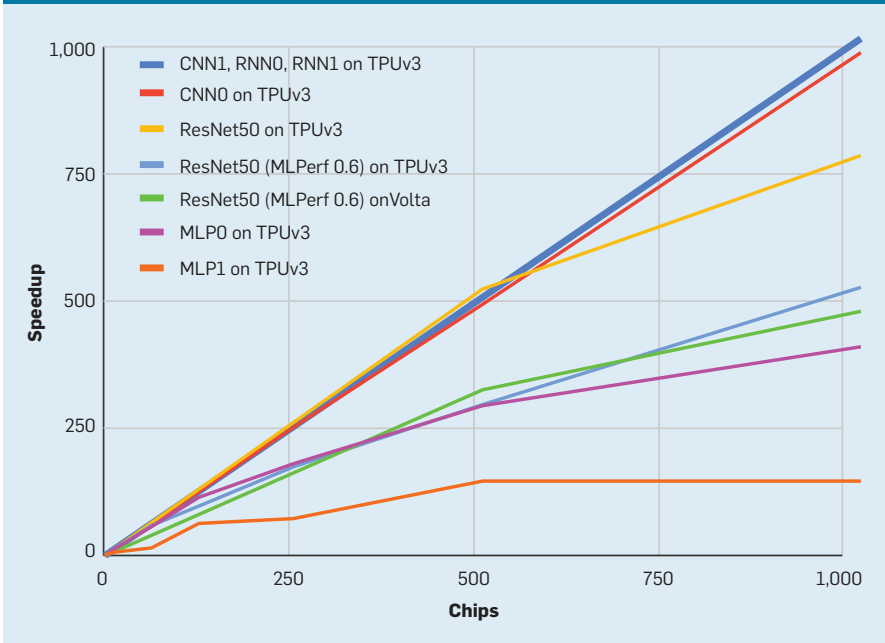


Table 8. Days to train MLPerf 0.5 benchmarks on one TPUv2 chip. See Table 1 for time to train production applications.

ResNet50	SSD	Mask R-CNN	GNMT	Transformer
0.8	0.3	1.9	0.2	0.3

Table 9. Traditional versus TPU supercomputer Top500 and Green500 rank (June 2019) for Linpack and AlphaZero.

Name	Cores	Benchmark	Data	Peta Flop/s	% of Peak	Mega-watts	GFlop/Watt	Top 500	Green 500
Tianhe	4865k	Linpack	32/64 bit	61.4	61%	18.48	3.3	4	57
SaturnV	22k	Linpack	32/64 bit	1.1	59%	0.97	5.1	469	1
ABCI	392k	Linpack	32/64 bit	19.9	61%	1.65	14.4	8	3
TPUv2	0.5k	AlphaZero	16/32 bit	9.9	84%	0.12	79.9	22	2
TPUv3	2k	AlphaZero	16/32 bit	86.9	70%	0.59	146.3	4	1

See article for caveats about comparing Linpack on 64-bit floating point to ML training on 16-bit floating point.

MLPerf 0.6 benchmarks are much smaller than the production applications; Table 8 shows time to train them on one TPUv2 chip is orders of magnitude less than in Table 1. Thus, we include six production applications largely to show substantial programs that can scale to supercomputer size. The MLPs are limited by embeddings and run only at 14% and 40% of perfect linear scale up on 1,024 TPUv3 chips, but one runs at 96% and three at 99%!

Note that CNN1 is an image recognition DNN much like ResNet101. It scales much better on TPUs because Google’s internal image datasets are much larger than what ResNet50 uses (Imagenet).

Traditional vs. DSA supercomputer performance. Traditional supercomputers measure performance using the high-performance computing (HPC) benchmark Linpack and ranking the Top500 (top500.org). The related Green500 list re-ranks the Top500 based on performance per Watt. For these large computers to get utilization above 60%, HPC expands the size of the matrix being solved (*weak scaling*). (For which Linpack has long been criticized within HPC.¹³) The TPU scale up, however, uses production programs on real-world datasets.

Table 9 shows where PetaFLOPs/second and FLOPs/Watt of AlphaZero on TPUv2/v3 would rank in the Top500 and Green500 lists. This comparison is imperfect: conventional supercomputers crunch 32- and 64-bit data rather than the 16- and 32-bit data of TPUs. However, TPUs are running a real application on real data versus a weakly scaled benchmark on synthetic data. TPUv3 has 44x the FLOPs/Watt of Tianhe and 10x of SaturnV and ABCI.

The Fujitsu ABCI supercomputer in Table 9 includes 2,176 Intel CPUs along with 4352 Volta GPUs. Besides

Table 10. Time to train supercomputers from NVIDIA, Fujitsu, and Google on the ResNet-50 benchmark from MLPerf 0.6.

	NVIDIA cluster	ABCI Supercomputer	TPUv3 Supercomputer
MLP	1536 Voltas + 192 CPUs	2048 Voltas + 1024 CPUs	1024 TPUv3s + 128 CPUs
Transformer	80 seconds	70 seconds	77 seconds

running Linpack, Fujitsu submitted a ResNet-50 result for MLPerf 0.6 using 2,048 GPUs. Table 10 shows time to train for ResNet-50 in MLPerf 0.6 and the number of chips for an NVIDIA GPU cluster, the Fujitsu ABCI supercomputer, and a Google TPUv3 supercomputer. Fujitsu varied from the strict benchmark MLPerf 0.6 closed guidelines of the other submissions—they changed the LARS optimizer and the momentum hyperparameter—so it’s not an apples-to-apples comparison. These changes improve performance by 10%–15%, which would also help NVIDIA and TPUv3.

Related Work

A survey documents over 25 years of custom neural network chips,³ but recent DNN successes led to an explosion in their development. Most designs focus on inference; far fewer, including the TPUv2/v3, target training. We are not aware of any other results that show state-of-the-art accuracy on a working DSA hardware for training.

Of the five training startups, SambaNova has not yet published. Cerebras uses a whole silicon wafer to build their system, essentially treating 84 large “dies” as a single unit.²⁴ Each “die” has 220MB of SRAM along with about 5k cores, yielding a total of 18GB of on-chip memory and 400k cores that collectively use 15 kilowatts. Like GraphCore, there is no DRAM in the system, so they target small batch sizes to reduce memory needs. The GraphCore¹⁵ GC2 chip holds 1,216 Intelligence Processing Units that support

seven threads, each of which has a peak performance of 100GFLOPs/s or 122TFLOPs/s per chip, almost identical to the peak performance of TPUv3 and Volta. It relies on the 300MB on-chip SRAM for memory, with two GC2 chips per PCIe board. The Habana Gaudi³⁸ has eight VLIW SIMD cores, four stacks of HBM2 memory, bf16 arithmetic, and eight 100Gbit/sec Ethernet links to connect many chips together to form larger systems. Wave Computing’s²⁸ Dataflow Processing Unit chip has 16k processors, 8k arithmetic units, 16MB of on-chip memory, and novelty relies on asynchronous logic instead of a clock. It has external DRAM, offering both Hybrid Memory Cube and DDR4 ports. As of February 2020, none of the five training startups has reported training accuracy or time-to-solution.

Academic training studies include the DianNao family of architectures (one of which trains)⁷ and ScaleDeep;³⁷ to our knowledge, neither has been fabricated.

Several studies explored reduced-precision training with accelerator construction in mind. Intel’s Flexpoint²² is a block FP format,³⁹ although those developers switched to using bf16 for their DNN chips.⁴⁰ De Sa et al.¹⁰ reduced precision and relaxed cache coherence. HALP¹¹ also made algorithmic changes to reduce quantization noise and uses 8-bit integers to train some models. None is yet available in a commercial system.

TPUv2/v3 are not the first domain-specific supercomputers to show large efficiency, performance, and scaling

gains. Anton systems³³ showed two order-of-magnitude speedups over traditional supercomputers on molecular dynamics workloads. They also resulted from hardware/software/algorithm codesign, with custom chips, interconnect, and arithmetic.

Conclusion

Benchmarks suggests the TPUv3 chip performs similarly to the contemporary Volta GPU chip, but parallel scaling for production applications is stronger for the TPUv3 supercomputer:

- ▶ Three scale to 1,024 chips at 99% linear speedup;
- ▶ One scales to 1,024 chips at 96% linear speedup; and
- ▶ Two scale to 1,024 chips but are limited by embeddings.

Remarkably, a TPUv3 supercomputer runs a production application using real-world data at 70% of peak performance, higher than general-purpose supercomputers run the Linpack benchmark using weak scaling of manufactured data. Moreover, TPU supercomputers with 256–1,024 chips running a production application have 5x–10x performance/Watt of the #1 traditional supercomputer on the Green500 list running Linpack and 24x–44x of the #4 supercomputer on the Top500 list. Reasons for this success include the built-in ICI network, large systolic arrays, and bf16 arithmetic, which we expect will become a standard data type for DNN DSAs.

TPUv2/v3 have smaller dies in an older semiconductor process and lower cloud prices despite being less mature at many levels of hardware/software system stack than CPUs and GPUs. These good results despite technological disadvantages suggests the TPU approach is cost-effective and can deliver high architectural efficiency into the future.

Going forward, our ravenous DNN colleagues want the fastest computer that we can build.² Despite Moore's Law ending, we expect the demand for faster DNN-specific supercomputers to grow even more quickly than Moore predicted. Trying to satisfy that demand without the help of Moore's Law offers exciting new challenges for computer architects for at least a decade.¹⁷

Acknowledgments

The authors analyzed TPU systems that involved contributions from many

Googlers. Many thanks to the hardware and software teams and engineers for making TPU supercomputers possible, including Paul Barham, Eli Bendersky, Dehao Chen, Chiachen Chou, Jeff Dean, Peter Hawkins, Blake Hechtman, Mark Heffernan, Robert Hundt, Michael Isard, Fritz Kruger, Naveen Kumar, Sameer Kumar, Chris Leary, Hyouk-Joong Lee, David Majnemer, Lifeng Nai, Thomas Norrie, Tayo Oguntebi, Andy Phelps, Bjarke Roune, Brennan Saeta, Julian Schrittwieser, Andy Swing, Shibo Wang, Tao Wang, Yujing Zhang, and many more. **C**

References

1. Abadi, M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. 2016; arXiv preprint arXiv:1603.04467.
2. Amodei, D. and Hernandez, D. AI and compute, 2018; <https://blog.openai.com/aiandcompute>.
3. Asanović, K. Programmable neurocomputing. *The Handbook of Brain Theory and Neural Networks, 2nd Edition*. M.A. Arbib, ed. MIT Press, 2002.
4. Bahdanau, D., Cho, K. and Bengio, Y. Neural machine translation by jointly learning to align and translate. 2014; arXiv preprint arXiv:1409.0473.
5. Chen, J. et al. Revisiting distributed synchronous SGD. 2016; arXiv preprint arXiv:1604.00981.
6. Chen, M.X. et al. The best of both worlds: Combining recent advances in neural machine translation. 2018; arXiv preprint arXiv:1804.09849.
7. Chen, Y. et al. Dadiannao: A machine-learning supercomputer. In *Proceedings of the 47th Int'l Symp. on Microarchitecture*, (2014), 609–622.
8. Chiu, C.C. et al. State-of-the-art speech recognition with sequence-to-sequence models. In *Proceedings of the IEEE Int'l Conference on Acoustics, Speech and Signal Processing*, (Apr. 2018), 4774–4778.
9. Clark, J. Google turning its lucrative Web search over to AI machines. Bloomberg Technology, Oct. 26, 2015.
10. De Sa, C. et al. Understanding and optimizing asynchronous low-precision stochastic gradient descent. In *Proceedings of the 44th Int'l Symp. on Computer Architecture*, (2017), 561–574.
11. De Sa, C. et al. High-accuracy low-precision training. 2018; arXiv preprint arXiv:1803.03383.
12. Dean, J. et al. Large scale distributed deep networks. *Advances in Neural Information Processing Systems*, (2012), 1223–1231.
13. Dongarra, J. The HPC challenge benchmark: a candidate for replacing Linpack in the Top500? In *Proceedings of the SPEC Benchmark Workshop*, (Jan. 2007); www.spec.org/workshops/2007/austin/slides/Keynote_Jack_Dongarra.pdf.
14. Duchi, J., Hazan, E. and Singer, Y., Adaptive subgradient methods for online learning and stochastic optimization. *J. Machine Learning Research* 12 (July 2011), 2121–2159.
15. Graphcore Intelligence Processing Unit. (<https://www.graphcore.ai/products/ipu>)
16. Hennessy, J.L. and Patterson, D.A. *Computer Architecture: A Quantitative Approach, 6th Edition*. Elsevier, 2019.
17. Hennessy, J.L. and Patterson, D.A. A new golden age for computer architecture. *Commun. ACM* 62, 2 (Feb. 2019), 48–60.
18. Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015; arXiv preprint arXiv:1502.03167.
19. Jouppi, N.P. et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Int'l Symp. on Computer Architecture*, (June 2017), 1–12.
20. Jouppi, N.P., Young, C., Patil, N. and Patterson, D. A domain-specific architecture for deep neural networks. *Commun. ACM* 61, 9 (Sept. 2018), 50–59.
21. Kalamkar, D. et al. A study of Bfloat16 for deep learning training. 2019; arXiv preprint arXiv:1905.12322.
22. Köster, U. et al. Flexpoint: An adaptive numerical

format for efficient training of deep neural networks. In *Proceedings of the 31st Conf. on Neural Information Processing Systems*, (2017).

23. Kung, H.T. and Leiserson, C.E. Algorithms for VLSI processor arrays. *Introduction to VLSI Systems*, 1980.
24. Lie, S. Wafer scale deep learning. In *Proceedings of the IEEE Hot Chips 31 Symp.*, (Aug 2019).
25. Mellemudi, N. et al. Mixed precision training with 8-bit floating point. 2019; arXiv preprint arXiv:1905.12334.
26. Micikevicius, P. et al. Mixed precision training. 2017; arXiv preprint arXiv:1710.03740.
27. Mikolov, T. et al. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (2013), 3111–3119.
28. Nicol, C. A dataflow processing chip for training deep neural networks. In *Proceedings of the IEEE Hot Chips 29 Symp.*, (Aug 2017).
29. Olah, C. Deep learning, NLP, and representations. Colah's blog, 2014; <http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/>.
30. Polyak, B.T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 4, 5 (1964), 1–17.
31. Robbins, H. and Monro, S. A Stochastic approximation method. *The Annals of Mathematical Statistics* 22, 3 (Sept. 1951), 400–407.
32. Shallue, C.J. et al. Measuring the effects of data parallelism on neural network training. 2018; arXiv preprint arXiv:1811.03600.
33. Shaw, D.E. et al. Anton, a special-purpose machine for molecular dynamics simulation. *Commun. ACM* 51, 7 (July 2008), 91–97.
34. Silver, D. et al. A general reinforcement learning algorithm that Master's chess, Shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.
35. Thottethodi, M. and Vijaykumar, T. Why the GPGPU is less efficient than the TPU for DNNs. *Computer Architecture Today Blog*, 2019; www.sigarch.org/why-the-gpgpu-is-less-efficient-than-the-tpu-for-dnns/
36. Vaswani, A. et al. Attention is all you need. *Advances in Neural Information Processing Systems* (2017), 5998–6008.
37. Venkataramani, S. et al. Scaleddeep: A scalable compute architecture for learning and evaluating deep networks. In *Proceedings of the 45th Int'l Symp. on Computer Architecture*, (2017), 13–26.
38. Ward-Foxton, S. Habana debuts record-breaking AI training chip, (June 2019); https://www.eetimes.com/document.asp?doc_id=1334816.
39. Wilkinson, J.H. *Rounding Errors in Algebraic Processes, 1st Edition*. Prentice Hall, Englewood Cliffs, NJ, 1963.
40. Yang, A. Deep learning training at scale Spring Crest Deep Learning Accelerator (Intel® Nervana™ NNP-T). In *Proceedings of the Hot Chips*, (Aug. 2019); www.hotchips.org/hc31/HC31_1.12_Intel_Intel_AndrewYang.v0.92.pdf.
41. Ying, C. et al. Image classification at supercomputer scale. 2018; arXiv preprint arXiv:1811.06992.
42. Zoph, B. and Le, Q.V. Neural architecture search with reinforcement learning. 2019; arXiv preprint arXiv:1611.01578.

Norman P. Jouppi is a Distinguished Hardware Engineer at Google, Mountain View, CA, USA.

Doe Hyun Yoon is a staff software engineer at Google, Mountain View, CA, USA.

George Kurian is a senior staff software engineer at Google, Mountain View, CA, USA.

Sheng Li is a staff software engineer and tech lead on ML Accelerator Optimization at Scale at Google, Mountain View, CA, USA.

Nishant Patil is a senior staff software engineer at Google, Mountain View, CA, USA.

James Laudon is an engineering director at Google, Mountain View, CA, USA.

Cliff Young is a software engineer at Google, Mountain View, CA, USA.

David Patterson is a Distinguished Engineer at Google, Mountain View, CA, USA, a professor of Graduate School at the University of California, Berkeley, CA, USA, and Director of the RISC-V International Open Source Laboratory at Berkeley, CA, and Shenzhen, China.

Copyright held by authors/owners.

ACM Welcomes the Colleges and Universities Participating in ACM's Academic Department Membership Program

ACM offers an Academic Department Membership option, which allows universities and colleges to provide ACM Professional Membership to their faculty at a greatly reduced collective cost.

The following institutions currently participate in ACM's Academic Department Membership program:

- Abilene Christian University
- Afrisol Technical College, Zimbabwe
- Alfred State College
- Amherst College
- Appalachian State University
- Augusta University, School of Computer and Cyber Sciences
- Ball State University
- Bellevue College
- Berea College
- Binghamton University
- Boise State University
- Bridgewater State University
- Bryant University
- California Baptist University
- Calvin College
- Clark University
- Colgate University
- Colorado School of Mines
- Columbus State University
- Cornell University
- Creighton University
- Cuyahoga Community College
- Denison University
- European University (Tbilisi, Georgia)
- Franklin University
- Gallaudet University
- Georgia Institute of Technology
- Georgia State University Perimeter College
- Governors State University
- Harding University
- Harvard University
- Harvey Mudd College
- Hochschule für Technik Stuttgart - University of Applied Sciences
- Hofstra University
- Hope College
- Howard Payne University
- Indiana University Bloomington
- Kent State University
- Klagenfurt University, Austria
- Madinah College of Technology, Saudi Arabia
- Massasoit Community College
- Messiah College
- Metropolitan State University
- Missouri State University
- Modesto Junior College
- Monash University, Australia
- Montclair State University
- Mount Holyoke College
- New Jersey Institute of Technology
- New Mexico State University
- Northeastern University
- Ohio State University
- Old Dominion University
- Pacific Lutheran University
- Pennsylvania State University
- Potomac State College of West Virginia University
- Purdue University Northwest
- Regis University
- Rhodes College
- Rochester Institute of Technology
- Rutgers University
- Saint Louis University
- San José State University
- Shippensburg University
- Simmons University
- Spelman College
- St. John's University
- Stanford University
- State University of New York at Fredonia
- State University of New York at Oswego
- Stetson University
- Trine University
- Trinity University
- Union College
- Union University
- Univ. do Porto, Faculdade de Eng. (FEUP)
- University at Albany, State University of New York
- University of Alabama
- University of Arizona
- University of California, Riverside
- University of California, San Diego
- University of Colorado Boulder
- University of Colorado Denver
- University of Connecticut
- University of Houston
- University of Illinois at Chicago
- University of Jamestown
- University of Liechtenstein
- University of Lynchburg
- University of Maribor, Slovenia
- University of Maryland, Baltimore County
- University of Memphis
- University of Namibia
- University of Nebraska at Kearney
- University of Nebraska Omaha
- University of New Mexico
- University of North Dakota
- University of Pittsburgh
- University of Puget Sound
- University of Southern California
- University of St. Thomas
- University of the Fraser Valley
- University of Victoria, BC Canada
- University of Wisconsin–Parkside
- University of Wyoming
- Virginia Commonwealth University
- Wake Forest University
- Wayne State University
- Wellesley College
- Western New England University
- William Jessup University

Through this program, each faculty member receives all the benefits of individual professional membership, including *Communications of the ACM*, member rates to attend ACM Special Interest Group conferences, member subscription rates to ACM journals, and much more.

Blockchain technology can shape innovation and competition in digital platforms, but under what conditions?

