

31st IEEE INTERNATIONAL Parallel and Distributed Processing SYMPOSIUM



May 29-June 2, 2017
Buena Vista Palace Hotel
Orlando, Florida USA

www.ipdps.org

Orlando is home to a rich offering of indoor and outdoor attractions. Located a mile from Walt Disney World® and 4 miles from Epcot, the Buena Vista Palace Hotel is a 5-minute walk from Downtown Disney with a complimentary shuttle to all Disney Theme Parks and Water Parks. The sprawling Lake Buena Vista resort offers a full menu of amenities and family friendly activities as well as ideal meeting space for IPDPS 2017.

IPDPS 2017 CALL FOR PAPERS

Authors are invited to submit manuscripts that present original unpublished research in all areas of parallel and distributed processing, including the development of experimental or commercial systems. Work focusing on emerging technologies and interdisciplinary work covering multiple IPDPS areas are especially welcome. During submission, authors can indicate up to three subject areas that can come from any track. Topics of interest include, but are not limited to:

- **Parallel and distributed algorithms**, focusing on topics such as: numerical and combinatorial parallel algorithms for analysis, machine learning and simulation; parallel algorithms for accelerators, neuromorphic architectures, and other non-traditional systems; algorithms for cloud computing; power-aware parallel algorithms; streaming algorithms; domain-specific parallel and distributed algorithms; performance modeling and analysis of parallel and distributed algorithms; run-time algorithms and protocols for resource management, communication and synchronization on parallel and distributed systems..
- **Applications of parallel and distributed computing**, including computational and data-enabled science and engineering, big data applications, parallel crowd sourcing, large-scale social network analysis, management of big data, cloud and grid computing, scientific, biological and medical applications, and mobile computing. Papers focusing on applications using novel commercial or research architectures, big data approaches, or discussing scalability toward the exascale level are encouraged.
- **Parallel and distributed architectures**, including architectures for instruction-level and thread-level parallelism; petascale and exascale systems designs; novel big data architectures; special purpose architectures, including graphics processors, signal processors, network processors, media accelerators, and other special purpose processors and accelerators; impact of technology on architecture; network and interconnect architectures; parallel I/O and storage systems; architecture of the memory hierarchy; power-efficient and green computing architectures; dependable architectures; and performance modeling and evaluation.
- **Parallel and distributed software**, including parallel and multicore programming languages and compilers, runtime systems, operating systems, resource management including, middleware for supercomputers, grids, clouds, and data centers, libraries, performance modeling and evaluation, parallel programming paradigms, and programming environments and tools. Papers focusing on novel software systems for big data and exascale systems are encouraged.

GENERAL CHAIR

Michela Taufer
(University of Delaware, USA)

PROGRAM CHAIR

Marc Snir
(University of Illinois at Urbana Champaign, USA)

PROGRAM VICE-CHAIRS

- **Algorithms:**
Pierre Fraigniaud (IRIF, France)
- **Applications:**
Robert D. Moser (UT Austin, USA)
- **Architecture:**
Hillery Hunter (IBM Research, USA)
& Robert Senger (IBM Research, USA)
- **Software:**
Pavan Balaji (Argonne National Lab, USA)
- **Multidisciplinary:**
Torsten Hoefler (ETH Zurich, Switzerland)

KEYNOTES & TECHNICAL SESSIONS

WORKSHOPS & PHD FORUM

COMMERCIAL PARTICIPATION

Details at www.ipdps.org

IMPORTANT DATES

October 18, 2016	Submit Abstract
October 23, 2016	Submit Paper
Nov 28 - Dec 5, 2016	Review Feedback & Author Response
January 8, 2017	Author Notification
After January 8, 2017	Deadlines for Paper Submissions to Most Workshops

SPONSORED BY:



Technical Committee on Parallel Processing

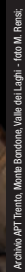
IN COOPERATION WITH:



ACM SIGARCH & SIGHPC

IEEE Computer Society Technical Committee on Computer Architecture

IEEE Computer Society Technical Committee on Distributed Processing



Raytheon
BBN Technologies

Departments

- 5 **Editor's Letter**
Academic Rankings Considered Harmful!
By Moshe Y. Vardi
-
- 7 **Cerf's Up**
Keeping the Internet Open
By Vinton G. Cerf
-
- 8 **Letters to the Editor**
Election Auditing and Verifiability
-
- 25 **Calendar**
-
- 98 **Careers**

Last Byte

- 104 **Q&A**
Hello, Underworld
Stefan Savage's innovative research has focused on strengthening the security, privacy, and reliability of networks.
By Leah Hoffmann

News



- 11 **Reconciling Quantum Physics with Math**
Mathematicians explore the root of many problems in developing a proof for the Kadison-Singer problem.
By Chris Edwards
-
- 14 **GPUs Reshape Computing**
Graphical processing units have emerged as a major powerhouse in the computing world, unleashing huge advancements in deep learning and AI.
By Samuel Greengard
-
- 17 **The Edge of the Uncanny**
Scientists are learning more about what makes robots and chatbots engaging.
By Gregory Mone