BY CHRISTIAN CATALINI AND JOSHUA S. GANS

Some Simple Economics of the Blockchain

IN OCTOBER 2008, a few weeks after the Emergency Economic Stabilization Act rescued the U.S. financial system from collapse, Satoshi Nakamoto³⁴ introduced a cryptography mailing list to Bitcoin, a peer-to-peer electronic cash system “based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.” With Bitcoin, for the first time, value could be reliably transferred between two distant, untrusting parties without the need of an intermediary. Through a clever combination of cryptography and game theory, the Bitcoin ‘blockchain’—a distributed, public transaction ledger—could be used by any participant in the network to cheaply verify and settle transactions in the cryptocurrency. Thanks to rules designed to incentivize the propagation of new

legitimate transactions, to reconcile conflicting information, and to ultimately agree at regular intervals about the true state of a shared ledger (a blockchain)^a in an environment where not all participating agents can be trusted, Bitcoin was also the first platform, at scale, to rely on decentralized, Internet-level ‘consensus’ for its operations. Without involving a central clearinghouse or market maker, the platform was able to settle the transfer of property rights in the underlying digital token (bitcoin) by simply combining a shared ledger with an incentive system designed to securely maintain it.

From an economics perspective, this new market design solution provides some of the advantages of a centralized digital platform (for example, the ability of participants to rely on a shared network and benefit from network effects) without some of the consequences the presence of an intermediary may introduce such as increased market power, ability to renege on commitments to ecosystem participants, control over participants’ data, and presence of a single point of failure. As a result, relative to existing financial networks, a cryptocurrency such as Bitcoin may be able to offer lower barriers to entry for new service providers and application developers, and an alternative monetary policy for

a See online appendix for more details; <https://dl.acm.org/doi/10.1145/3359552>

» key insights

- We discuss how blockchain technology can shape innovation and competition by identifying two key costs affected by the technology: the cost of verification and the cost of networking.
- The cost of verification relates to the ability to cheaply verify state.
- The cost of networking relates to the ability to bootstrap and operate a marketplace without assigning control to a centralized intermediary. This is achieved by combining the ability to verify state with economic incentives targeted at rewarding state transitions that are particularly valuable from a network perspective.

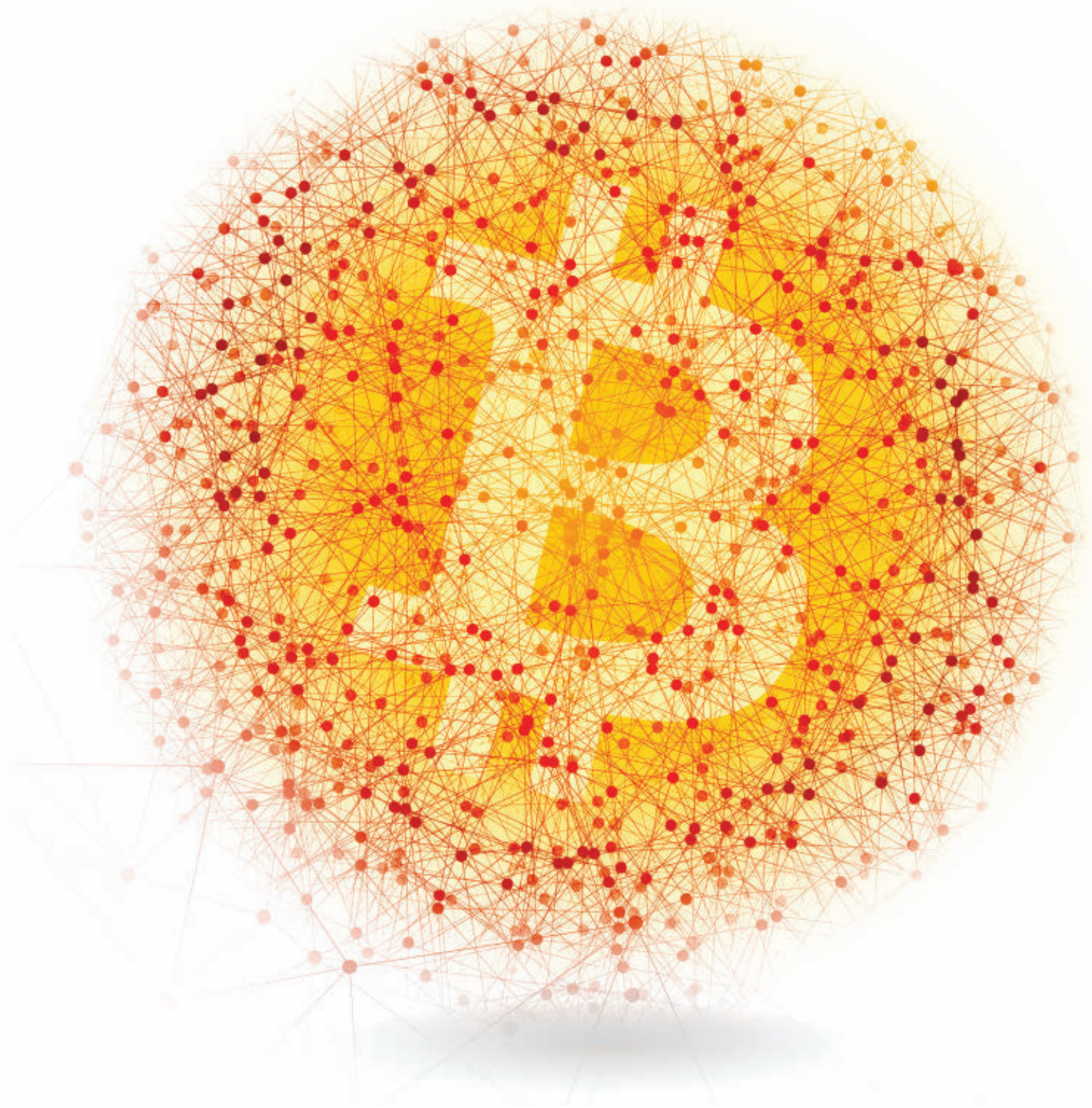


IMAGE BY ANDRZJ BORYS ASSOCIATES

individuals that do not live in countries with trustworthy institutions. Key commitments encoded in the Bitcoin protocol are its fixed supply, predetermined release schedule, and the fact that rules can only be changed with support from a majority of participants. While the resulting ecosystem may not offer an improvement for individuals living in countries with reliable and independent central banks, it may represent an option in countries that are unable to maintain their monetary policy commitments. Of course,

the open and “permissionless” nature of the Bitcoin network, and the inability to adjust its supply also introduce new challenges, as the network can be used for illegal activity, and the value of the cryptocurrency can fluctuate wildly with changes in expectations about its future success, limiting its use as an effective medium of exchange.

In the article, we rely on economic theory to explain how two key costs affected by blockchain technology—the cost of verification of state, and the cost of networking—change the types

of transactions that can be supported in the economy. These costs have implications for the design and efficiency of digital platforms, and open opportunities for new approaches to data ownership, privacy, and licensing; monetization of digital content; auctions and reputation systems.

While the reduction in the cost of verification has economic consequences mostly on the intensive margin of production (improving existing applications), on the extensive margin (new applications), the reduction in the cost

of networking is more consequential: Bitcoin was the first digital platform to be bootstrapped in a decentralized fashion without resorting to investments by an intermediary or planner. As early adopters and investors experimented with the cryptocurrency in the hope that the network would increase in users, security^b and value, the underlying token appreciated, generating the positive feedback loop needed to attract subsequent batches of users. This organic diffusion process uses high-powered incentives similar to the venture capital model to reward early adopters for taking risks and dedicating their time, effort, and capital to a new platform. The same incentive system is now used by startups to raise capital and lower switching costs for the user base and developer community of entrenched digital incumbents. This allows them to compete in a context where network effects are strongly in favor of established players.

Whereas the reduction in the cost of verification is what allows Bitcoin to settle transactions without an intermediary, the reduction in the cost of networking is what allowed its ecosystem to scale in the first place: Within eight years, the digital, scarce token native to Bitcoin went from having no value to a total market capitalization of \$180B,^c and is considered by investors to be part of a new asset class and a novel type of store of value.

Beyond the idiosyncratic market design choices behind Bitcoin, the ability to track transaction attributes, settle trades, and enforce contracts across a wide variety of digital assets is what makes blockchain technology a general-purpose technology. Entries on a distributed ledger can represent ownership in currency, digital content, intellectual property, equity, information, contracts, financial and physical assets. As a result, the scaling model pioneered by Bitcoin has been adopted

by open source projects and startups interested in creating platforms for the exchange of other types of scarce, digital goods. For example, Ethereum used its own token, Ether, to bootstrap a decentralized marketplace for computing power and applications, Filecoin for data storage, BAT for digital advertising, and Blockstack for digital identity.

The new types of networks that can be created using the technology challenge the business models of incumbent digital platforms and financial institutions, and open opportunities for novel approaches to the exchange of digital assets, data ownership and monetization, information licensing, and privacy. Whereas the utopian view has argued that blockchain has the potential to transform every digital service by removing the need for intermediaries, we argue it is more likely to change the nature of intermediation by reducing the market power of intermediaries, and by progressively redefining how they add value to transactions.^d This transformation will unfold slowly because even in sectors that are well-suited for a more decentralized exchange of digital assets such as finance, there are currently substantial legal and regulatory frictions to adoption. While blockchain allows for the costless verification of state when all relevant information is born digital, most markets also rely on external information—including information about identity—to ensure safe and compliant exchanges. As a result, ‘last mile’ frictions limit the conditions under which blockchain-based networks can replace existing infrastructure, as complementary innovations are needed to ensure that the shared data managed through a consensus protocol is kept in sync with critical offline information and events.

After reviewing pertinent literature, we discuss the effects of the reduction in the cost of verification, later focusing on the reduction in the cost of bootstrapping and operating a network.

^d While financial intermediaries are charging high fees for cross-border payments, this revenue stream will disappear if blockchain-based payment networks commodify the transfer of value. This does not mean that intermediaries will not be able to provide added value services on top of basic payments.

Literature

This article contributes to the nascent literature on blockchain by providing an economic framework for understanding how the technology changes the types of transactions and networks that can be sustained in the economy. By focusing on the two key economic costs the technology influences, we abstract away from some of the idiosyncratic choices different protocols make (for example, in terms of privacy, consensus algorithms, and presence of mining versus not), and surfaces high-level dimensions that have implications for market structure and competition with existing digital platforms. This level of analysis allows us to highlight commonalities between protocols that may be different at a more fine-grained technical level, but ultimately share a similar trust and competition model, and will thus have a similar impact on how rents are allocated between users, developers and nodes providing resources to a network. An online appendix (<https://dl.acm.org/doi/10.1145/3359552>) provides additional technical details on how some of the most popular cryptocurrencies work, and a taxonomy of transactions that the technology can support (for example, auctions, smart contracts, digital identity and property rights, and audit trails).

Previous research in this emerging area has focused on providing an overview of Bitcoin and its operations;^{7,35} has combined theory and data to explain the velocity of Bitcoin and its use across countries as an investment vehicle, for gambling and illegal online markets;² and has studied the role early adopters play in the diffusion and use of Bitcoin within a large-scale, field experiment.¹⁵

Researchers have also examined competition between alternative cryptocurrencies and their differences;^{17,19-21} the changes they entail for trading behavior;²⁹ their integration with flat-based currencies and direct use for providing citizens with central bank money;^{8,36,43} alternative payment systems;^{5,42} implications for regulation and governance;^{16,26,49,50} and the privacy trade-offs cryptocurrencies and digital wallets introduce for consumers.²

From a business perspective, scholars have compared the transforma-

^b In a proof-of-work blockchain, the security of the public ledger depends on the amount of computing power that is dedicated to verifying and extending the log of transactions (that is, dedicated to “mining”).

^c The market capitalization is calculated as the number of tokens (approximately 16.8M bitcoin) times the value of each token (the Bitcoin to USD exchange rate was \$10,633 in January 2018; <https://coinmarketcap.com/> - accessed 01-22-2018).

tion brought about by blockchain to the introduction of communication protocols such as TCP/IP,^{24,25} and have explored applications to digital platforms beyond finance and implications for the boundaries of the firm.^{10,11}

Cost of Verification

Markets facilitate the voluntary exchange of goods and services between buyers and sellers. For an exchange to be executed, key attributes of a transaction need to be verified by the parties involved. When an exchange takes place in person the buyer can usually directly assess the quality of the goods, and the seller can verify the authenticity of the cash. The only intermediary involved in this scenario is the central bank issuing and backing the fiat currency used in the exchange. When a transaction is performed online instead, one or more financial intermediaries broker it by verifying, for example, that the buyer has sufficient funds. Intermediaries add value to marketplaces by reducing information asymmetry and the risk of moral hazard through third-party verification. This often involves imposing additional disclosures, monitoring participants, maintaining trustworthy reputation systems, and enforcing contractual clauses. As markets scale in size and geographic reach, verification services become more valuable, as most parties do not have preexisting relationships, but rely on intermediaries to ensure the safety of transactions and enforce contracts. In the extreme case where verification costs are prohibitively high, markets unravel, and beneficial trades do not take place.^e

In exchange for their services, intermediaries typically charge a fee. This is one of the costs buyers and sellers incur when they cannot efficiently verify all the relevant transaction attributes by themselves. Additional costs may stem from the intermediary having access to transaction data (a privacy risk) and being able to select which transactions to execute (a censorship risk).

e Over distance, intermediaries are key for verifying the quality of products or services, and reputation of buyers and sellers. High verification costs reduce market thickness³⁹ and prevent beneficial exchanges from taking place.



Blockchain technology can prevent information leakage by allowing market participants to verify transaction attributes and enforce contracts without exposing the underlying information to a third party.



These costs are exacerbated when intermediaries gain market power, often as a result of the informational advantage they develop over transacting parties through their intermediation services.⁴⁴ Transacting through an intermediary always involves some degree of disclosure to a third party, and increases the chance that the information will be later reused outside of the original contractual arrangement. Moreover, as an increasingly large share of economic and social activity is digitized, keeping data secure has become more problematic and information leakage more prevalent. Classic examples are the theft of social security numbers (for example, Equifax hack) and credit card data (for example, Target's data breach), or the licensing of customer data to advertisers. Blockchain technology can prevent information leakage by allowing market participants to verify transaction attributes and enforce contracts without exposing the underlying information to a third party.^f This allows an agent to verify that some piece of information is true (for example, good credit standing), without full access to all background information (for example, past transaction records): that is, the technology allows for the verification of transaction attributes in a privacy-preserving way.

Digitization has pushed verification costs for many types of transactions close to zero. When the relevant information is digital, blockchain technology contributes to this process by allowing for costless verification.^g Of course, at the interface between an offline record and its digital representation blockchain applications still face substantial frictions and “last mile” costs.⁴⁵ This explains why, despite claims by technology enthusiasts about the value of using the technology across a variety of applications including supply chain monitoring and digital identity, use cases outside of cryptocurrency and fintech

f This is achieved by combining a distributed ledger with zero-knowledge cryptography. Examples include cryptocurrencies such as Zcash and Zcoin.

g In practice, verification costs will never be exactly zero. What we mean by ‘costless’ is low enough to be irrelevant from an economic perspective relative to the value of the transaction.

(settings where key information and assets are digital) have been extremely limited. The link between online “on-chain” activities recorded on a blockchain and offline “off-chain” events introduces major challenges which cannot be overcome without complementary innovations. For example, a blockchain such as the Bitcoin one can be used to cheaply verify ownership and exchanges of its native digital asset. While this technically allows anyone to send and receive bitcoin globally without using an intermediary or being censored, actually being able to spend bitcoin to buy goods and services offline still runs into last mile issues. Hence, while Bitcoin has been used in countries with hyperinflation to escape devaluation, its use as a medium of exchange has been limited, and governments can still shape how these digital assets are used at the interface between the digital and the physical world. Similarly, information about identity is often used to increase the safety of market interactions, reduce fraud and build robust digital reputation systems, but being able to link an online action and digital record on a blockchain to an offline individual or entity is as expensive with blockchain technology as it would be with more traditional solutions. This drastically limits the benefits blockchain and smart contracts can bring in the absence of complementary technology (for example, a tamper-proof GPS sensor), firms and institutions that can help ensure the digital records are accurate to begin with.

The high-level process of verification is described in the accompanying figure: When a digital transaction is born, it immediately inherits some basic attributes, such as the fact that it exists and when it was created, information about the seller and buyer involved and their credentials, and so on. We typically rely on these attributes to perform subsequent actions (for example, once funds are transferred, the seller may ship the goods). Some of these actions take place every time (for example, settlement), whereas others are only triggered by specific events. A particularly interesting subset of future events are those that require additional verification. For example, a problem may emerge, and transaction attributes may need to be checked through an audit. The audit could range from actual auditors accessing the relevant logs or requesting additional information from market participants, to the execution of an internal process designed to handle the exception. Such processes tend to be costly, may involve labor and capital, and may require a third party to mediate between buyer and seller. The ideal outcome of an audit is the resolution of the problem that emerged.

Blockchain technology affects this flow by allowing, when a problem emerges, for the costless verification of digital information. Any transaction attribute or information on the agents and goods involved that is stored on a distributed ledger can be cheaply verified, in real time, by any

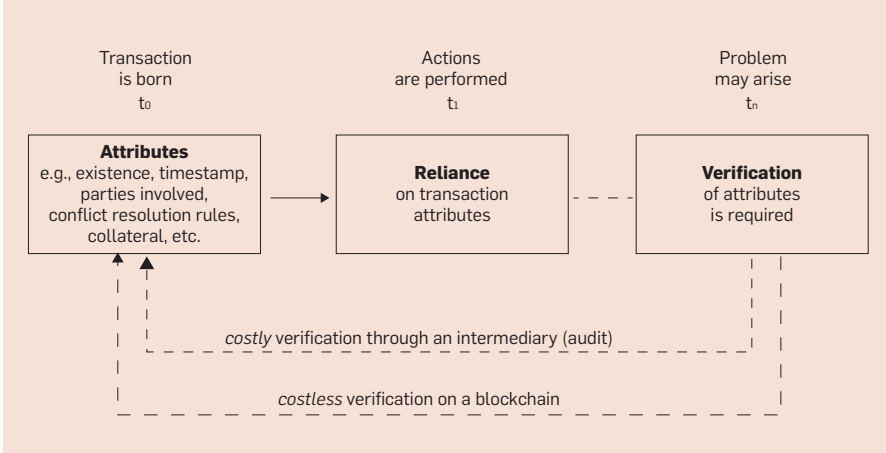
market participant. Trust in the intermediary is replaced with trust in the underlying code and consensus rules.^h These rules define how a distributed network reaches agreement, at regular intervals, about the true state of the shared data it needs to maintain to operate a well-functioning marketplace. At a minimum, such shared data can represent past transactions and outstanding balances in an underlying, cryptographic token (that is, it could be a snapshot of the ownership rights in the token). In more complex applications, the shared data can also cover the rules and data required to perform a specific operation (such as, to run an application, verify that a contract clause is enforced). These operations, often referred to as “smart contracts,”ⁱ can be automated in response to new events, adding flexibility to the verification process. For example, on a shared ledger used to exchange financial assets, transacting institutions can agree, ex-ante, on the rules for the settlement and reconciliation of trades, as well as on the process they will follow and third parties they will involve if an audit is necessary or a dispute emerges. Trusted, independent oracles can also be incorporated to ensure that such financial contracts can respond to market conditions and new information (for example, to implement a weather derivative, a smart contract can aggregate information across multiple weather sources to assess if a payout has to be made).

As with past improvements in information and communication technology, reductions in the cost of verification enable the unbundling of services that were previously offered together, as part of the steps traditionally performed by an intermediary can now be delivered through a shared ledger. This allows these steps to be collectively owned and

h If we think of the audit capability of a third party as surveillance or monitoring, blockchain technology can deliver “sousveillance,”³⁰ that is, an audit embedded within the marketplace.

i N. Szabo (1996): “The basic idea of smart contracts is that many kinds of contractual clauses [...] can be embedded in the hardware and software we deal with, in such a way as to make breach of contract expensive [...] for the breacher. A canonical real-life example [...] is the humble vending machine;” <https://bit.ly/2WZqMxM>


Costly verification through an intermediary (audit) versus costless digital verification on a blockchain.




managed by a broader group of ecosystem stakeholders, in a way that resembles collaboration among competitors and complementors in standard setting organizations,⁶ or open source foundations. The effects of this change have been mostly felt on the intensive margin of production (that is, on improving the efficiency of pre-existing use cases), as firms are experimenting with moving different types of transactions to blockchain-based systems to reduce settlement and reconciliation costs.

As a consequence, applications resulting from the reduction in the cost of verification have been complementary to incumbents, as they improve existing value-chains by lowering the cost of tracking ownership and trading digital assets without reducing the market power of existing players. Furthermore, even when verification can be automated, intermediaries can still add value and retain influence over a market by supporting regulatory compliance, market safety, handling edge cases (for example, a chargeback), and certifying information that requires labor-intensive, offline forms of verification. This explains why implementations of the technology targeted at identity and provenance have been slower to diffuse: While the verification of digital attributes can be cheaply implemented on a blockchain, the initial mapping between offline events and their digital representations is still costly to bootstrap and maintain. Therefore, as digital verification costs fall, key complements to it that can improve the process of offline verification become more valuable.