Viewpoints

- 20 **Law and Technology**
No Easy Answers in the Fight Over iPhone Decryption
A look at the legal background and future possibilities for an issue that is likely to reoccur.
By Felix Wu
-
- 23 **The Profession of IT**
Software Quality
Software users are looking more and more for software that delights.
By Peter J. Denning
-
- 26 **Broadening Participation**
"For All" in "Computer Science For All"
Seeking to expand inclusiveness in computer science education.
By Richard E. Ladner and Maya Israel
-
- 29 **Viewpoint**
Designing AI Systems that Obey Our Laws and Values
Calling for research on automatic oversight for artificial intelligence systems.
By Amitai Etzioni and Oren Etzioni
-
- 32 **Viewpoint**
Helping Conference Attendees Better Understand Research Presentations
Sharing lessons learned from a lecture program for making technical material more accessible to conference attendees.
By Ethan Katz-Bassett et al.
-
- 35 **Viewpoint**
A New Look at the Semantic Web
Seeking to make Web data "smarter" by utilizing a new kind of semantics.
By Abraham Bernstein, James Hendler, and Natalya Noy

Practice



44

38 **Introducing Research for Practice**

Expert-curated guides to the best of CS research.

By Peter Bailis, Simon Peter, Justine Sherry

42 **Bad Software Architecture Is a People Problem**

When people don't work well together they make bad decisions.

By Kate Matsudaira

44 **10 Optimizations on Linear Search**

The operations side of the story.

By Thomas A. Limoncelli



Articles' development led by **acmqueue**
queue.acm.org

**About the Cover:**

Information once found only in publications is now housed just as often in complex datasets and databases. The ability to cite digital information in the same manner as conventional citations has frustrated authors. This month's cover story explores why data citation is a computational problem and presents how to generate data citations automatically. Cover illustration by Yippiehey.

Contributed Articles



58

50 **Why Data Citation Is a Computational Problem**

Using database views to specify citable units is the key to generating citations to data.

By Peter Buneman, Susan Davidson, and James Frew



Watch the author discuss her work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/why-data-citation-is-a-computational-problem>

58 **Dynamic Presentation Consistency Issues in Smartphone Mapping Apps**

Smartphone mapping apps routinely fail to follow centuries-old mapmaking dynamic consistency principles and practices.

By Hanan Samet, Sarana Nutanong, and Brendan C. Fruin

Review Articles

68 **Learning Executable Semantic Parsers for Natural Language Understanding**

Semantic parsing is a rich fusion of the logical and the statistical worlds.

By Percy Liang

Research Highlights

78 **Technical Perspective**
The Dawn of Computational Light Transport

By Kyros Kutulakos

79 **Imaging the Propagation of Light through Scenes at Picosecond Resolution**
By Andreas Velten, Di Wu, Belen Masia, Adrian Jarabo, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Mounqi Bawendi, Diego Gutierrez, and Ramesh Raskar

Watch the author discuss his work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/imaging-the-propagation-of-light-through-scenes-at-picosecond-resolution>

87 **Technical Perspective**
Jupiter Rising

By Andrew W. Moore

88 **Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network**

By Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Hong Liu, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat



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

Bobby Schnabel

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

Bernard Rous

Director, Office of Group Publishing

Scott E. Delman

ACM COUNCIL

President

Alexander L. Wolf

Vice-President

Vicki L. Hanson

Secretary/Treasurer

Erik Altman

Past President

Vinton G. Cerf

Chair, SGB Board

Patrick Madden

Co-Chairs, Publications Board

Jack Davidson and Joseph Konstan

Members-at-Large

Eric Allman; Ricardo Baeza-Yates;

Cherri Pancake; Radia Perlman;

Mary Lou Soffa; Eugene Spafford;

Per Stenström

SGB Council Representatives

Paul Beame; Jenna Neefe Matthews;

Barbara Boucher Owens

BOARD CHAIRS

Education Board

Mehran Sahami and Jane Chu Prey

Practitioners Board

George Neville-Neil

REGIONAL COUNCIL CHAIRS

ACM Europe Council

Dame Professor Wendy Hall

ACM India Council

Srinivas Padmanabhuni

ACM China Council

Jiaquan Sun

PUBLICATIONS BOARD

Co-Chairs

Jack Davidson; Joseph Konstan

Board Members

Ronald F. Boisvert; Karin K. Breitman;

Terry J. Coatta; Anne Condon; Nikil Dutt;

Roch Guérin; Carol Hutchins;

Yannis Ioannidis; Catherine McGeoch;

M. Tamer Özsu; Mary Lou Soffa; Alex Wade;

Keith Webster

ACM U.S. Public Policy Office

Renee Dopplick, Director

1828 L Street, N.W., Suite 800

Washington, DC 20036 USA

T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association

Mark R. Nelson, Executive Director

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.