On one extreme, blockchain technology can be used to settle trades of digital assets that are completely self-contained within a shared ledger (for example, bitcoin, ether). The consensus rules established in the code define how tokens are created and earned, and how the network reaches agreement about the true state of ownership over time.^j The cost of verifying transaction attributes and enforcing



While the verification of digital attributes can be cheaply implemented on a blockchain, the initial mapping between offline events and their digital representations is still costly to bootstrap and maintain.



simple contracts for self-contained tokens can be extremely low. This is what allows for value to be transferred through Bitcoin across the globe at a relatively low cost. Of course, compliance with Know-Your-Customer (KYC) and Anti-Money-Laundering (AML) rules may require individuals and firms to sustain additional costs to credibly link their offline identities with their Bitcoin ones, but as long as individuals agree that the underlying token has value, using it as a store of value and medium of exchange is possible. Similarly, a native crypto token can be used to facilitate low-cost transactions of digital resources, such as computation (Ethereum), data storage (Filecoin), bandwidth, or to track equity ownership, electricity, as in all these cases verifying the exchange of a resource is not too expensive.

On the other extreme, when entries on a shared ledger are digital representations of offline identities, products, services and related transactions, costless verification is difficult to achieve. Under this scenario, the reduction in the cost of verification is contingent on maintaining a credible link between offline events and their online record. This link is cheaper to establish when offline attributes are easy to capture and expensive to alter or fake: for example, in the case of diamonds, Everledger uses the physical properties of the gems as a digital fingerprint that can be recorded and tracked on a blockchain as the products move through the supply chain. In many cases, maintaining a robust link between offline events and distributed ledgers is very expensive, and may require not only one or more trusted intermediaries, but also multiple parties to agree on rules for secure data entry and sharing. In the absence of a strong link between offline and online events, asymmetric information and moral hazard will be an issue in these markets. In this context, Internet of Things devices are instrumental in expanding the set of contracts that can be automated on a blockchain because they can be used to record real-world information (for example, through sensors and GPS devices) and substitute labor-intensive verification with inexpensive hardware.

Overall, when last-mile problems are limited—such as in the case of

^j Changes in the rules are implemented through a voting process similar to standard setting negotiations, and disagreements can lead to part of the network forking to launch a platform with different market design.

digital assets that are native to a blockchain—decentralized verification goes from being costly, scarce and prone to abuse, to being cheap and reliable. While this process is unlikely to be more efficient on a per transaction basis than verification through a centralized intermediary, the ability to perform it without trusting a third party can lead to savings from increased competition, the absence of centralized control, higher privacy and censorship resistance, and the removal of single points of failure. At the same time, when frictions between offline events and their digital representations are high, these improvements are unlikely to materialize in the absence of complementary innovations, as intermediaries will still be able to control key existing complements to digital verification and use them to exert influence over market participants.

As decentralized verification becomes cheaper, the scale at which it can be efficiently implemented also drops: On a distributed ledger, data integrity can be built, from the ground up, from the most basic transaction attributes to more complex ones. For example, a robust reputation system can be constructed from the full set of interactions an economic agent has throughout the economy, increasing transparency and accountability. Expensive audits and due diligence can be progressively substituted with more frequent and fine-grained verification to ensure market safety and reduce the risk of moral hazard. A lower cost of verification also makes it easier to define property rights at a more granular scale than before, as any digital asset (or small fraction of it) can be traded, exchanged or tracked at a low cost on a shared ledger.^k

Cost of Networking

The ability to verify state (for example, the current ownership status of a digital asset) at a lower cost because of the reduction in the cost of verification allows a blockchain protocol to not only reach consensus about the history and

^k In the same way that Twitter, because of the 140 character limitation, enabled new forms of communication, costless verification has the potential to change how information markets, digital property rights, and payments are designed.

A permissionless blockchain protocol allows a network of economic agents to agree, at regular intervals, on the true state of a set of shared data without assigning residual rights to trusted entities.

proposed evolution of a digital asset, but also to define rules for state transitions that are particularly valuable from a network perspective. These transitions can be used to reward participants for performing actions that accelerate adoption and increase network value and welfare. For example, the protocol can be used to incentivize behavior that builds network effects (both in terms of users and applications), ensures the network has sufficient resources available to meet demand, guarantees its security, encourages savings or spending behavior. Taken together, these incentives lower the cost of networking, that is, the cost of bootstrapping, operating and scaling an economic network.

Whereas a reduction in the cost of verification is a necessary condition for a reduction in the cost of networking—as it is the ability to verify state that allows economic agents to establish property rights on network resources and define incentives without relying on an intermediary—it is not a sufficient condition, as implementations can take advantage of the former without the latter. In particular, when a blockchain protocol is permissioned and the entities developing it retain control over which participants can update and verify state, transitions are not fully defined by code and self-contained within the system, but rather can be influenced by external parties through fiat. As a result, from an economics perspective, the network will operate under constraints similar to those of traditional digital platforms, and participants will have to trust the platform architect and core constituents through formal and relational contracts or past reputation, among others. This tension is an important one from an organizational perspective, as it determines if a blockchain network can be considered a novel organizational form versus not.¹²

A permissionless blockchain protocol, instead, allows a network of economic agents to agree, at regular intervals, on the true state of a set of shared data without assigning residual rights to trusted entities. The flexibility in terms of what such shared data represents across settings (for example, currency, intellectual property, and financial assets, contracts) makes it

a general-purpose technology (GPT). GPTs typically take a long time to diffuse through the economy, but also lead to productivity gains across multiple industries.^{9,22,33,37} Classic examples of GPTs include the steam engine, electricity, and the Internet. While permissionless networks have been compared to communication protocols such as TCP/IP—which focus on how information is packetized and routed through the Internet—they fundamentally differ from them because they allow for the secure provision, transfer and enforcement of property rights. On these networks, trust in a platform operator is replaced by trust in the underlying incentives, code and consensus rules. As a result, market power of the intermediary, privacy risk and censorship risk can be potentially reduced. The switch in the trust model also introduces new challenges, as bugs in the code can leave participants with little recourse beyond trying to coordinate a hard fork of the network. Issues with this new trust model have resulted from benign programming mistakes (such as the Parity wallet library removal),^l from deliberate attempts at defrauding investors by promising high returns in the absence of any real technical or business plan (as in the case of fraudulent initial coin offerings), as well as from malicious attacks (such as the DAO hack, which led to a split of the Ethereum network).^m Similarly, while blockchain protocols can be designed to offer participants a high degree of privacy (for example, Zk-Stark, Zcash, and Monero), and users can take additional measures to protect their privacy from the public (for example, using a mixing service, not reusing addresses), many shared ledgers such as the Bitcoin one are pseudonymous,ⁿ allowing third parties to deanonymize transactions and trace movements of funds over time.

Whereas permissioned networks only take advantage of the reduction in the cost of verification, permissionless ones build on the first by adding a self-

contained incentives system to also deliver a decrease in the cost of launching and operating a network without relying on trusted intermediaries. The effects of this reduction in the cost of networking are felt both in the phase of bootstrapping a new platform, and in the phase of operating it. In the first phase, a native token can be used to create incentives for adoption and to fund the development and scaling of the network, for example by having mining rewards or by raising capital through an initial coin offering (ICO). In the second phase, market design is used to define the conditions under which participants can earn tokens for contributing resources to the network (for example, computing power in the case of Bitcoin, computing and applications for Ether, disk storage for Filecoin, digital content and advertising in the case of the Basic Attention Token).

Since during the bootstrapping phase the actual utility the network can deliver to users is limited by its small scale, and network effects work against users switching from existing alternatives, this phase relies on contributions from early adopters and investors with positive expectations about the future value of the network. As in open source projects,^{47,48} early adopters may be willing to dedicate time and effort to support a new network because they want to create a viable alternative to established products or they derive utility from advancing the underlying technology (for example, consumption utility from early access, from working on novel, complex problems, job-market signaling). Investors, instead, as in traditional equity finance, may come in early because they expect the token to appreciate in value and reward their investment.¹⁴ Of course, individuals can be simultaneously early adopters and investors and contribute both effort and capital to these projects. For this set of individuals, the presence of a native token serves a similar purpose to founder and early-employee equity in startups and allows these projects to attract talent without raising investment from traditional angels and venture capitalists. Since it only takes a few lines of code to write a smart contract for an initial coin offering, open source codebases can be forked or imitated at a

low cost, and regulation is still uncertain in many jurisdictions, the ability to profit from launching a new cryptocurrency or manipulating its trading have attracted a large number of bad actors and speculators.

While lower entry barriers and the presence of technical investors could in theory open up capital for new types of entrepreneurs and ideas that traditional investors may be more reluctant to fund, the absence of regulation and oversight also allows fraudulent projects to blend in with legitimate ones and raise capital from unsophisticated investors. Combined with the fact that the value of a new token is, in most cases, purely based on expectations about its future success, and that such expectations, because of technical, regulatory and market uncertainty can rapidly turn when new information emerges or sentiment evolves, the valuations of cryptocurrencies have been extremely volatile. The resulting turmoil and speculative bubbles have made it more difficult for investors to identify high-quality projects and teams, have attracted speculators and low-quality entrants and have shifted attention from technology R&D to short-term speculative returns.

If in the first phase of growth of a blockchain-based network, incentives are predominantly targeted at accelerating adoption, in the second phase the key challenges from a market design perspective are ensuring that the incentives continue to support contributions of key resources to the ecosystem and avoiding a tragedy of the commons. By design, the protocol layer is a shared resource among all network participants, and everyone benefits from investments in it—from better security to removing technical constraints on throughput, latency or liveness. At the same time, because of the public good nature of these improvements, in the absence of proper governance, a blockchain-based network may fail to invest enough resources on them. From a valuation perspective, whereas the bootstrapping phase of a new token is associated with extremely high volatility, as uncertainty around a network's potential is resolved, it should enter a more stable growth trajectory.^o

^o This is similar to the process of early-stage startup funding and growth.

^l See <https://bit.ly/2Uyv3GP>

^m See <https://www.bloomberg.com/features/2017-the-ether-thief/>

ⁿ Like a writer writing under a pseudonym, if a Bitcoin user is ever tied to an address, the history of her transactions can be read on the blockchain.

Overall, relative to blockchain implementations that only take advantage of the reduction in the cost of verification (for example, permissioned networks), those that also benefit from the reduction in the cost of networking (for example, permissionless ones) are different on at least four dimensions. First, they are less likely to leave market power in the hands of their founders or early participants. This limits the ability of any party to unilaterally censor transactions or exclude participants from the network, and removes single points of failure, as the network does not depend on the availability of one or a few key players to operate.^p

Second, they are less reliant on off-chain governance, relational contracts and laws to support their operations, as by design, to take advantage of the lower cost of networking they need to embed as much as possible of the incentives and governance rules required for their operations into the protocol. Of course, permissionless networks still need off-chain governance and coordination between their key stakeholders to execute a hard fork, implement controversial changes, or respond to an attack, but relative to more closed networks that rely on trusted intermediaries they leave less discretion to any single party, and end up codifying more of their rules into their codebases.

Third, they involve a lower privacy risk, as no single entity (or group of entities) has preferential access to or visibility over the information generated by the network.^q In traditional platforms, the privacy risk is particularly salient in markets where consumers pay for services by allowing intermediaries to access and monetize their data, an issue that is increasingly relevant because of the role such data can play in the training of AI algorithms.¹ Whereas the trend of consumers relinquishing private information in ex-

change for free or subsidized digital services is unlikely to change because of blockchain technology—as small incentives and frictions can be used by digital platforms to persuade even privacy sensitive individuals to relinquish sensitive information²—startups in this space are experimenting with approaches that give users greater control over how, when and why their private data is accessed and monetized.

Fourth, blockchain implementations that take advantage of the lower cost of networking inevitably induce architectural changes in how firms create and capture value within markets. Architectural innovations, by destroying the usefulness of the assets and accumulated knowledge of incumbents,²³ open opportunities for entrants to reshape the dimensions firms compete on, and experiment with new business models. In particular, by allowing for the separation of some of the benefits of network effects from the costs of market power—since even in the absence of a platform architect participants in a blockchain network are able to rely on shared infrastructure—the technology offers new ways to reward contributors, allocate rents in a marketplace, and build applications on top of shared data while preserving the privacy of the underlying information. In traditional digital marketplaces, platform operators have wide visibility over all interactions that take place on their networks, and users are unable to directly custody or control the digital assets they use or create while transacting on them. This is a direct result of the inability of these systems to generate and trade scarce, digital assets and establish digital property rights without also assigning control over them to a third-party (usually the platform operator). Before Bitcoin, for example, a central clearing house of some type was necessary to prevent the copying and double spending of digital cash. Bitcoin solves this problem by allowing users to self-custody digital tokens and exchange them without relinquishing control over them to a third-party. This reduces switching costs between digital wallets and offers users a higher degree of privacy from service providers. Interestingly, while blockchain technology provides individuals and organizations with the opportunity to self-custody

and exchange digital assets without the need for traditional intermediaries such as banks, significant work is needed before users can reap the full benefits of this change—such as greater privacy, higher portability between service providers, and increased competition—as many implementations lack the convenience and usability of the centralized solutions consumers are used to. For example, while Bitcoin users can store and protect their own private keys, a large number of them rely on third-party wallets to do so, essentially trusting these entities with their funds as in traditional systems.

Conclusion

The article focuses on two key costs affected by blockchain technology: the cost of verification, and the cost of networking. For markets to thrive, participants must be able to efficiently verify and audit transaction attributes, including, for example, the credentials and reputation of the parties involved, characteristics of assets exchanged, and external events and information that have implications for contractual arrangements.

Outside the boundaries of an organization, this is typically achieved by relying on trusted intermediaries. In exchange for their services, intermediaries charge fees and capitalize on their ability to observe all transactions taking place within their marketplaces. This informational advantage, combined with network effects and economies of scale, gives them substantial market power and control over market participants. Consequences of market power include higher prices, user lock-in and high switching costs, the presence of single points of failure, censorship risk, barriers to innovation, and reduced privacy.

Blockchain technology, by reducing the costs of running decentralized networks of exchange, allows for the creation of ecosystems where the benefits from network effects and shared digital infrastructure do not come at the cost of increased market power and data access by platform operators. This reduction in the cost of networking has profound consequences for market structure, as it allows open source projects and startups to directly compete with entrenched incumbents through

p The censorship risk is visible when an intermediary revokes or degrades access to a participant, and when it loses control over the marketplace because of an attack or technical failure. All three cases have been observed in online platforms, which are concentrated markets because of network effects and economies of scale in data collection, storage, and processing.

q Privacy may still be a concern if a public ledger exposes information about participants and their transactions.²


the design of platforms where the rents from direct and indirect network effects are shared more widely among participants (for example, users, application developers, and investors), and no single entity has full control over the underlying digital assets.

Because of the absence of a central clearing house or market maker, these novel networks, when permissionless, exhibit low barriers to entry and innovation. As long as applications are compatible with the rules of the protocol, they can be deployed without permission from other participants, and compete for market share. This reduces the expropriation risk application developers face when building on top of traditional digital platforms. Furthermore, since contributors can participate in governance in a way that is often proportional to their stake in the system, these networks can democratically evolve over time to accommodate changes that are beneficial to the majority of their constituents.^r


From a talent acquisition perspective, unlike open source projects, the digital platforms built on top of crypto tokens do not have to rely solely on pro-social contributions of time and labor and job market signaling²⁷ to support their development. Using a native token, they can directly incentivize early contributions by developers, investors and early adopters. This novel source of funding combines crowdfunding with the simultaneous crowdsourcing of key resources needed to scale a platform and attract both developer and user activity on to it. Because of the reduction in the cost of verification, this model also allows for equity in the system to be defined at a much narrower scale, and to be allocated to a wider population of participants in response to verifiable contributions of resources.

Similarly, by allowing for the definition of scarce digital property rights, native tokens allow decentralized networks of exchange to coordinate activity around shared objectives and

r Minorities that disagree with a change face reduced lock-in because they can fork and launch a backward-compatible platform. At the same time, since forks introduce uncertainty and may decrease overall value, off-chain governance is needed to support fundamental changes in market design.



These changes allow for the design of novel types of networks that blend features of competitive markets with the more nuanced forms of governance used within vertical integrated firms and online platforms.



transact digital resources without assigning market power to a market maker. Through blockchain-based networks, individuals and organizations can source ideas, information, capital and labor, and enforce contracts for digital assets with substantially reduced frictions. These changes allow for the design of novel types of networks that blend features of competitive markets with the more nuanced forms of governance used within vertically integrated firms and online platforms.^s

Whereas intermediaries will still be able to add substantial value to transactions by focusing on tasks that are complementary to digital verification (for example, secure recording of offline events, curation, and certification of identity and services), they are likely to face increased competition because of the ability to establish and exchange digital assets on decentralized open networks without them.^t This challenges some of their revenue sources and reduces their influence over markets, opening up opportunities for new business models and novel approaches to data privacy, ownership and portability, as well as to the regulation of networks that should be considered public utilities. By reducing barriers to entry within sectors that are currently heavily concentrated because of network effects and control over data, the technology may enable a new wave of innovation in digital services, and greater consumer choice.

For these changes to materialize, however, substantial hurdles will have to be overcome. First, the technology will need to reach a level of performance (for example, throughput, latency, and cost per transaction) comparable to traditional networks. While decentralization inevitably comes at a cost, the gains from greater competition, openness, privacy and censorship resistance will have to outweigh the lower efficiency of blockchain networks to make adoption worthwhile. Hybrid networks that

s For example, the hedge fund Numerai uses smart contracts to reward contributions to its financial prediction model by a distributed community of data scientists.

t Beyond financial applications, early applications that may be affected by these changes are those that involve the exchange of digital content, media, and new types of digital assets and goods.

embrace key features of permissionless systems—such as low barriers to entry and a competitive market for resources and applications—while initially borrowing trust from existing institutions to overcome scaling problems, may also provide a viable transition path when performance is an obstacle to adoption.

Second, regulatory frameworks will have to evolve to reduce uncertainty for founders and network participants, and to provide stronger protections for investors and early adopters. Because of their similarities but also their differences with equity,¹⁴ crypto tokens lend themselves to both legitimate fundraising activity by high quality entrepreneurs, as well as fragrant abuse by fraudsters.¹³ As in other technological bubbles, this constitutes a challenge for the space, as investors have a difficult time separating projects worth supporting from the much larger number of low-quality imitators, and entry by speculators has brought extreme price volatility and additional risks to the market.

Third, and possibly most important, blockchain technology, like other technological advancements, is not a panacea for every possible technical and market challenge a digital ecosystem may face. As discussed throughout this article, the technology can add substantial value under fairly narrow conditions: 1) when last mile problems are not severe and digital verification can be implemented in a novel or more fine-grained way because of a reduction in the cost of verifying state without assigning control to an intermediary; 2) when the reduction in the cost of networking allows participants to allocate rents from a digital platform more efficiently between users, developers, and investors; 3) when the combination of a reduction in both costs (verification and networking) allows for the definition of new types of digital assets and property rights; 4) when there is a need for greater privacy and ability for users to control when and how their data is accessed and used. When none of these conditions are met instead, more centralized solutions that rely on traditional intermediaries and relational contracts are unlikely to be replaced, as the benefits of transitioning to a blockchain-based system are unlikely to counterbalance the costs introduced by a decentralized

infrastructure and governance, and the replication of state across the network.

Acknowledgments

We are thankful to Al Roth, Muneeb Ali, Naval Ravikant, Nicola Greco, Tim Simcoe, Scott Stern, Catherine Tucker, and Jane Wu for helpful discussions. ■

References

- Agrawal, A., Gans, J., and Goldfarb, A. The simple economics of machine intelligence. *Harvard Business Rev.*, (2016), 17.
- Athey, S., Catalini, C., and Tucker, C. The Digital Privacy Paradox: Small Money, Small Costs, Small Talk. National Bureau of Economic Research Working Paper, 2017.
- Athey, S., Parashkevov, I., Sarukkai, V., and Xia, J. Bitcoin pricing, adoption, and usage: Theory and evidence. Research paper, Stanford University, 2016.
- Ausubel, L.M. et al. The lovely but lonely Vickrey auction. *Combinatorial Auctions 17* (2006), 22–26.
- Beck, R., Czepluch, J. S., Lollike, N., and Malone, S. Blockchain—The gateway to trust-free cryptographic transactions. *ECIS*, 2016, Paper 153.
- Bekkers, R., Catalini, C., Martinelli, A., Righi, C., and Simcoe, T. Disclosure rules and declared essential patents. Discussion paper, National Bureau of Economic Research, 2019.
- Böhme, R., Christin, N., Edelman, B., and Moore, T. Bitcoin: Economics, technology, and governance. *J. Economic Perspectives* 29, 2 (2015), 213–38.
- Bordo, M.D. and Levin, A.T. Central Bank Digital Currency and the Future of Monetary Policy. National Bureau of Economic Research Working Paper, 2017.
- Bresnahan, T.F. and Trajtenberg, M. General-purpose technologies—Engines of growth? *J. Econometrics* 65, 1 (1995), 83–108.
- Catalini, C. How blockchain applications will move beyond finance. *Harvard Business Rev.*, (2017).
- Catalina, C. How blockchain technology will impact the digital economy. *Oxford Business Law Blog*, (2017).
- Catalini, C. and Bostlego, J. Blockchain Technology and Organization Science: Decentralization Theatre or Novel Organizational Form? MIT Working Paper, 2019.
- Catalini, C., Bostlego, J. and Zhang, K. Technological Opportunity, Bubbles and Innovation: The Dynamics of Initial Coin Offerings. Working Paper, 2018.
- Catalini, C. and Gans, J. S. Initial coin offerings and the value of crypto Tokens. Discussion paper. National Bureau of Economic Research, 2018.
- Catalini, C. and Tucker, C. When early adopters don't adopt. *Science*, 357, 6347 (2017), 135–136.
- Davidson, S., De Filippi, P. and Potts, J. Economics of blockchain. Working Paper, 2016.
- Dwyer, G.P. The economics of Bitcoin and similar private digital currencies. *J. Financial Stability* 17 (2015), 81–91.
- Edelman, B., Ostrovsky, M. and Schwarz, M. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *Amer. Economic Rev.* 97, 1 (2007), 242–259.
- Gandal, N. and Halaburda, H. Competition in the Cryptocurrency Market. NET Institute Working Paper, 2014.
- Gans, J.S. and Halaburda, H. Some economics of private digital currency. *Economic Analysis of the Digital Economy*, (2015), 257–276. University of Chicago Press.
- Halaburda, H. and Sarvary, M. *Beyond Bitcoin: The Economics of Digital Currencies*. Springer, 2016.
- Helpman, E. *General Purpose Technologies and Economic Growth*. MIT Press, Cambridge, MA, 1998.
- Henderson, R.M. and Clark, K.B. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Q.* (1990), 9–30.
- Iansiti, M. and Lakhani, K.R. The truth about blockchain. *Harvard Business Rev.* 95, 1 (2017), 118–127.
- Ito, J., Narula, N. and Ali, R. The blockchain will do to the financial system what the Internet did to media. *Harvard Business Rev.* (2017)
- Kiviat, T.I. Beyond bitcoin: Issues in regulating blockchain transactions. *Duke LJ* 65, 569 (2015).
- Lerner, J. and Tirole, J. Some simple economics of open source. *J. Industrial Economics* 50, 2 (2002), 197–234.
- Luca, M. Designing online marketplaces: Trust and

- reputation mechanisms. *Innovation Policy and the Economy* 17, 1 (2017), 77–93.
- Malinova, K. and Park, A. Market Design with Blockchain Technology. Working Paper, 2016.
 - Mann, S., Nolan, J. and Wellman, B. Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments. *Surveillance & Society* 1, 3 (2002), 331–355.
 - Milgrom, P.R. *Putting auction theory to work*. Cambridge University Press, 2004.
 - Morton, F.S. Consumer benefit from use of the Internet. *Innovation Policy and the Economy* 6 (2006), 67–90.
 - Moser, P. and Nicholas, T. Was electricity a general-purpose technology? Evidence from historical patent citations. *Amer. Economic Rev.* 94, 2 (2004), 388–394.
 - Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. White Paper, 2008.
 - Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.
 - Raskin, M., and Yermack, D. Digital currencies, decentralized ledgers, and the future of central banking. National Bureau of Economic Research Working Paper, 2016.
 - Rosenberg, N. and Trajtenberg, M. A general-purpose technology at work: The Cortiss steam engine in the late 19th century US. National Bureau of Economic Research Working Paper, 2001.
 - Roth, A.E. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica* 70, 4 (2002), 1341–1378.
 - Roth, A.E. The art of designing markets. *Harvard Business Rev.* 85, 10 (2007), 118.
 - Roth, A.E., and Ockenfels, A. Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the Internet. *Amer. Economic Rev.* 92, 4 (2002), 1093–1103.
 - Rothkopf, M.H., Teisberg, T.J., and Kahn, E.P. Why are Vickrey auctions rare? *J. Political Economy* 98, 1 (1990), 94–109.
 - Rysman, M., and Schuh, S. New innovations in payments. *Innovation Policy and the Economy* 17, 1 (2017), 27–48.
 - Seretakis, A. Blockchain, Securities Markets and Central Banking. Working Paper, 2017.
 - Stiglitz, J.E. Information and the change in the paradigm in economics. *Amer. Economic Rev.* 92, 3 (2002), 460–501.
 - Tucker, C. and Catalini, C. What blockchain can't do. *Harvard Business Rev.* (2018).
 - Von Hippel, E. *Democratizing Innovation*. MIT Press, 2005.
 - Von Hippel, E.A. Open source projects as horizontal innovation networks—by and for users. 2002.
 - Von Hippel, E. and Von Krogh, G. Open source software and the private collective innovation model: Issues for organization science. *Organization Science* 14, 2 (2003), 209–223.
 - Walport, M. Distributed ledger technology: Beyond block chain. U.K. Government Office for Science, 2016.
 - Wright, A. and De Filippi, P. Decentralized blockchain technology and the rise of lex cryptographia. 2015.

Christian Catalini (catalini@mit.edu) is the Theodore T. Miller Career Development Professor at MIT, associate professor of Technological Innovation, Entrepreneurship, and Strategic Management at the MIT Sloan School of Management, and founder of MIT Cryptoeconomics Lab, Cambridge, MA, USA.

Joshua S. Gans (joshua.gans@rotman.utoronto.ca) is a professor of strategic management and holder of the Jeffrey Skoll Chair in Technical Innovation and Entrepreneurship at the Rotman School of Management, University of Toronto, CA.

Copyright held by authors/owners.



Watch the authors discuss this work in this exclusive *Communications* video. <https://cacm.acm.org/videos/economics-of-the-blockchain>

research highlights

P. 92

Technical Perspective Why ‘Correct’ Computers Can Leak Your Information

By Mark D. Hill

P. 93

Spectre Attacks: Exploiting Speculative Execution

By Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom

P. 102

Technical Perspective ASIC Clouds: Specializing the Datacenter

By Parthasarathy Ranganathan

P. 103

ASIC Clouds: Specializing the Datacenter for Planet-Scale Applications

By Michael Bedford Taylor, Luis Vega, Moein Khazraee, Ikuo Magaki, Scott Davidson, and Dustin Richmond

Technical Perspective

Why ‘Correct’ Computers Can Leak Your Information

By Mark D. Hill

INFORMATION SECURITY IS important, as much of life’s private information is now stored on shared computers accessible from anywhere in the world. Many attacks begin by exploiting flaws in a system’s implementation (bugs) or specification. Most exploited flaws today are in software, as software presents a large attack surface. While much rarer, hardware flaws can cause even correct software to leak information and fixing can even require new hardware.

As the complexity of modern systems has grown, we have become dependent on abstraction to manage it, and yet this gives rise to subtle classes of flaws when the assumptions that underpin these abstractions are violated. Abstractions in the logical systems can be perfect: once matrix properties are proven, they apply to all arrays of numbers. Abstractions of the physical world are approximations or models. For example, while light is neither a particle nor a wave, both are useful models.

A useful abstraction in computer science is the *instruction set architecture* (ISA) that separates software and hardware. Today, only a few commercially successful ISAs separate many millions of lines of software from scores of hardware implementations. Moreover, since IBM System/360 in 1964, these ISAs are specified as the *timing-independent functional behavior of “instructions”* that are somewhat primitive (for example, branches, loads, and adds) and where each processing core logically executes instructions sequentially.

In 2018, Spectre—detailed in the paper that follows—demonstrated that a computer the CS community believed to be correct—its implementation follows its ISA—could rapidly leak information to a malicious adversary. The public revelation began with three Spectre variants—including Meltdown—in January,^{2,3,4} expanded to a dozen by October,¹ and has continued to grow since then. Spectre is possible because the

Spectre is possible because the ISA—like any abstraction of the physical world—is imperfect.

ISA—like any abstraction of the physical world—is imperfect. In particular, the *timing-independent* ISA is implemented with a supposedly hidden microarchitecture that is *all about timing*: its purpose is to make a computer as fast as possible within cost constraints.

Spectre shows that current ISAs—call them Architecture 1.0—are inadequate to protect information. Spectre exploits two key micro-architecture techniques:


Instruction speculation. A processor core seeks to execute dozens of instructions concurrently by speculating past branches, committing ISA changes if speculation is correct and rolling them back when speculation is wrong. Perversely, Spectre speculatively executes instructions whose ISA changes it knows will be rolled back. Its subtle goal is to leave microarchitectural “breadcrumbs” of a supposedly hidden secret.

Caching. Each processor core uses a hierarchy of caches to make memory accesses 100X faster than DRAM memory. Like a hash table, each cache keeps data in buckets called sets to aid lookup. Caches are invisible to the ISA so their sets don’t need to be restored on incorrect speculation. Spectre exploits this to place, and later find, “breadcrumbs” that reveal a secret. It thus uses the contents of a cache as a “side channel” to transmit a (secret) data value. Microarchitecture structures beyond caches have also been exploited.³

There has been some progress addressing Spectre since it was publicly

released.¹ Software fixes include adding memory fences (that may retard speculation), placing secrets in separate address spaces, selectively flushing caches, and converting indirect branches into pseudo-returns. These changes can hurt performance and can rely on undocumented chip implementation features. Hardware fixes disable features or patch microcode of current chips—where possible—or await changes deployed in new chips.

I recommend the following paper as a much gentler introduction to Spectre than the original paper.³ It excellently reviews how speculative execution and caches can be exploited, presents specific exploits using speculative branches that are direct (Variant 1) and indirect (Variant 2), touches on other variants, and concludes discussing software and hardware options for mitigating Spectre.

In the long run, do we manage or eliminate Spectre? We can manage it by working to discover and patch variants as they arise, much as society manages crime. More boldly, I assert we should seek to eliminate Spectre by defining an Architecture 2.0 that can be refined into implementations with provable properties regarding information exfiltration, including via microarchitecture timing. While this is hard (or even not completely possible), it is important as society depends on public computer systems to store our private information. 

References

- Hill, M.D., Masters, J., Ranganathan, P., Turner, P., and Hennessy, J.L. On the Spectre and Meltdown processor security vulnerabilities. *IEEE Micro* 39, 2 (2019), 9–19.
- Horn, J. Reading privileged memory with a side-channel. Project Zero; <https://bit.ly/35Ujuvii>.
- Kocher, P. et al. Spectre attacks: Exploiting speculative execution; arXiv preprint arXiv:1801.01203.
- Lipp, M. et al. Meltdown: Reading kernel memory from user space. In *Proceedings of the 27th USENIX Security Symp.* USENIX Association, 2018.

Mark D. Hill is John P. Morgridge Professor and Gene M. Amdahl Professor of Computer Sciences at the University of Wisconsin, Madison, WI, USA.

Copyright held by author.

Spectre Attacks: Exploiting Speculative Execution

By Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom

Abstract

Modern processors use branch prediction and speculative execution to maximize performance. For example, if the destination of a branch depends on a memory value that is in the process of being read, CPUs will try to guess the destination and attempt to execute ahead. When the memory value finally arrives, the CPU either discards or commits the speculative computation. Speculative logic is unfaithful in how it executes, can access the victim's memory and registers, and can perform operations with measurable side effects.

Spectre attacks involve inducing a victim to speculatively perform operations that would not occur during correct program execution and which leak the victim's confidential information via a side channel to the adversary. This paper describes practical attacks that combine methodology from side-channel attacks, fault attacks, and return-oriented programming that can read arbitrary memory from the victim's process. More broadly, the paper shows that speculative execution implementations violate the security assumptions underpinning numerous software security mechanisms, such as operating system process separation, containerization, just-in-time (JIT) compilation, and countermeasures to cache timing and side-channel attacks. These attacks represent a serious threat to actual systems because vulnerable speculative execution capabilities are found in microprocessors from Intel, AMD, and ARM that are used in billions of devices.

Although makeshift processor-specific countermeasures are possible in some cases, sound solutions will require fixes to processor designs as well as updates to instruction set architectures (ISAs) to give hardware architects and software developers a common understanding as to what computation state CPU implementations are (and are not) permitted to leak.

1. INTRODUCTION

Computations performed by physical devices often leave observable side effects beyond the computation's nominal outputs. Side-channel attacks focus on exploiting these side effects to extract otherwise-unavailable secret information. Since their introduction in the late 90s,¹⁴ various physical effects such as power consumption have been leveraged to extract cryptographic keys as well as other secrets.¹³

External side-channel measurements can be used to extract secret information from complex devices such as PCs and mobile phones. However, because these devices often execute code from a potentially unknown origin, they face additional threats in the form of software-based

attacks, which do not require external measurement equipment. Although some attacks exploit software logic errors, other software attacks leverage hardware properties to infer sensitive information. Attacks of the latter type include microarchitectural attacks exploiting cache timing^{3, 6, 17} and branch prediction history.¹ Software-based techniques have also been used to induce computation errors, such as fault attacks that alter physical memory¹¹ or internal CPU values.²⁵

Several microarchitectural design techniques have facilitated the increase in processor speed over the past decades. One such advancement is speculative execution, which is widely used to increase performance and involves having the CPU guess likely future execution directions and prematurely execute instructions on these paths. More specifically, consider an example where the program's control flow depends on an uncached value located in external physical memory. As this memory is much slower than the CPU, it often takes several hundred clock cycles before the value becomes known. Rather than wasting these cycles by idling, the CPU attempts to guess the direction of control flow, saves a checkpoint of its register state, and proceeds to speculatively execute the program on the guessed path. When the value eventually arrives from memory, the CPU checks the correctness of its initial guess. If the guess was wrong, the CPU discards the incorrect speculative execution by reverting the register state back to the stored checkpoint, resulting in performance comparable to idling. However, if the guess was correct, the speculative execution results are committed, yielding a significant performance gain as useful work was accomplished during the delay.

From a security perspective, speculative execution involves executing a program in possibly incorrect ways. However, because CPUs are designed to maintain functional correctness by reverting the results of incorrect speculative executions to their prior states, these errors were previously assumed to be safe.

In this paper, we analyze the security implications of such incorrect speculative execution. We present a class of microarchitectural attacks which we call *Spectre attacks*. At a high level, Spectre attacks trick the processor into speculatively executing instruction sequences that should not have been executed under correct program execution. As the effects of these instructions on the nominal CPU state are eventually reverted, we call them *transient instructions*. Transient

The original version of this paper appeared in *Proceedings of the 40th IEEE Symposium on Security and Privacy* (May 2019).

instructions can, however, have observable effects that convey information. By influencing which transient instructions are speculatively executed, we are able to leak information from within the victim's memory address space.

Spectre attacks can be applied to leak information across a broad range of security domains. In this paper, we describe several implementations and variations, such as attacks that extract information from other processes and from kernel memory, and that violate sandboxes enforced by programming languages.

At a high level, Spectre attacks violate memory isolation boundaries by combining speculative execution with data exfiltration via microarchitectural covert channels. More specifically, to mount a Spectre attack, an attacker starts by locating or introducing a sequence of instructions within the process address space which, when executed, acts as a covert channel transmitter that leaks the victim's memory or register contents. The attacker then tricks the CPU into speculatively and erroneously executing this instruction sequence, thereby leaking the victim's information over the covert channel. Finally, the attacker retrieves the victim's information over the covert channel. Although the changes to the nominal CPU state resulting from this erroneous speculative execution are eventually reverted, previously leaked information or changes to other microarchitectural states of the CPU, for example, cache contents, can survive nominal state reversion.

The above description of Spectre attacks is general and needs to be concretely instantiated with a way to induce erroneous speculative execution as well as with a microarchitectural covert channel. Although many choices are possible for the covert channel component, the implementations described in this work use cache-based covert channels,²⁴ that is, Flush+Reload²⁹ and Evict+Reload.^{5,15}

The underlying vulnerability arises from the composition of widely used microarchitectural features, rather than an implementation error in a single component. We have verified the vulnerability in all processors tested that implement speculative execution, such as multiple designs from Intel, AMD, and ARM. This contrasts with a related issue, Meltdown,¹⁶ which exploits a vulnerability specific to many Intel and a few ARM processors, which allows user-mode instructions to infer the contents of kernel memory.

Following the practice of responsible disclosure, we participated in an embargo of the results. This process was unusually complex due to the large number of stakeholders and affected products.

2. BACKGROUND

In this section, we introduce some of the microarchitectural components of modern high-speed processors as well as several attack techniques.

2.1. Speculative execution

Often, the processor does not know the future instruction stream of a program. For example, this occurs when out-of-order execution reaches a conditional branch instruction whose direction depends on preceding instructions whose execution is not completed yet. In such cases, the processor

can preserve its current register state, make a prediction as to the path that the program will follow, and *speculatively* execute instructions along the path. If the prediction turns out to be correct, the results of the speculative execution are committed (i.e., saved), yielding a performance advantage over idling during the wait. Otherwise, when the processor determines that it followed the wrong path, it abandons the work it performed speculatively by reverting its register state and resuming along the correct path.

We refer to instructions which are performed erroneously (i.e., as the result of a misprediction), but may leave microarchitectural traces, as *transient instructions*. Although the speculative execution maintains the architectural state of the program as if execution followed the correct path, microarchitectural elements may be in a different (but valid) state than before the transient execution.

Speculative execution on modern CPUs can run several hundred instructions ahead.

2.2. Branch prediction

During speculative execution, the processor makes guesses as to the likely outcome of branch instructions. Better predictions improve performance by increasing the number of speculatively executed operations that can be successfully committed.

Branch predictors of modern processors can have multiple prediction mechanisms for direct and indirect branches. Indirect branch instructions can jump to arbitrary target addresses computed at runtime, such as instructions that jump to an address in a register, memory location, or on the stack (e.g., “`jmp eax`” on x86). Return instructions are a type of indirect branch, and modern CPUs often include additional mechanisms for predicting return addresses.

For conditional branches, recording the target address is not necessary for predicting the outcome of the branch, because the destination is typically encoded in the instruction although the condition is determined at runtime. To improve predictions, the processor maintains a record of branch outcomes, both for recent direct and indirect branches.

2.3. The memory hierarchy

To bridge the speed gap between the faster processor and the slower memory, processors use a hierarchy of successively smaller but faster caches. The caches divide the memory into fixed size chunks called *lines*, with typical line sizes being 64 or 128 bytes. When the processor needs data from memory, it first checks if the *L1* cache contains a copy. In the case of a *cache hit*, that is, the data is found in the cache, the data is retrieved from the *L1* cache and used. Otherwise, in the case of a *cache miss*, the procedure is repeated to attempt to retrieve the data from the next cache levels, and finally the external memory. Once a read is completed, the data is typically stored in the cache (and a previously cached value is evicted to make room) in case it is needed again in the near future.

2.4. Microarchitectural side-channel attacks

The microarchitectural components discussed above improve the processor performance by predicting future program behavior. To that aim, they maintain state that

depends on past program behavior and assume that future behavior is similar to or related to past behavior.

When multiple programs execute on the same hardware, either concurrently or via time-sharing, changes in the microarchitectural state caused by the behavior of one program may affect other programs. This, in turn, may result in unintended information leaks from one program to another.

Initial microarchitectural side-channel attacks exploited timing variability¹⁴ and leakage through the L1 data cache²⁷ to extract keys from cryptographic primitives. Over the years, channels have been demonstrated over multiple microarchitectural components, such as lower level caches^{10, 17} and branch history.¹

In this work, we use the Flush+Reload technique,^{6, 29} and its variant Evict+Reload.⁵ Using these techniques, the attacker begins by evicting a cache line from the cache that is shared with the victim. After the victim executes for a while, the attacker measures the time it takes to perform a memory read at the address corresponding to the evicted cache line. If the victim accessed the monitored cache line, the data will be in the cache, and the access will be fast. Otherwise, if the victim has not accessed the line, the read will be slow. Hence, by measuring the access time, the attacker learns whether the victim accessed the monitored cache line between the eviction and probing steps.

The main difference between the two techniques is the mechanism used for evicting the monitored cache line from the cache. In the Flush+Reload technique, the attacker uses a dedicated machine instruction, for example, x86's `clflush`, to evict the line. Using Evict+Reload, eviction is achieved by forcing contention on the cache set that stores the line. Due to the limited size of the cache, reading several other memory locations that map to the same cache set can cause the processor to discard (evict) the desired line.

3. ATTACK OVERVIEW

Spectre attacks induce a victim to speculatively perform operations that would not occur during strictly serialized in-order processing of the program's instructions, and that leak victim's confidential information via a covert channel to the adversary.

In most cases, the attack begins with a setup phase, where the adversary performs operations that mistrain the processor so that it will later make an exploitably erroneous speculative prediction. In addition, the setup phase may include steps that help induce speculative execution, such as manipulating the cache state to remove data that the processor will need to determine the actual control flow. During the setup phase, the adversary can also prepare the covert channel that will be used for extracting the victim's information, for example, by performing the flush or evict part of a Flush+Reload or Evict+Reload attack.

During the second phase, the processor speculatively executes instruction(s) that transfer confidential information from the victim context into a microarchitectural covert channel. This may be triggered by having the attacker request that the victim performs an action, for example, via an API

call. In other cases, the attacker may leverage the speculative (mis-)execution of its own code to obtain sensitive information from the same process. For example, attack code, which is sandboxed by an interpreter, a just-in-time compiler, or a "safe" language, may wish to read memory it is not supposed to access. Although speculative execution can potentially expose sensitive data via a broad range of covert channels, the examples given cause speculative execution to first read a memory value at an attacker-chosen address and then perform a memory operation that modifies the cache state in a way that exposes the value.

For the final phase, the sensitive data is recovered. For Spectre attacks using Flush+Reload or Evict+Reload, the recovery process consists of timing the access to memory addresses in the cache lines being monitored.

Spectre attacks only assume that speculatively executed instructions can read from memory that the victim process could access normally, for example, without triggering a page fault or exception. Hence, Spectre is orthogonal to Meltdown,¹⁶ which exploits scenarios where some CPUs allow out-of-order execution of user instructions to read kernel memory. Consequently, even if a processor prevents speculative execution of instructions in user processes from accessing kernel memory, Spectre attacks still work.

4. VARIANT 1: EXPLOITING CONDITIONAL BRANCH MISPREDICTION

In this section, we demonstrate how conditional branch misprediction can be exploited by an attacker to read arbitrary memory from another context, for example, another process.

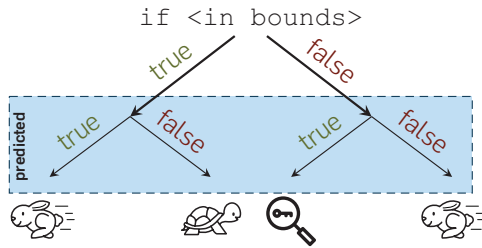
Consider the case where the code here is part of a function (e.g., a system call or a library) receiving an unsigned integer `x` from an untrusted source. The process running the code has access to an array of unsigned bytes: `array1` of size `array1_size`, and a second byte array `array2` of size 1 MB.

```
if (x < array1_size)
    y = array2[array1[x] * 4096];
```

The code fragment begins with a bounds check on `x`. This check is essential for security because it prevents the processor from reading sensitive memory outside of `array1`. Otherwise, an out-of-bounds input `x` could trigger an exception or could cause the processor to access sensitive memory by supplying `x = (address of a secret byte to read) - (base address of array1)`.

Figure 1 illustrates the four cases of the bounds check in combination with speculative execution. Before the result of the bounds check is known, the CPU speculatively executes code following the condition by predicting the most likely outcome of the comparison. There are many reasons why the result of a bounds check may not be immediately known, for example, a cache miss preceding or during the bounds check, congestion of a required execution unit, complex arithmetic dependencies, or nested speculative execution.

Figure 1. Before the correct outcome of the bounds check is known, the branch predictor continues with the most likely branch target, leading to an overall execution speed-up if the outcome was correctly predicted. However, if the bounds check is incorrectly predicted as true, an attacker can leak secret information in certain scenarios.



However, as illustrated, a correct prediction of the condition in these cases leads to faster overall execution.

Unfortunately, during speculative execution, the conditional branch for the bounds check can follow the incorrect path. In this example, suppose an adversary causes the code to run such that:

- the value of x is maliciously chosen (out-of-bounds), such that $\text{array1}[x]$ resolves to a secret byte k somewhere in the victim's memory;
- array1_size and array2 are uncached, but k is cached; and
- previous operations received values of x that were valid, leading the branch predictor to assume the `if` will likely be true.

This cache configuration can occur naturally or can be created by an adversary, for example, by causing eviction of array1_size and array2 and then having the kernel use the secret key in a legitimate operation.

When the compiled code above runs, the processor begins by comparing the malicious value of x against array1_size . Reading array1_size results in a cache miss, and the processor faces a substantial delay until its value is available from DRAM. In the meantime, the branch predictor assumes the `if` will be true, then speculative execution adds x to the base address of array1 and requests the data at the resulting address from the memory subsystem. This read is a cache hit, and quickly returns the value of the secret byte k . Speculative execution continues, using k to compute the address of $\text{array2}[k*4096]$, and sending a request to read this address from memory (resulting in a cache miss). At some point after the read from array2 is initiated, the processor realizes that its speculative execution was erroneous and rewinds its register state. However, the speculative read from array2 affects the cache state in an address-specific manner, where the address depends on k .

To complete the attack, the adversary measures which location in array2 was brought into the cache, for example, via Flush+Reload or Prime+Probe. This reveals the value of k , because the victim's speculative execution cached $\text{array2}[k*4096]$, causing $\text{array2}[i*4096]$ to read quickly for

$i = k$, but slowly for all other $k \in [0, 255]$. Alternatively, by using Evict+Time, the adversary can immediately call the target function again with an in-bounds value x' and measure how long this second call takes. If $\text{array1}[x']$ equals k , then the location accessed in array2 is in the cache, and the operation will tend to be faster. (The multiplication by 4096 simplifies the attack by ensuring that each potential value of k maps to a different memory page, avoiding effects due to intra-page prefetching.)

Many different scenarios can lead to exploitable leaks using this variant. For example, instead of performing a bounds check, the mispredicted conditional branch(es) could be checking a previously computed safety result or an object type. Similarly, the code that is speculatively executed can take other forms, such as leaking a comparison result into a fixed memory location or may be spread over a much larger number of instructions. The cache status described above is also more restrictive than may be required. For example, in some scenarios, the attack works even if array1_size is cached, for example, if branch prediction results are applied during speculative execution even if the values involved in the comparison are known. As a result, mitigation efforts are likely to be ineffective if targeted narrowly to a specific code pattern or scenario (see Sections 6 and 7).

4.1. Experimental results

We performed experiments on multiple Intel x86 processor architectures (Ivy Bridge, Haswell, Broadwell, Skylake, and Kaby Lake) and AMD Ryzen. The Spectre vulnerability was observed on all these CPUs, and we observed that speculative execution can run hundreds of instructions ahead. Similar results were observed on both 32- and 64-bit modes, and under both Linux and Windows. Some processors based on the ARM architecture also support speculative execution, and our initial testing confirmed that ARM Cortex-A57 and Cortex-A53 and Qualcomm Kyro 280 CPUs.

4.2. Example implementation in C

Proof-of-concept code in C for x86 processors is found in the full paper or is available from <https://gist.github.com/anonymous/99a72c9c1003f8ae0707b4927ec1bd8a>. This unoptimized implementation can read around 10KB/s on an i7-4650U with a low (<0.01%) error rate.

4.3. Example implementation in JavaScript

We developed a proof-of-concept in JavaScript and tested it in Google Chrome version 62.0.3202, which allows a Website to read private memory from the process in which it runs. The code is illustrated in Listing 1.

On branch-predictor mistraining passes, `index` is set (via bit operations) to an in-range value. On the final iteration, `index` is set to an out-of-bounds address into `simpleByteArray`. We used a variable `localJunk` to ensure that operations are not optimized out. The `| 0` operation converts the value to a 32-bit integer, acting as an optimization hint to the JavaScript interpreter. Like other optimized JavaScript engines, V8 performs just-in-time compilation to convert JavaScript into machine language. Dummy operations were placed in the code surrounding Listing 1 to make

```

1 if (index < simpleByteArray.length) {
2   index = simpleByteArray[index | 0];
3   index = (((index * 4096) | 0) & (32*1024*1024-1)) | 0;
4   localJunk ^= probeTable[index|0] | 0;
5 }

```

Listing 1. Exploiting speculative execution via JavaScript.

```

1 cmpl r15, [rbp-0xe0]           ; Compare index (r15) against simpleByteArray.length
2 jnc 0x24dd099bb870              ; If index >= length, branch to instruction after movq below
3 REX.W leaq rsi, [r12+rdx*1]  ; Set rsi = r12 + rdx = addr of first byte in simpleByteArray
4 movzqbl rsi, [rsi+r15*1]     ; Read byte from address rsi+r15 (= base address + index)
5 shll rsi, 12                   ; Multiply rsi by 4096 by shifting left 12 bits
6 andl rsi, 0x1fffffff            ; AND reassures JIT that next operation is in-bounds
7 movzqbl rsi, [rsi+r8*1]     ; Read from probeTable
8 xorl rsi, rdi                 ; XOR the read result onto localJunk
9 REX.W movq rdi, rsi          ; Copy localJunk into rdi

```

Listing 2. Disassembly of JavaScript example from Listing 1.

`simpleByteArray.length` be stored in local memory so that it can be removed from the cache during the attack. See Listing 2 for the resulting disassembly output from D8.

As the `clflush` instruction is not accessible from JavaScript, we use cache eviction instead,¹⁹ that is, we access other memory locations in a way such that the target memory locations are evicted afterward. The leaked results are conveyed via the cache status of `probeTable[n*4096]` for $n \in [0, 255]$, so the attacker has to evict these 256 cache lines. The length parameter (`simpleByteArray.length` in the JavaScript code and `[ebp-0xe0]` in the disassembly) needs to be evicted as well.

JavaScript does not provide access to the `rdtscp` instruction, and Chrome intentionally degrades the accuracy of its high-resolution timer to dissuade timing attacks using `performance.now()`. However, the Web Workers feature of HTML5 makes it simple to create a separate thread that repeatedly decrements a value in a shared memory location.²² This approach yields a high-resolution timer that provides sufficient resolution.

4.4. Example implementation exploiting eBPF

As a third example of exploiting conditional branches, we developed a reliable proof-of-concept which leaks kernel memory from an unmodified Linux kernel without patches against Spectre by abusing the extended BPF (eBPF) interface. eBPF is a Linux kernel interface based on the Berkeley Packet Filter (BPF)¹⁸ that can be used for a variety of purposes, such as filtering packets based on their contents. eBPF permits unprivileged users to trigger the interpretation or JIT compilation and subsequent execution of user-supplied, kernel-verified eBPF bytecode in the context of the kernel. The basic concept of the attack is similar to the concept of the attack against JavaScript.

In this attack, we use the eBPF code only for the speculatively executed code. We use native code in user space to acquire the covert channel information. This is a difference to the JavaScript example above, where both functions are implemented in the scripted language. To speculatively access secret-dependent locations in user-space memory, we perform speculative out-of-bounds memory accesses to

an array in kernel memory, with an index large enough that the user-space memory is accessed instead.

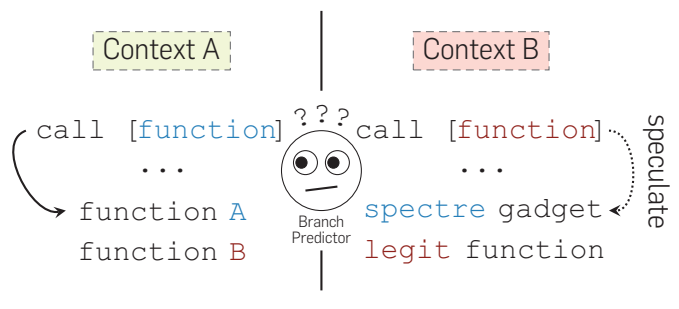
See the full paper for additional details.

5. VARIANT 2: POISONING INDIRECT BRANCHES

In this section, we demonstrate how indirect branches can be poisoned by an attacker and the resulting misprediction of indirect branches can be exploited. If the determination of the destination address of an indirect branch is delayed, due to a cache miss, speculative execution will often continue at a location predicted from previous code execution.

In Spectre variant 2, the adversary mistrains the branch predictor with malicious destinations, such that speculative execution continues at a location chosen by the adversary. This is illustrated in Figure 2, where the branch predictor is (mis-)trained in one context and applies the prediction in a different context. More specifically, the adversary can misdirect speculative execution to locations that would never occur during a legitimate program execution. This is an extremely powerful means for attackers, for example, enabling exposure of victim's memory even in the absence of an exploitable conditional branch misprediction leveraged in Section 4.

Figure 2. The branch predictor is (mis-)trained in the attacker-controlled context A. In context B, the branch predictor makes its prediction on the basis of training data from context A, leading to speculative execution at an attacker-chosen address which corresponds to the location of the Spectre gadget in the victim's address space.



For a simple example attack, we consider an attacker seeking to read a victim's memory, who has control over two registers when an indirect branch occurs. This commonly occurs in real-world binaries because functions manipulating externally received data routinely make function calls although registers contain values that an attacker controls. Often these values are ignored by the called function and instead they are simply pushed onto the stack in the function prologue and restored in the function epilogue.

The attacker also needs to locate a "Spectre gadget," that is, a code fragment whose speculative execution will transfer the victim's sensitive information into a covert channel. For this example, a simple and effective gadget would be formed by two instructions (which do not necessarily need to be adjacent) where the first adds (or XORs, subtracts, etc.) the memory location addressed by an attacker-controlled register R1 onto an attacker-controlled register R2, followed by any instruction that accesses memory at the address in R2. In this case, the gadget provides the attacker control (via R1) over which address to leak and control (via R2) over how the leaked memory maps to an address which is read by the second instruction. On the CPUs we tested, the gadget must reside in memory executable by the victim for the CPU to perform speculative execution. However, with several megabytes of shared libraries mapped into most processes,⁵ an attacker has ample space to search for gadgets without even having to search in the victim's own code.

The choice of gadget depends on what state is known or controlled by the adversary, where the information sought by the adversary resides (e.g., registers, stack, memory, etc.), the adversary's ability to control speculative execution, what instruction sequences are available to form gadgets, and what channels can leak information from speculative operations. For example, a cryptographic function that returns a secret value in a register may become exploitable if the attacker can simply induce speculative execution at an instruction that brings memory from the address specified in the register into the cache. Likewise, although the example above assumes that the attacker controls two registers, the attacker's control over a single register, value on the stack, or memory value is sufficient for some gadgets.

In many ways, exploitation is similar to return-oriented programming (ROP),²³ except that the correctly written software is vulnerable, gadgets are limited in their duration but need not terminate cleanly (because the CPU will eventually recognize the speculative error), and gadgets must exfiltrate data via side channels rather than explicitly. Still, speculative execution can perform complex sequences of instructions, such as reading from the stack, performing arithmetic, branching (including multiple times), and reading memory.

The full paper includes details about branch predictor behavior and mistraining techniques for a range of processors, as well as attack implementations targeting a Microsoft Windows application and the KVM hypervisor.

6. VARIATIONS

So far, we have demonstrated attacks that leverage changes in the state of the cache that occur during speculative

execution. Future processors (or existing processors with different microcode) may behave differently, for example, if measures are taken to prevent speculatively executed code from modifying the cache state. In this section, we examine potential variants and conclude that virtually any observable effect of speculatively executed code can potentially lead to leaks of sensitive information. Although the following techniques are not needed for the processors we tested, it is essential to understand potential variations when designing or evaluating mitigations.

Spectre variant 4. Spectre variant 4 uses speculation in the store-to-load forwarding logic.⁷ The processor speculates that a load does not depend on the previous store. The exploitation mechanics are similar to variant 1 and 2 that we discussed in detail in this paper.

Evict+Time. The Evict+Time attack²⁰ works by measuring the timing of operations that depend on the state of the cache. This technique can be adapted to use Spectre as follows. Consider the code:

```
if (false but mispredicts as true)
  read array1[R1]
  read [R2]
```

Suppose register R1 contains a secret value. If the speculatively executed memory read of `array1[R1]` is a cache hit, then nothing will go on the memory bus, and the read from `[R2]` will initiate quickly. If the read of `array1[R1]` is a cache miss, then the second read may take longer, resulting in different timing for the victim thread. In addition, other components in the system that can access memory (such as other processors) may be able to sense the presence of activity on the memory bus or other effects of the memory read. We note that this attack can work even if speculative execution does not modify the contents of the cache. All that is required is that the state of the cache affects the timing of speculatively executed code or some other property that ultimately becomes visible to the attacker.

Instruction timing. Spectre vulnerabilities do not necessarily need to involve caches. Instructions whose timing depends on the values of the operands may leak information on the operands. In the following example, the multiplier is occupied by the speculative execution of `multiply R1, R2`. The timing of when the multiplier becomes available for `multiply R3, R4` (either for out-of-order execution or after the misprediction is recognized) could be affected by the timing of the first multiplication, revealing information about R1 and R2.

```
if (false but mispredicts as true)
  multiply R1, R2
  multiply R3, R4
```

Contention on the register file. Suppose the CPU has a register file with a finite number of registers available for storing checkpoints for speculative execution. In the following example, if `condition` on R1 in the second "if" is true, then an extra speculative execution checkpoint will be created

than if `condition` on `R1` is false. If an adversary can detect this checkpoint, if speculative execution of code in hyper-threads is reduced due to a shortage of storage, this reveals information about `R1`.

```
if (false but mispredicts as true)
  if (condition on R1)
    if (condition)
```

Variations on speculative execution. Even code that contains no conditional branches can potentially be at risk. For example, consider the case where an attacker wishes to determine whether `R1` contains an attacker-chosen value `X` or some other value. The ability to make such determinations is sufficient to break some cryptographic implementations. The attacker mistrains the branch predictor such that after an interrupt occurs, the interrupt return mispredicts to an instruction that reads memory [`R1`]. The attacker then chooses `X` to correspond to a memory address suitable for Flush+Reload, revealing whether `R1 = X`. Although the `iret` instruction is serializing on Intel CPUs, other processors may apply branch predictions.

Leveraging arbitrary observable effects. Virtually any observable effect of speculatively executed code can be leveraged to create the covert channel that leaks sensitive information. For example, consider a processor that has been designed so that speculative reads cannot modify the cache. When the code here runs, the speculative lookup in `array2` still occurs, and its timing will be affected by the cache state entering speculative execution. This timing in turn can affect the depth and timing of subsequent speculative operations. Thus, by manipulating the state of the cache prior to speculative execution, an adversary can potentially leverage virtually any observable effect from speculative execution.

```
if (x < array1_size){
  y = array2[array1[x] * 4096];
  // do something detectable when
  // speculatively executed
}
```

The final observable operation could involve virtually any side channel or covert channel, such as contention for resources (buses, arithmetic units, etc.) and conventional side-channel emanations (such as electromagnetic radiation or power consumption).

A more general form of this would be:

```
if (x < array1_size) {
  y = array1[x];
  // something using y that is observable
  // when speculatively executed
}
```

7. MITIGATION OPTIONS

Several countermeasures for Spectre attacks have been proposed. Each addresses one or more of the features that the attack relies upon. We now discuss these countermeasures and their applicability, effectiveness, and cost.

7.1. Preventing speculative execution

Speculative execution is required for Spectre attacks. Ensuring that instructions are executed only when the control flow leading to them is ascertained would prevent speculative execution and, with it, Spectre attacks. Although effective as a countermeasure, this would cause a significant degradation in the performance of the processor.

Although current processors do not appear to have methods that allow software to disable speculative execution, such modes could be added in future processors, or potentially be introduced via microcode changes. Still, this solution is unlikely to provide an immediate fix to the problem.

Alternatively, the software could be modified to use *serializing* or *speculation blocking* instructions that ensure that instructions following them are not executed speculatively. For x86, CPU vendors recommend the use of the `lfence` instruction.⁹ The safest approach to protect conditional branches would be to add such an instruction on the two outcomes of every conditional branch, but this amounts to disabling branch prediction and would dramatically reduce performance. An improved approach is to use static analysis⁹ to reduce the number of speculation blocking instructions required, as many code paths do not have the potential to read and leak out-of-bounds memory. In contrast, Microsoft's C compiler MSVC takes an approach of defaulting to unprotected code unless the static analyzer detects a known bad code pattern but, as a result, misses many vulnerable code patterns.¹²

The approach requires that all potentially vulnerable software is instrumented. Hence, for protection, updated software binaries and libraries are required. This could be an issue for legacy software. In addition, this approach is primarily focused on variant 1, and does not address all variants.

7.2. Preventing access to secret data

Other countermeasures can prevent speculatively executed code from accessing secret data. One such measure, used by the Google Chrome Web browser, is to execute each Website in a separate process.²⁶ Because Spectre attacks only leverage the victim's permissions, an attack such as the one we performed using JavaScript (cf. Section IV-C) would not be able to access data from the processes assigned to other Websites.

WebKit employs two strategies for limiting access to secret data by speculatively executed code.²¹ The first strategy replaces array bounds checking with index masking. Instead of checking that an array index is within the bounds of the array, WebKit applies a bit mask to the index, ensuring that it is not much bigger than the array size. Although masking may result in access outside the bounds of the array, this limits the distance of the bounds violation, preventing the attacker from accessing arbitrary memory. The second strategy protects access to pointers by xoring them with a pseudo-random *poison* value. An adversary who does not know the poison value cannot use a poisoned pointer (although various cache attacks could leak the poison value), and the poison value ensures that mispredictions on the branch instructions used for type checks will result in pointers associated with the type being used for another type.

7.3. Preventing data from entering covert channels

Future processors could potentially track whether data was fetched as the result of a speculative operation and, if so, prevent that data from being used in subsequent operations that might leak it. However, current processors do not generally have this capability.

7.4. Limiting data extraction from covert channels

To exfiltrate information from transient instructions, Spectre attacks use a covert communication channel. Multiple approaches have been suggested for mitigating such channels (cf. Ge et al.⁴). A common approach is to degrade timers, which may decrease attack performance, but does not guarantee that attacks are not possible.

7.5. Preventing branch poisoning

To prevent indirect branch poisoning, Intel and AMD extended the ISA with mechanisms for limiting adversaries' ability to influence indirect branch speculation.^{2,8} The performance impact varies from a few percent to a factor of 4 or more, depending on which countermeasures are employed, how comprehensively they are applied (e.g., limited use in the kernel vs. full protection for all processes), and the efficiency of the hardware and microcode implementations.

Google suggests an alternative mechanism for preventing indirect branch poisoning called *retpolines*.²⁸ A retpoline is a code sequence that replaces indirect branches with return instructions. The construct further contains code that makes sure that the return instruction is predicted to a benign endless loop through the return stack buffer, although the actual target destination is reached by pushing it on the stack and returning to it, that is, using the *ret* instruction. When return instructions can be predicted by other means, the method may be impractical. Intel issued microcode updates for some processors, which fall back to the BTB for the prediction, to disable this fallback mechanism.⁹

8. CONCLUSION

A fundamental assumption underpinning software security techniques is that the processor will faithfully execute program instructions, such as its safety checks. This paper presents Spectre attacks, which leverage the fact that speculative execution violates this assumption. The techniques we demonstrate are practical, do not require any software vulnerabilities, and allow adversaries to read private memory and register contents from other processes and security contexts.

Software security fundamentally depends on having a clear common understanding between hardware and software developers as to what information CPU implementations are (and are not) permitted to expose from computations. As a result, although the countermeasures described in the previous section may help limit practical exploits in the short term, they are only stop-gap measures as there is typically formal architectural assurance as to whether any specific code construction is safe across today's processors—much less future


designs. As a result, we believe that long-term solutions will require fundamentally changing instruction set architectures.

More broadly, there are trade-offs between security and performance. The vulnerabilities in this paper, as well as many others, arise from a long-standing focus in the technology industry on maximizing performance. As a result, processors, compilers, device drivers, operating systems, and numerous other critical components have evolved compounding layers of complex optimizations that introduce security risks. As the costs of insecurity rise, these design choices need to be revisited. In many cases, alternative implementations optimized for security will be required.

Acknowledgments

Several authors of this paper found Spectre independently, ultimately leading to this collaboration. We thank Mark Brand from Google Project Zero for contributing ideas. We thank Intel for their professional handling of this issue through communicating a clear timeline and connecting all involved researchers. We thank ARM for technical discussions on aspects of this issue. We thank Qualcomm and other vendors for their fast response upon disclosing the issue. Finally, we want to thank our reviewers for their valuable comments.

Daniel Gruss, Moritz Lipp, Stefan Mangard, and Michael Schwarz were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681402).

Daniel Genkin was supported by NSF awards #1514261 and #1652259, financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology, the 2017–2018 Rothschild Postdoctoral Fellowship, and the Defense Advanced Research Project Agency (DARPA) under Contract #FA8650-16-C-7622. 

References

1. Aciğmez, O., Koç, Ç.K., Seifert, J.-P. Predicting Secret Keys Via Branch Prediction. In: *CT-RSA*, 2007.
2. Advanced Micro Devices, Inc. Software Techniques for Managing Speculation on AMD Processors, 2018. [Online]. <http://developer.amd.com/wordpress/media/2013/12/Managing-Speculation-on-AMD-Processors.pdf>
3. Bernstein, D.J. Cache-Timing Attacks on AES. 2005. [Online]. <http://cr.ypt.org/antiforgery/cachetiming-20050414.pdf>
4. Ge, Q., Yarom, Y., Cock, D., Heiser, G. A survey of microarchitectural timing attacks and countermeasures on contemporary hardware. *J. Cryptogr. Eng.*, 1, 8 (2018), 1–27.
5. Gruss, D., Spreitzer, R., Mangard, S. Cache template attacks: Automating attacks on inclusive last-level caches. In *USENIX Security Symposium*, 2015.
6. Gullasch, D., Bangert, E., Krenn, S. Cache games—Bringing access-based cache attacks on AES to practice. In *S&P*, 2011.
7. Horn, J. Speculative execution, variant 4: Speculative store bypass, 2018. [Online]. <https://bugs.chromium.org/p/project-zero/issues/detail?id=1528>
8. Intel Corp. Speculative Execution Side Channel Mitigations, Jan. 2018. [Online]. <https://software.intel.com/sites/default/files/managed/c5/63/336996-Speculative-Execution-Side-Channel-Mitigations.pdf>
9. Intel Corp. Intel Analysis of Speculative Execution Side Channels, Jan. 2018. [Online]. <https://newsroom.intel.com/wpcontent/uploads/sites/11/2018/01/Intel-Analysis-of-Speculative-Execution-Side-Channels.pdf>
10. Irazoqui Apecechea, G., Eisenbarth, T., Sunar, B. SSA: A shared cache attack that works across cores and defies VM sandboxing—and its application to AES. In *S&P*, 2015.

11. Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J.H., Lee, D., Wilkerson, C., Lai, K., Mutlu, O. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In *ISCA*, 2014.
12. Kocher, P. Spectre mitigations in Microsoft's C/C++ compiler, 2018. [Online]. <https://www.paulkocher.com/doc/MicrosoftCompilerSpectreMitigation.html>
13. Kocher, P., Jaffe, J., Jun, B. Differential power analysis. In *CRYPTO*, 1999.
14. Kocher, P.C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO*, 1996.
15. Lipp, M., Gruss, D., Spreitzer, R., Maurice, C., Mangard, S. ARMageddon: Cache attacks on mobile devices. In *USENIX Security Symposium*, 2016.
16. Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Fogh, A., Horn, J., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., Hamburg, M. Meltdown: Reading kernel memory from user space. In *USENIX Security Symposium (to appear)*, 2018.
17. Liu, F., Yarom, Y., Ge, Q., Heiser, G., Lee, R.B. Last-level cache side-channel attacks are practical. In *S&P*, 2015.
18. McCanne, S., Jacobson, V. The BSD packet filter: A new architecture for user-level packet capture. In *USENIX Winter*, 1993.
19. Oren, Y., Kemerlis, V.P., Sethumadhavan, S., Keromytis, A.D. The spy in the sandbox: Practical cache attacks in JavaScript and their implications. In *CCS*, 2015.
20. Osvik, D.A., Shamir, A., Tromer, E. Cache attacks and countermeasures: The case of AES. In *CT-RSA*, 2006.
21. Pizlo, F. What spectre and meltdown mean for WebKit, Jan. 2018. [Online]. <https://webkit.org/blog/8048/what-spectreand-meltdown-mean-for-webkit/>
22. Schwarz, M., Maurice, C., Gruss, D., Mangard, S. Fantastic timers and where to find them: High-resolution microarchitectural attacks in JavaScript. In *Financial Cryptography*, 2017.
23. Shacham, H. The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86). In *CCS*, 2007.
24. Sibert, O., Porras, P.A., Lindell, R. The Intel 80x86 processor architecture: Pitfalls for secure systems. In *S&P*, 1995.
25. Tang, A., Sethumadhavan, S., Stolfo, S. CLKSCREW: Exposing the perils of security-oblivious energy management. In *USENIX Security Symposium*, 2017.
26. The Chromium Projects. Site Isolation. [Online]. <http://www.chromium.org/Home/chromiumsecurity/site-isolation>
27. Tsunoo, Y., Saito, T., Suzaki, T., Shigeri, M., Miyauchi, H. Cryptanalysis of DES implemented on computers with cache. In *CHES*, 2003.
28. Turner, P. Retpoline: A software construct for preventing branch-target-injection. [Online]. <https://support.google.com/faqs/answer/7625886>
29. Yarom, Y., Falkner, K. Flush + reload: A high resolution, low noise, L3 cache side-channel attack. In *USENIX Security Symposium*, 2014.

Paul Kocher (<https://www.paulkocher.com>) (paul@paulkocher.com), Independent.

Jann Horn (jannah@google.com), Google Project Zero.

Anders Fogh (Anders_fogh@hotmail.com), G DATA Advanced Analytics.

Daniel Genkin (genkin@umich.edu), University of Michigan.

Daniel Gruss, Moritz Lipp, and Michael Schwarz (daniel.gruss@iaik.tugraz.at), Graz University of Technology.

Werner Haas and Thomas Prescher (werner.haas,thomas.prescher@cyberus-technology.de), Cyberus Technology.

Mike Hamburg (mhamburg@rambus.com), Rambus, Cryptography Research Division.

Yuval Yarom (yval@cs.adelaide.edu.au), University of Adelaide and Data61.

© 2020 ACM 0001-0782/20/7 \$15.00

Computing and the National Science Foundation, 1950-2016

Building a Foundation for Modern Computing

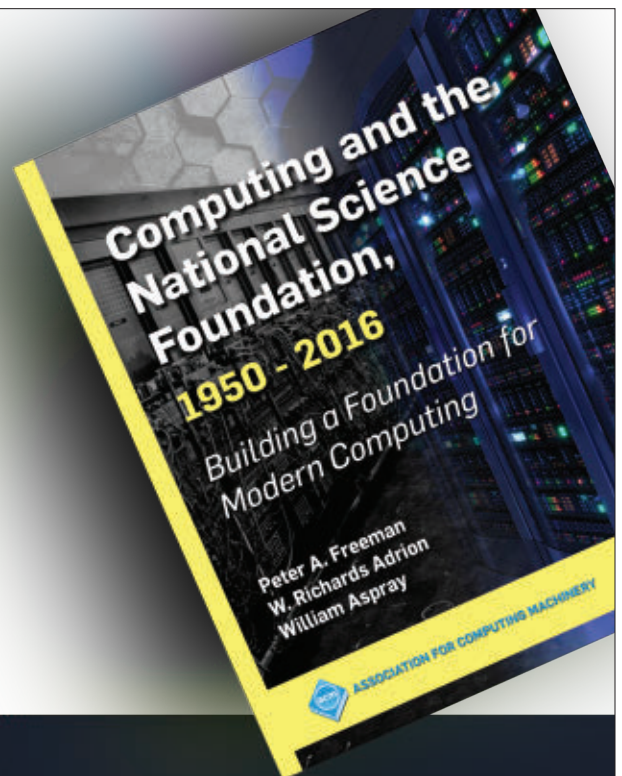
Peter A. Freeman
W. Richards Adrion
William Aspray

ISBN: 978-1-4503-7271-8

DOI: 10.1145/3335772

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



ACM BOOKS
 Collection II

Technical Perspective

ASIC Clouds: Specializing the Datacenter

By Parthasarathy Ranganathan

THE COMPUTER ARCHITECTURE community is at an interesting crossroads. Moore's Law is slowing down, stressing traditional assumptions around computing getting cheaper and faster over time—assumptions that underpin a significant fraction of the economic growth over the past few decades. But at the same time, our demand continues to grow at phenomenal rates, with deeper analysis over growing volumes of data, new diverse workloads in the cloud, smarter edge devices, and new security constraints. Is the situation dire, or is this the beginning of a new phase in the evolution of system architecture?

Two recent trends provide hope that it is the latter! The first trend, at a microarchitecture level, is around *specialization* or domain-specific hardware/software codesign. Compared to a general-purpose processor, a specialized architecture such as an ASIC (application-specific integrated circuit) customizes the design for a specific application or workload class. A good example is Google's TPU series of ASICs. Such specialization leads to significant area and power efficiencies. The trade-off, of course, is we now do not have the volume advantages of a general-purpose system, whether it is around software ecosystem support (and ease of development) or around amortization of costs associated with building a custom chip (notably, the non-recurring expenses or NRE). The second trend, at a system level, is around warehouse-scale computing, or more broadly *cloud* computing, a computing model that treats the entire "datacenter as a computer." This model helps amortize costs across larger ensembles, but also provides additional benefits around ubiquitous access, simpler system management, and better encapsulation of hardware under higher-level software interfaces and abstractions. Initially popularized by large Internet services such as search, email, and social networks, cloud computing is now increasingly being adopted by traditional enterprises as well.

What happens when we combine these two trends? Can we build pur-


pose-built, warehouse-scale datacenters *customized* for (not just comprised of) large-scale arrays of ASIC accelerators or, to use a term coined in the following paper, *ASIC clouds*?

Interestingly, a proof point already exists, and from a very surprising source—bitcoin mining. Consider recent designs from companies such as Bitmain^a or Bitfury^b that use ASICs with tens to hundreds of cores custom-designed to run Bitcoin's hashing algorithm. Hundreds of these chips are assembled into custom boards, and hundreds of these boards are assembled into custom racks or containers in very specialized datacenters. Bitfury even goes one step further, using specialized immersion cooling to submerge its servers. While such bitcoin-mining ASIC clouds have demonstrably provided massive scale out, are they likely to gain more broader mainstream acceptance? How effective are these designs compared to traditional CPUs and GPUs and traditional datacenter designs? How do we reason about the architectural trade-offs or pervasive specialization from the ASIC to the server to the datacenter?

The following paper addresses these issues and more. The authors distill the lessons from bitcoin mining systems to develop a broader architectural framework for ASIC clouds. Specifically, they propose a hierarchical design, starting with core specialized functions that are replicated across ASICs and connected with a custom on-chip network; ASIC voltages are customizable allowing trade-offs for energy efficiency and total costs of ownership. Multiple ASICs are assembled together in a specialized server with custom cooling and power delivery systems and workload-tailored DRAM and I/O subsystems. Multiple servers are further assembled into racks and datacenters, again with computation-specific customization of thermals and power delivery. Using this architecture, the study examines ASIC clouds

for four applications that span a diverse range of properties: different flavors of bitcoin mining, but also deep learning and video transcoding. Their results show the promise of ASIC clouds—two to three orders of magnitude improved efficiency advantages compared to traditional CPU- or GPU-based approaches.

Perhaps an even more exciting contribution of the paper is a methodology that federates different modeling approaches to derive pareto-optimal ASIC cloud configurations. Starting with data extracted from "place-and-route" circuit optimizations at the circuits level and computational fluid dynamics models at the systems level, this approach performs an exhaustive search to find the best design optimized across a number of parameters: the area per ASIC and the number of ASICs and their operating voltage, the number of DRAM chips associated per ASIC, and choices around the case design and the power delivery and cooling subsystems. A notable contribution is a refinement of a large amount of data into a "two-for-two" rule on when ASIC clouds are appropriate.

This is but the start of an interesting direction of exploration for the broader community. Given the nascent and fast-evolving nature of current ASIC solutions, how do we enable ASIC clouds to adapt rapidly to changing accelerator designs, to diversity across different classes of accelerators? Can the holistic design of ASIC clouds enable additional optimizations, for example, around addressing the speed and NRE of future specialized designs? These and other open questions highlight how we are entering an era of significant change, one where it is not "business as usual." The architecture and methodology in this paper provide a foundation, and a baseline, to explore more interesting ideas at the confluence of two of the most exciting ideas the community is rallying around. 

Parthasarathy Ranganathan is a Distinguished Engineer and area technical lead for hardware and datacenters at Google, San Francisco, CA, USA.

Copyright held by author.

a www.bitmain.com

b www.bitfury.com

ASIC Clouds: Specializing the Datacenter for Planet-Scale Applications

By Michael Bedford Taylor, Luis Vega, Moein Khazraee, Ikuo Magaki, Scott Davidson, and Dustin Richmond

Abstract

Planet-scale applications are driving the exponential growth of the Cloud, and datacenter specialization is the key enabler of this trend. GPU- and FPGA-based clouds have already been deployed to accelerate compute-intensive workloads. ASIC-based clouds are a natural evolution as cloud services expand across the planet. ASIC Clouds are purpose-built datacenters comprised of large arrays of ASIC accelerators that optimize the total cost of ownership (TCO) of large, high-volume scale-out computations. On the surface, ASIC Clouds may seem improbable due to high NREs and ASIC inflexibility, but large-scale ASIC Clouds have already been deployed for the Bitcoin cryptocurrency system. This paper distills lessons from these Bitcoin ASIC Clouds and applies them to other large scale workloads such as YouTube-style video-transcoding and Deep Learning, showing superior TCO versus CPU and GPU. It derives Pareto-optimal ASIC Cloud servers based on accelerator properties, by jointly optimizing ASIC architecture, DRAM, motherboard, power delivery, cooling, and operating voltage. Finally, the authors examine the impact of ASIC NRE and when it makes sense to build an ASIC Cloud.

1. INTRODUCTION

In the last decade, two parallel trends in the computational landscape have emerged. The first is the bifurcation of computation into two sectors: cloud and mobile. The second is the rise of dark silicon^{15, 3, 4, 2} and dark silicon aware design techniques^{13, 14, 10, 16, 11} such as specialization and near-threshold computation. Specialized hardware has existed in mobile computing for a while due to extreme power constraints; however, recently there has been an increase in the amount of specialized hardware showing up in cloud datacenters. Examples include Baidu's GPU-based cloud for distributed neural network acceleration, Microsoft's FPGA-based cloud for Bing Search,⁹ and by JP Morgan Chase for hedgefund portfolio evaluation.¹²

At the level of a single node, we know that ASICs can offer order-of-magnitude improvements in energy-efficiency and cost-performance over CPU, GPU, and FPGA.

Our recent papers^{8, 6, 7, 17} explore the concept of *ASIC Clouds* which are purpose-built datacenters comprised of large arrays of ASIC accelerators. ASIC Clouds are not ASIC supercomputers that scale up problem sizes for a single tightly coupled computation; rather, ASIC Clouds target scale-out workloads consisting of many independent but similar jobs, often on behalf of millions or billions of end-users.

As more and more services are built around the Cloud model, we see the emergence of planet-scale workloads (think Facebook's face recognition of uploaded pictures, or Apple's Siri voice recognition, or the IRS performing tax audits with neural nets) where datacenters are performing the same computation across many users. These scale-out workloads can easily leverage racks of ASIC servers containing arrays of chips that in turn connect arrays of replicated compute accelerators (RCAs) on an on-chip network. The large scale of these workloads creates the economical justification to pay the nonrecurring engineering (NRE) costs of ASIC development and deployment. As a workload grows, the ASIC Cloud can be scaled in the datacenter by adding more ASIC servers, unlike accelerators in say a mobile phone population,³ where the accelerator-to-processor ratio is fixed at tapeout.

Our research examined ASIC Clouds in the context of four key applications that show great potential for ASIC Clouds, such as YouTube-style video transcoding, Bitcoin and Litecoin mining, and Deep Learning. ASICs achieve large reductions in silicon area and energy consumption versus CPUs, GPUs, and FPGAs. We show how to specialize the ASIC server to maximize efficiency, employing optimized ASICs, a customized printed circuit board (PCB), custom-designed cooling systems and specialized power delivery systems, and tailored DRAM and I/O subsystems. ASIC voltages are customized in order to tweak energy efficiency and minimize total cost of ownership (TCO). The datacenter itself can also be specialized, optimizing rack-level and datacenter-level thermals and power delivery to exploit the knowledge of the computation. We developed tools that consider all aspects of ASIC Cloud design in a bottom-up way, and methodologies that reveal how the designers of these novel systems can optimize TCO in real-world ASIC Clouds. Finally, we proposed a new rule that explains when it makes sense to design and deploy an ASIC Cloud, considering the engineering expense (NRE) of designing the machines.

Notably, the original version of this paper^{1, 8} predicted Machine Learning ASIC Clouds, before Google announced the first Tensor Processing cloud in 2016.⁵ The same paper also predicted video transcoding clouds before Facebook's

The content of this paper draws from "ASIC Clouds: Specializing the Data Center," published in *Proceedings of the IEEE Int. Symp. Computer Architecture*, June 2016, and from "Specializing the Planet's Computation: ASIC Clouds" published in *IEEE Micro*, June 2017.

Mount Shasta video transcoding ASIC Cloud design was announced in March 2019.

2. ASIC CLOUD ARCHITECTURE

At the heart of any ASIC Cloud is an energy-efficient, high-performance, specialized replicated compute accelerator, or RCA, that is multiplied up by having multiple copies per ASICs, multiple ASICs per server, multiple servers per rack, and multiple racks per datacenter as shown Figure 1. Work requests from outside the datacenter will be distributed across these RCAs in a scale-out fashion. All system components can be customized for the application to minimize TCO.

Each ASIC interconnects its RCAs using a customized on-chip network. The ASIC's control plane unit also connects to this network and schedules incoming work from the ASIC's off-chip router onto the RCAs. Next, the packaged ASICs are arranged in lanes on a customized PCB, and connected to a controller which bridges to the off-PCB interface (1-100 GigE, RDMA, PCI-e, etc). In some cases, DRAMs may connect directly to the ASICs. The controller can be implemented by an FPGA, microcontroller, or a Xeon processor and schedules remote procedure calls (RPCs) that come from the off-PCB interface on to the ASICs. Depending on the application, it may implement the nonacceleratable part of the workload or perform UDP/TCP-IP offload.

Each lane is enclosed by a duct and has a dedicated fan blowing air through it across the ASIC heatsinks. Our simulations indicate that using ducts results in better cooling performance compared to conventional or staggered layout. The PCB, fans, and power supply are enclosed in a 1U server, which is then assembled into racks in a datacenter. Based on ASIC needs, the PSU and DC/DC converters are customized for each server.

3. DESIGNING AN ASIC CLOUD

Our ASIC Cloud Server configuration evaluator, as shown in Figure 2a, starts with a Verilog implementation of an accelerator, or a detailed evaluation of the accelerator's

properties from the research literature. In the design of an ASIC Server, we must decide how many chips should be placed on the PCB and how large, in mm² of silicon, each chip should be. The size of each chip determines how many RCAs will be on each chip. In each duct-enclosed lane of ASIC chips, each chip receives around the same amount of airflow from the intake fans, but the most downstream chip receives the hottest air, which includes the waste heat from the other chips. Therefore, the thermally bottlenecking ASIC is the one in the back, shown in our detailed Computational Fluid Dynamics (CFD) simulations as shown in Figure 2b. Our simulations show that breaking a fixed heat source into smaller ones with the same total heat output improves the mixing of warm and cold area, resulting in lower temperatures. Using thermal optimization techniques, we established fundamental connection between an RCA's properties, the number of RCAs placed in an ASIC, and how many ASICs go on a PCB in a server. Given these properties, our heat sink solver determines the optimal heat sink configuration. Results are validated with the CFD simulator. In the sidebar entitled "Design Space Evaluation," we show how we apply this evaluation flow across the design space in order to determine TCO and Pareto optimal points that trade off \$ per op/s (an accelerator's hardware cost efficiency) and W per op/s (an accelerator's energy efficiency).

4. APPLICATION CASE STUDIES

To explore ASIC Clouds across a range of accelerator properties, we examined four applications: Bitcoin mining, Litecoin mining, Video Transcoding, and Deep Learning that span a diverse range of properties, as shown in Figure 3.

Perhaps the most mature of these applications is Bitcoin mining. Our inspiration for ASIC Clouds came from our intensive study of Bitcoin mining clouds,⁴ which are one of the first known instances of a real life ASIC Cloud. Figure 4 shows the massive scale out of the Bitcoin mining workload, which in 2015 operated at the performance of 3.2 billion GPUs. Bitcoin

Figure 1. High-level abstract architecture of an ASIC Cloud. Specialized replicated compute accelerators (RCA) are multiplied up by having multiple copies per ASICs, multiple ASICs per server, multiple servers per rack, and multiple racks per datacenter. Server controller can be an FPGA, microcontroller, or a Xeon processor. Power delivery and cooling system are customized based on ASIC needs. If required, there would be DRAMs on the PCB as well.

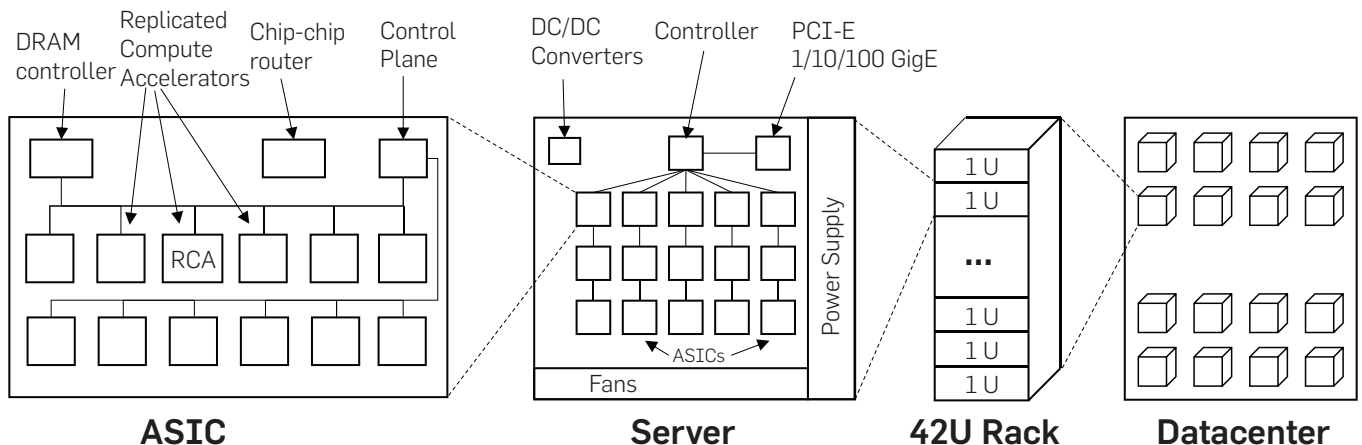


Figure 2. Evaluating an ASIC configuration. (a) The server cost, per server hash rate, and energy efficiency are evaluated using RCA properties and a flow that optimizes server heatsinks, die size, voltage, and power density. (b) Thermal verification of an ASIC Cloud server using CFD tools to validate the flow results. The farthest ASIC from the fan has the highest temperature and is the bottleneck for power per ASIC at a fixed voltage and energy efficiency.

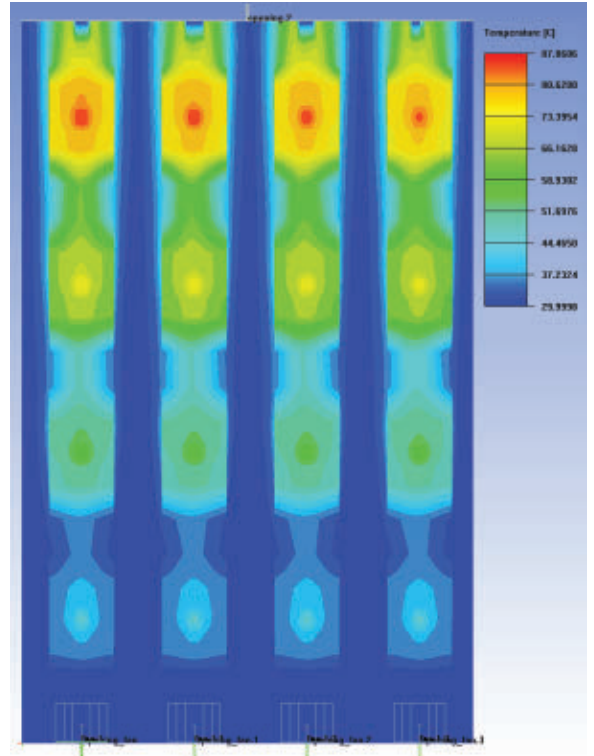
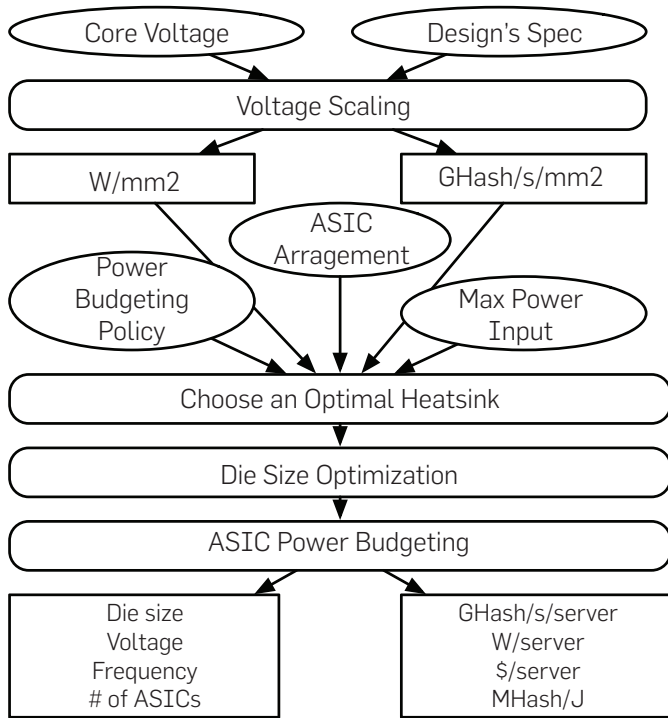
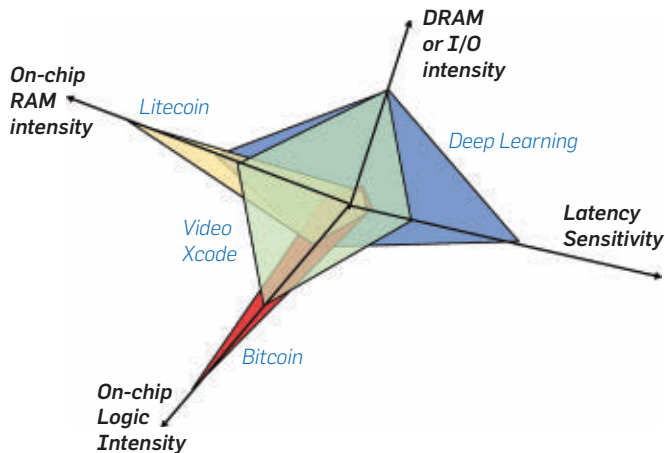


Figure 3. Accelerator properties. We explored applications with diverse requirements.



clouds have undergone a rapid ramp from CPU to GPU to FPGA to the most advanced ASIC technology available today. Bitcoin is a very logic intensive design which has high power density and no need for SRAM or external DRAM.

Litecoin is another popular cryptocurrency mining system that has been deployed into clouds. Unlike Bitcoin, it is an SRAM-intensive application which has low power density.

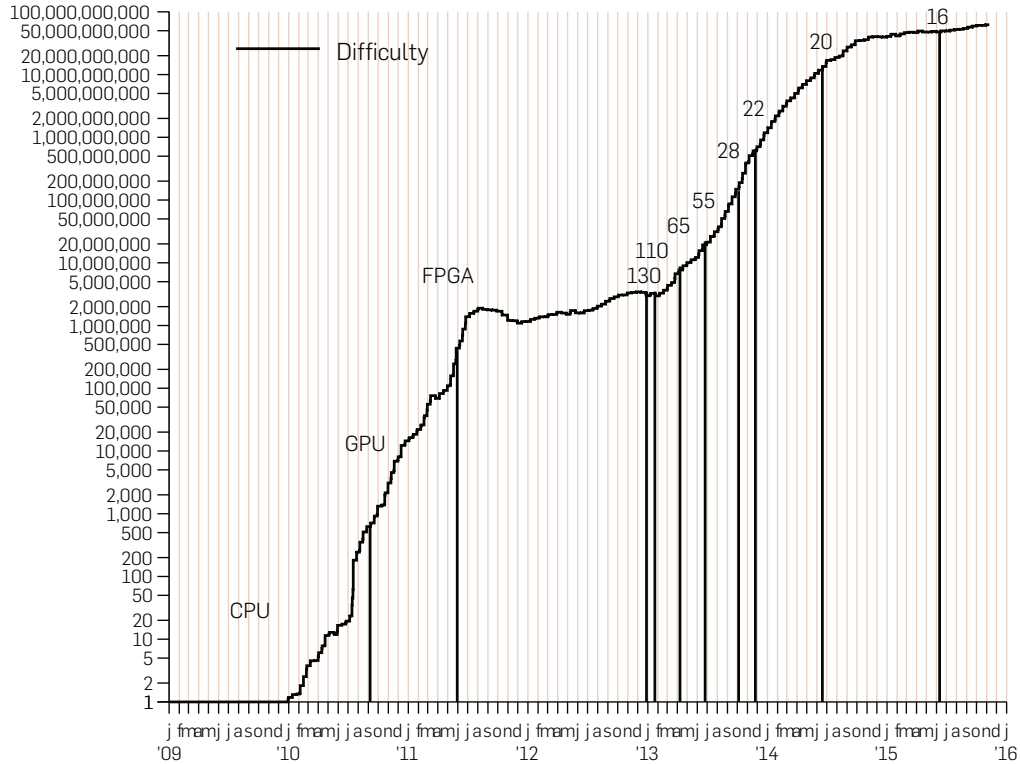
Video Transcoding, which converts from one video format to another, currently takes almost 30 high-end Xeon servers to do in real-time. As every cell phone can easily be a video source, as well as every Internet-of-Things device, it has the potential to be an unimaginably large planet-scale computation. Video Transcoding is an external memory-intensive application that needs DRAMs next to each ASIC and also high off-PCB bandwidth.

Finally, Deep Learning is extremely compute-intensive and is likely to be used by every human on the planet. Deep Learning is often latency-sensitive so our Deep Learning neural net accelerator has a tight low-latency SLA.

For our Bitcoin and Litecoin studies, we developed the RCA and got the required parameters such as gate count from placed and routed designs in UMC 28nm using Synopsys IC compiler and analysis tools (e.g., PrimeTime). For Deep Learning and Video Transcoding, we extract properties from accelerators designed in the research literature.

Design space exploration is application-dependent, and there are frequently additional constraints. For example, for video transcode application, we model the PCB real estate occupied by these DRAMs, which are placed on either side of the ASIC they connect to, perpendicular to airflow. As the number of DRAMs increases, the number of ASICs placed in a lane decreases for space reasons. We model the more expensive PCBs required by DRAM, with more layers and better signal/power integrity. We employ

Figure 4. Evolution of Specialization, Bitcoin cryptocurrency mining clouds. Numbers are ASIC nodes, in nm, which annotate the first date of release of a miner on that technology. Difficulty is the ratio of the total Bitcoin hash throughput of the world, relative to the initial mining network throughput, which was 7.15 MH/s. In the 6-year period preceding Nov 2015, the throughput increased by a factor of 50 billion times, corresponding to a world hash rate of approximately 575 million GH/s.

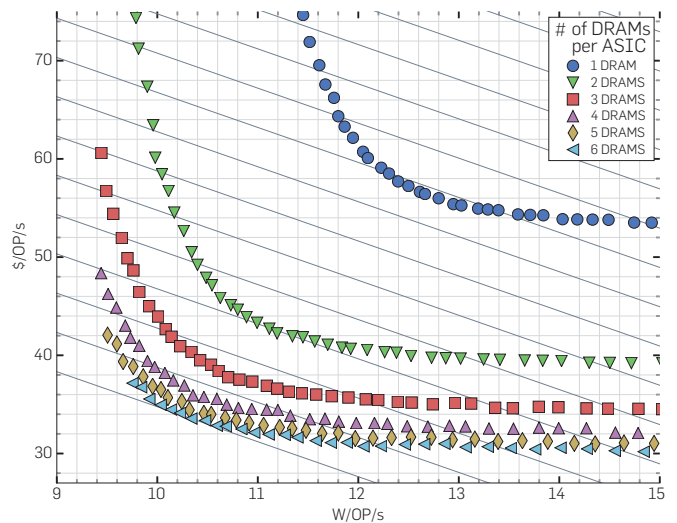


two 10-GigE ports as the off-PCB interface for network-intensive clouds, and model the area and power of the memory controllers.

After having all thermal constraints in place, we optimized ASIC server design targeting two conventional key metrics, namely cost per op/s and power per op/s, and then apply TCO analysis. TCO analysis incorporates the datacenter-level constraints such as the cost of power delivery inside the datacenter, land, depreciation, interest, and the cost of energy itself. With these tools, we can correctly weight these two metrics and find the overall optimal point (TCO-optimal) for the ASIC Cloud.

Our ASIC Cloud infrastructure explores a comprehensive design space, such as DRAMs per ASIC, logic voltage, area per ASIC, and number of chips. DRAM cost and power overhead are significant, and so the Pareto-optimal Video Transcoder designs ensure DRAM bandwidth is saturated, linked chip performance to DRAM count. As voltage and frequency are lowered, area increases to meet the performance requirement. Figure 5 shows the Video Transcode Pareto curve for 5 ASICs per lane and different number of DRAMs per ASIC. The tool is composed of two tiers. The top tier uses brute force to explore all of the possible configurations in order to find the energy-optimal, cost-optimal, and TCO-optimal points are chosen based on the Pareto results. The leaf tier consists of a variety of “expert solvers” that compute optimal properties of the server components; for example, CFD simulations for heat sinks, DC-DC

Figure 5. Pareto curve example for Video Transcode. Exploring different number of DRAMs per ASIC and logic voltage for optimal TCO per performance point. Voltage increases from left to right. Diagonal lines show equal TCO per performance values and the closer to the origin the lower the TCO per performance. This plot is for 5 ASICs per lane.



converter allocation, circuit area/delay/voltage/energy estimators, and DRAM property simulation. In many cases, these solvers export their data as large tables of memoized numbers for every component to the brute force solver.

Figure 6. ASIC Cloud optimization results for four applications. Each table presents energy-optimal, TCO-optimal, and cost optimal server properties. Energy optimal server uses lower voltage to increase the energy efficiency. Cost optimal servers use higher voltage to increase silicon efficiency. TCO-optimal has a voltage between these two and balances energy versus silicon cost.

	Energy optimal	TCO optimal	Cost optimal
ASICs per server	120	72	24
Logic Voltage (V)	0.400	0.459	0.594
Clock Freq. (MHz)	71	149	435
Die Area (mm ²)	599	540	240
GH/s/server	7,292	8,223	3,451
W/server	2,645	3,736	2,513
\$/server	12,454	8,176	2,458
W/GH/s	0.363	0.454	0.728
\$/GH/s	1.708	0.994	0.712
TCO/GH/s	3.344	2.912	3.686

(a) Bitcoin

	Energy optimal	TCO optimal	Cost optimal
ASICs per server	120	120	72
Logic Voltage (V)	0.459	0.656	0.866
Clock Freq. (MHz)	152	576	823
Die Area (mm ²)	600	540	420
MH/s/server	405	1,384	916
W/server	783	3,662	3,766
\$/server	10,971	11,156	6,050
W/MH/s	1.934	2.645	4.113
\$/MH/s	27.09	8.059	6.607
TCO/MH/s	37.87	19.49	23.70

(b) Litecoin

	Energy optimal	TCO optimal	Cost optimal
DEAMs per ASIC	3	6	9
ASICs per Server	64	40	32
Logic Voltage (V)	0.538	0.754	1.339
Clock Freq. (MHz)	183	429	600
Die Area (mm ²)	564	498	543
Kfps/server	126	158	189
W/server	1,146	1,633	3,101
\$/server	7,289	5,300	5,591
W/Kfps	9.073	10.34	16.37
\$/Kfps	57.68	33.56	29.52
TCO/Kfps	100.3	78.46	97.91

(c) Video Transcode

	Energy optimal	TCO optimal	Cost optimal
Chip type	4x2	2x2	2x1
ASICs per server	32	64	96
Logic Voltage (V)	0.900	0.900	0.900
Clock Freq. (MHz)	606	606	606
TOps/s/server	470	470	353
W/server	3,278	3,493	2,971
\$/server	7,809	6,228	4,146
W/TOps/s	6.975	7.431	8.416
\$/TOps/s	16.62	13.25	11.74
TCO/TOps/s	46.22	44.28	46.51

(d) Deep learning

5. RESULTS

Details of optimal server configurations for energy-optimal, TCO-optimal, and cost-optimal designs for each of the applications are shown in Figure 6.

For example, for Video Transcode, the cost-optimal server packs the maximum number of DRAMs per lane, 36, maximizing performance. However, increasing the number of DRAMs per ASIC requires higher logic voltage (1.34V) and corresponding frequencies to attain performance within the max die area constraint, resulting in less energy-efficient designs. Hence, the energy-optimal design has fewer DRAMs per ASIC and per lane (24), although gaining back some performance by increasing ASICs per lane, which is possible due to lower power density at 0.54V. The TCO-optimal design increases DRAMs per lane, 30, to improve performance, but is still close to the optimal energy efficiency at 0.75V, resulting in a die size and frequency between the other two optimal points.

In Figure 7, we compare the performance of CPU Clouds versus GPU Clouds versus ASIC Clouds for the four applications that we presented. ASIC Clouds outperform CPU Cloud TCO per op/s by 6270x; 704x; and 8695x for Bitcoin, Litecoin, and Video Transcode, respectively. ASIC Clouds outperform GPU Cloud TCO per op/s by 1057x, 155x, and 199x, for Bitcoin, Litecoin, and Deep Learning, respectively.

6. FEASIBILITY OF ASIC CLOUDS: THE TWO-FOR-TWO-RULE

When does it make sense to design and deploy an ASIC Cloud? The key barrier is the cost of developing the ASIC Server, which includes both the mask costs (about \$1.5M for the 28 nm node we consider here and much higher for the latest 7nm node) and the ASIC design costs, which collectively comprise the nonrecurring engineering expense (NRE). To understand this trade-off, we proposed the

two-for-two rule. If the cost per year (i.e., the TCO) for running the computation on an existing cloud exceeds the NRE by 2X, and you can get at least a 2X TCO per operation/second improvement, then going ASIC Cloud is likely to save money. Figure 8 shows a wider range of breakeven points. Essentially, as the TCO exceeds the NRE by more and more, the required speedup to break even declines. As a result, almost any accelerator proposed in the literature, no matter how modest the speedup, is a candidate for ASIC Cloud, depending on the scale of the computation. Our research makes the key contribution of noting that in deployment of ASIC Clouds, NRE and scale can be more determinative than

the absolute speedup of the accelerator. The main barrier for ASIC Clouds is to reign in NRE costs so they are appropriate for the scale of the computation. In many research accelerators, TCO improvements are extreme (such as in Figure 7), but authors often unnecessarily target expensive, latest generation process nodes because they are more cutting edge. This tendency raises the NRE exponentially, reducing economic feasibility. A better strategy is to target the older nodes that still attain sufficient TCO improvements.

7. POST-PUBLICATION INSIGHT: YOU WANT TO TARGET EIGHT TIMES TCO IMPROVEMENT

The two-for-two rule examines a lower bound for what the TCO improvements of an ASIC cloud need to be, based on how large the pre-ASIC cloud TCO is compared to the NRE of building an accelerator and show that extreme hundred times TCO improvements are not needed.

Our subsequent experience post-publication of the ASIC cloud suggests another way to look at the question of how aggressive an accelerator is necessary. We believe in most cases that eight times TCO improvement is usually a good place to target when developing a new kind of ASIC cloud.

In most realistic scenarios, the pre-ASIC cloud TCO can be in the hundreds of millions or billions of dollars, far out-shadowing the ASIC development costs for all but the latest nodes (e.g., 7nm). Practically speaking, the first two times will reduce your TCO in half, that is, one billion dollars become 500 million dollars. The second two times will only save 250 million dollars, useful but not essential on the first ASIC iteration. The second two times is needed to provide risk margin for the performance and energy efficiency

Figure 7. CPU Cloud vs. GPU Cloud vs. ASIC Cloud “Deathmatch.” ASIC servers greatly outperform the best non-ASIC alternative in terms of TCO per op/s.

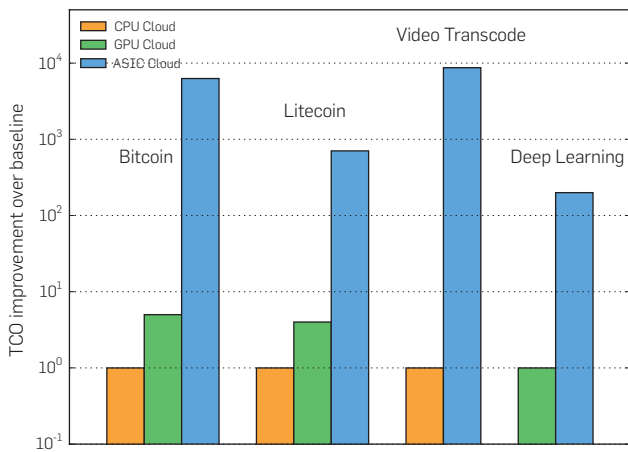
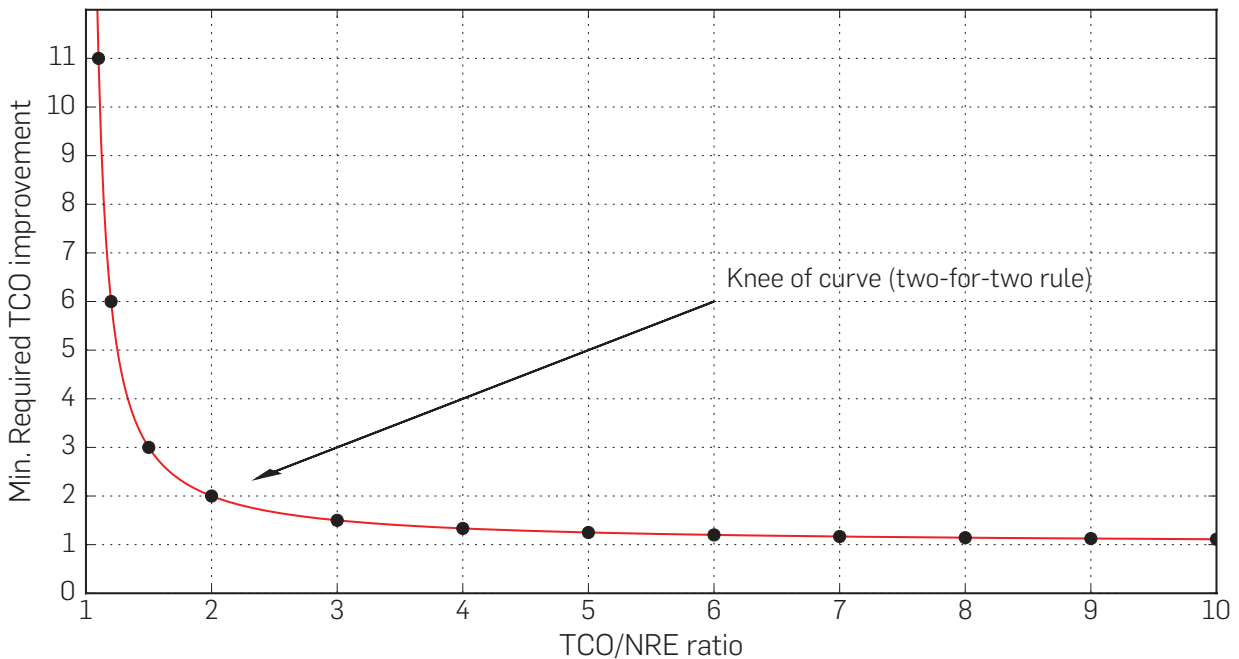


Figure 8. Two-for-two rule: moderate speed-up with low NRE beats high speed-up at high NRE. The points are break even points for ASIC Clouds.



uncertainty of the design—will the original software be optimized more making the chip less good relatively, will the chip have less than expected TCO improvement, et cetera. The final two times addresses the issue that the pre-ASIC cloud hardware (e.g., GPU or CPU) will also improve and could possibly improve by two times by the time you have deployed your ASIC cloud system.

8. CONCLUSION

Our research generalizes primordial Bitcoin ASIC Clouds into an architectural template that can apply across a range of planet-scale applications. Joint knowledge and control over datacenter and hardware design allow for ASIC Cloud designers to select the optimal design that optimizes energy and cost proportionally to optimize TCO. We demonstrated methodologies that can be used to design TCO-optimal clouds, answering long-standing questions even in contemporary Bitcoin ASIC Clouds. Our work analyses the impact of NRE and scale on deployment of ASIC Clouds, tying it to the TCO-improvement and in turn the energy and cost efficiency of the cloud.

Our work advances research practice by showing how to examine accelerators at a systems level instead of at the level of a single chip. We evaluate ASIC Cloud chip design, server design, and finally datacenter design in a cross-layer system-oriented way. This joint knowledge and control over datacenter and hardware design allow for ASIC Cloud designers to select the optimal design that optimizes energy and cost proportionally. We developed the tools and revealed how the designers of these novel systems can optimize the TCO in real-world ASIC Clouds.


We developed a rule of thumb for when it makes sense to go ASIC Cloud, the two-for-two rule. The main barrier for ASIC Clouds is to reign in NRE costs so they are appropriate for the scale of the computation. In many research accelerators, TCO improvements are extreme, but authors also target expensive, latest generation process nodes because they are more cutting edge. But this habit raises the NRE exponentially, reducing economic feasibility. Our most recent work⁶ suggests that a better strategy is to lower NRE cost by targeting older nodes that still have sufficient TCO per op/s benefit.

Looking to the future, our work suggests that both Cloud providers and silicon foundries would benefit by investing in technologies that reduce the NRE of ASIC design, such as open source IP such as RISC-V, in new labor-saving development methodologies for hardware and also in open source backend CAD tools. With time, mask costs fall by themselves, but currently older nodes such as 65 nm and 40 nm may provide suitable TCO per op/s reduction, with half the mask cost and only a small difference in performance and energy efficiency from 28, 16, or 7 nm. Foundries should take interest in ASIC Cloud's low-voltage scale out design patterns because they lead to greater silicon wafer consumption than CPUs within fixed environmental energy limits.

With the coming explosive growth of planet-scale computation, we must work to contain the exponentially growing environmental impact of datacenters across the world.

ASIC Clouds promise to help address this problem. By specializing the datacenter, they can do greater amounts of computation under environmentally determined energy limits. The future is planet-scale, and specialized ASICs will be everywhere.

Acknowledgments

This work was supported by both the JUMP ADA Center and the STARnet CFAR center, both funded by SRC and DARPA. Special thanks go to Partha Ranganathan for his support of our research over the years. 

References

1. ASIC clouds: Specializing the datacenter. *UCSD CSE Tech Report CS2016-1016*. May 8, 2016. https://csetechrep.ucsd.edu/Dienst/UI/2.0/Describe/ncstrLucsd_cse/CS2016-1016.
2. Esmaeilzadeh, H., Blem, E., Amant, R.S., Sankaralingam, K., Burger, D. Power limitations and dark silicon are challenging the future of multicore. In *TOCS*, 2012.
3. Goulding, N., et al. GreenDroid: A mobile application processor for a future of dark silicon. In *HOTCHIPS*, 2010.
4. Goulding-Hotta, N., Sampson, J., Venkatesh, G., Garcia, S., Auricchio, J., Huang, P.-C., Arora, M., Nath, S., Bhatt, V., Babb, J., Swanson, S., Taylor, M.B. The greendroid mobile application processor: An architecture for silicon's dark future. *IEEE Micro* 2, 31 (2011), 86–95.
5. Jouppi, N.P., et al. In-datacenter performance analysis of a tensor processing unit. In *International Symposium on Computer Architecture (ISCA)*, 2017.
6. Khazraee, M., et al. Moonwalk: NRE optimization in ASIC clouds. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2017.
7. Khazraee, M., Vega, L., Magaki, I., Taylor, M. Specializing a planet's computation: ASIC clouds. *IEEE Micro*, May 2017.
8. Magaki, I., et al. ASIC clouds: Specializing the datacenter. In *International Symposium on Computer Architecture (ISCA)*, 2016.
9. Putnam et al. A reconfigurable fabric for accelerating large-scale datacenter services. In *International Symposium on Computer Architecture (ISCA)*, 2014.
10. Sampson, J., Venkatesh, G., Goulding-Hotta, N., Garcia, S., Swanson, S., Taylor, M.B. Efficient complex operators for irregular codes. In *HPCA*, 2011.
11. Shafique, M., Garg, S., Henkel, J., Marculescu, D. The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In *DAC*, 2014.
12. Weston, S. FPGA accelerators at JP Morgan chase, 2011. Stanford Computer Systems Colloquium, <https://www.youtube.com/watch?v=9NqX1ETADn0>.
13. Taylor, M. A landscape of the new dark silicon design regime. *IEEE Micro*, Sept-Oct. 2013.
14. Taylor, M.B. Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse. In *DAC*, 2012.
15. Venkatesh, G., Sampson, J., Goulding, N., Garcia, S., Bryksin, V., Lugo-Martinez, J., Swanson, S., Taylor, M.B. Conservation cores: Reducing the energy of mature computations. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2010.
16. Venkatesh and others. Qscores: Configurable co-processors to trade dark silicon for energy efficiency in a scalable manner. In *MICRO*, 2011.
17. Xie, S., Davidson, S., Magaki, I., Khazraee, M., Vega, L., Zhang, L., Taylor, M.B. Extreme datacenter specialization for planet-scale computing: ASIC clouds. *SIGOPS Oper. Syst. Rev.* 1, 52 (2018), 96–108.

Michael Bedford Taylor (prof.taylor@gmail.com), University of Washington, WA, USA.

Luis Vega (vegaluis@cs.washington.edu), University of Washington, WA, USA.

Mooin Khazraee (mkhazrae@cs.ucsd.edu), UC San Diego, CA, USA.

Ikuo Magaki (ikuomagaki@icloud.com), UC San Diego, CA, USA.

Scott Davidson and **Dustin Richmond** ({stdavids, dustinar}@uw.edu), University of Washington, WA, USA.

volume
01

number
01

FIRST
ISSUE
PUBLISHED

ACM Transactions on Internet of Things
is now available in
the ACM Digital Library



ACM Transactions on Internet of Things (TIOT) publishes novel research contributions and experience reports in several research domains whose synergy and interrelations enable the IoT vision. TIOT focuses on system designs, end-to-end architectures, and enabling technologies, and on publishing results and insights corroborated by a strong experimental component.

Publish Your Work Open Access With ACM!

ACM offers a variety of Open Access publishing options to ensure that your work is disseminated to the widest possible readership of computer scientists around the world.



Please visit ACM's website to learn more about ACM's innovative approach to Open Access at:
www.acm.org/openaccess



Association for
Computing Machinery

[CONTINUED FROM P. 112] on a slower downstream flow than on a faster downstream flow. Show that swapping so that you paddle on a faster downstream flow reduces the time. For example, consider the proof provided by my colleague Ernie Davis: http://cs.nyu.edu/cs/faculty/shasha/papers/ernieproof_canoe.pdf

Challenge: Consider the situation show in the second figure here.

a) If the paddler P can paddle with a water speed of two kilometers per hour only, then what is the best route P can take and how long will it take?

b) If the paddler P can paddle with a water speed of three kilometers per hour for one hour but two kilometers per hour at all other times, then what is the best route P can take and how long will it take?

c) If the paddler P can paddle with a water speed of three kilometers per hour for two hours but two kilometers per hour at all other times, then what is the best route P can take and how long will it take?

d) If the paddler P can paddle with a water speed of three kilometers per hour for three hours but two kilometers per hour at all other times, then what is the best route P can take and how long will it take?

Solutions:

d. ABCDEF: nine hours

e. ABCEF: five hours (paddles at three kilometers per hour between E and F)

f. ACEF: four hours (paddles at three kilometers per hour between A and C and between E and F)

g. ACEF or AEF: three hours (paddles at three kilometers the whole time).

Paddling Upstart: Given a general network of different downstream flows and a variety of paddle speeds with their durations, what is the best route to take and at which speeds? Please design an algorithm and provide an implementation in some widely used computer language (or several:).

Dennis Shasha (dennisshasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, USA, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at <http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html>

Copyright held by author.



Dennis Shasha

DOI:10.1145/3401749

Upstart Puzzles

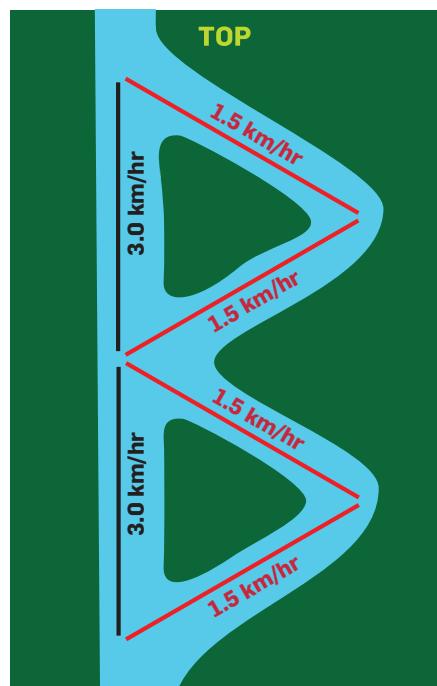
Strategic Paddling

Choosing how to best navigate turbulent current events.

A STRONG CANOE paddler can achieve a water speed of about nine kilometers per hour. Most paddlers can achieve water speeds of at least two kilometers per hour pretty much indefinitely.

If a waterway is flowing against the paddler at speed w and the paddler has a water speed of s , then the paddler will achieve a land speed of $s - w$. This has consequences.

Warm-Up: Consider a system of waterways like that in the first illustration in this column, where each segment is one kilometer long, the segments in black flow downstream at three kilometers per hour, and the segments in red



A canoe paddler can achieve a water speed of four kilometers per hour (km/hr) for one hour and two kilometers per hour indefinitely. What is the fastest way to go from the downstream point to the top point?

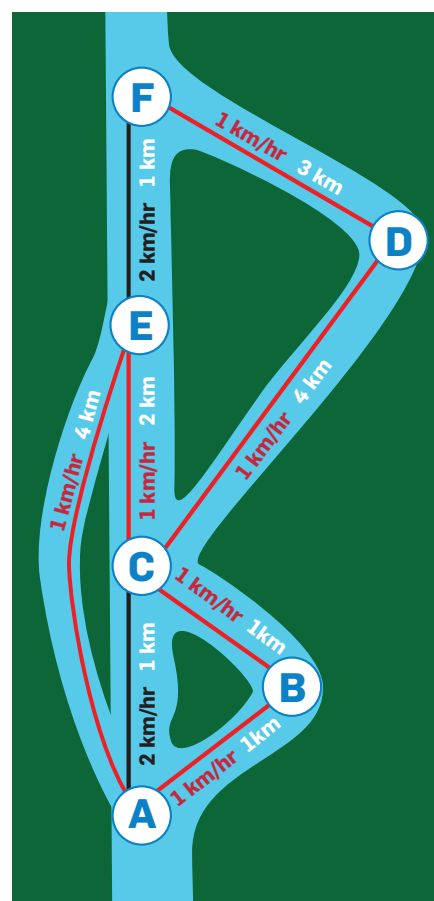
flow downstream at 1.5 kilometers per hour. Suppose a paddler is capable of paddling for a total of one hour at four kilometers per hour and then indefinitely at two kilometers per hour. What is the best route to choose from the bottom to the top and how long will that take?

Solution to Warm-Up: Paddle at four kilometers per hour (water speed) on the black segment, reaching the middle point after one hour. Then paddle at two kilometers per hour (water speed) on the top red segments. This will take another four hours to reach the top because the effective land speed is only 0.5 kilometers per hour.

OK, I think you are ready for more now.

Challenge: Suppose there is only one route of four kilometers along which the first two kilometers flow downstream at one kilometer per hour and the second two-kilometer segment flows downstream at two kilometers per hour. If a paddler can paddle at four kilometers per hour for one hour and three kilometers per hour for another hour, how should the paddler paddle to finish the course as fast as possible?

Solution: In the first hour, the paddler achieves a water speed of four kilometers per hour on the two-kilometer long, two kilometers per hour downstream segment thus finishing that segment. In the second hour, the paddler achieves a water speed of three kilometers per hour on the two kilometer-long one kilometer per hour downstream segment, thus completing the route. If the paddler paddles at three kilometers an hour on the first segment and then four kilometers an hour on the second segment, the time will be $2 \frac{1}{3}$ hours.



A network of waterways. Depending on the strength of the paddler (see text), the paddler may take different routes.

Challenge: Prove that if there is only a single route, then the paddler will reach the destination earlier by paddling as fast as possible on the fastest downstream flowing part as long as possible?

Proof strategy: Imagine there is another strategy in which it is better to paddle faster [CONTINUED ON P. 111]

volume

01

number

01

FIRST

ISSUE

PUBLISHED

ACM Transactions on Computing for Healthcare is now available in the ACM Digital Library



ACM Transactions on Computing for Healthcare (HEALTH) is a peer-reviewed journal for the publication of high-quality original research papers, survey papers, and challenge papers that have scientific and technological results pertaining to how computing is improving healthcare. HEALTH is multidisciplinary, intersecting CS, ECE, mechanical engineering, bio-medical engineering, behavioral and social science, psychology, and the health field, in general.



SIGGRAPH ASIA 2020 DAEGU

The 13th ACM SIGGRAPH Conference and
Exhibition on Computer Graphics and Interactive
Techniques in Asia

Conference 17 – 20 November 2020

Exhibition 18 – 20 November 2020

EXCO, Daegu, South Korea

Driving Diversity

SA2020.SIGGRAPH.ORG

[#SIGGRAPHAsia](https://twitter.com/SIGGRAPHAsia) | [#SIGGRAPHAsia2020](https://twitter.com/SIGGRAPHAsia2020)



Sponsored by



Organized by