STAFF

DIRECTOR OF GROUP PUBLISHING

Scott E. Delman

cacm-publisher@cacm.acm.org

Executive Editor

Diane Crawford

Managing Editor

Thomas E. Lambert

Senior Editor

Andrew Rosenbloom

Senior Editor/News

Larry Fisher

Web Editor

David Roman

Rights and Permissions

Deborah Cotton

Art Director

Andrij Borys

Associate Art Director

Margaret Gray

Assistant Art Director

Mia Angelica Balaquiot

Designer

Iwona Usakiewicz

Production Manager

Lynn D'Addesio

Director of Media Sales

Jennifer Ruzicka

Publications Assistant

Juliet Chance

Columnists

David Anderson; Phillip G. Armour;

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

acmhlp@acm.org

Letters to the Editor

letters@cacm.acm.org

WEBSITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/

ACM ADVERTISING DEPARTMENT

2 Penn Plaza, Suite 701, New York, NY

10121-0701

T (212) 626-0686

F (212) 869-0481

Director of Media Sales

Jennifer Ruzicka

jen.ruzicka@hq.acm.org

For display, corporate/brand advertising:

Craig Pitcher

pitcherc@acm.org T (408) 778-0300

William Sleight

wsleight@acm.org T (408) 513-3408

Media Kit acmm mediasales@acm.org

Association for Computing Machinery (ACM)

2 Penn Plaza, Suite 701

New York, NY 10121-0701 USA

T (212) 869-7440; F (212) 869-0481

EDITORIAL BOARD

EDITOR-IN-CHIEF

Moshe Y. Vardi

eric@cacm.acm.org

NEWS

Co-Chairs

William Pulleyblank and Marc Snir

Board Members

Mei Kobayashi; Michael Mitzenmacher;

Rajeev Rastogi

VIEWPOINTS

Co-Chairs

Tim Finin; Susanne E. Hambrusch;

John Leslie King

Board Members

William Aspray; Stefan Bechtold;

Michael L. Best; Judith Bishop;

Stuart I. Feldman; Peter Freeman;

Mark Guzdial; Rachelle Hollander;

Richard Ladner; Carl Landwehr;

Carlos Jose Pereira de Lucena;

Beng Chin Ooi; Loren Terveen;

Marshall Van Alstyne; Jeannette Wing

Q PRACTICE

Co-Chair

Stephen Bourne

Board Members

Eric Allman; Peter Bailis; Terry Coatta;

Stuart Feldman; Benjamin Fried;

Pat Hanrahan; Tom Killalea; Tom Limoncelli;

Kate Matsudaira; Marshall Kirk McKusick;

George Neville-Neil; Theo Schlossnagle;

Jim Waldo

The Practice section of the CACM

Editorial Board also serves as

the Editorial Board of *Q PRACTICE*.

CONTRIBUTED ARTICLES

Co-Chairs

Andrew Chien and James Larus

Board Members

William Aiello; Robert Austin; Elisa Bertino;

Gilles Brassard; Kim Bruce; Alan Bundy;

Peter Buneman; Peter Druschel; Carlo Ghezzi;

Carl Gutwin; Yannis Ioannidis;

Gal A. Kaminka; James Larus; Igor Markov;

Gail C. Murphy; Bernhard Nebel;

Lionel M. Ni; Kenton O'Hara; Sriram Rajamani;

Marie-Christine Rousset; Avi Rubin;

Krishan Sabnani; Ron Shamir; Yoav

Shoham; Larry Snyder; Michael Vitale;

Wolfgang Wahlster; Hannes Werthner;

Reinhard Wilhelm

RESEARCH HIGHLIGHTS

Co-Chairs

Azer Bestavros and Gregory Morrisett

Board Members

Martin Abadi; Amr El Abbadi; Sanjeev Arora;

Nina Balcan; Dan Boneh; Andrei Broder;

Doug Burger; Stuart K. Card; Jeff Chase;

Jon Crowcroft; Sandhya Dwaekadas;

Matt Dwyer; Alon Halevy; Norm Jouppi;

Andrew B. Kahng; Sven Koenig; Xavier Leroy;

Steve Marschner; Kobbi Nissim;

Steve Seitz; Guy Steele, Jr.; David Wagner;

Margaret H. Wright; Andreas Zeller

WEB

Chair

James Landay

Board Members

Marti Hearst; Jason I. Hong;

Jeff Johnson; Wendy E. MacKay

ACM Copyright Notice

Copyright © 2016 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 acmhlp@acm.org.

COMMUNICATIONS OF THE ACM

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

POSTMASTER

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

Printed in the U.S.A.



Association for
Computing Machinery





Moshe Y. Vardi

DOI:10.1145/2980760

Academic Rankings Considered Harmful!

ACADEMIC RANKINGS HAVE a huge presence in academia. College rankings by *U.S. News and World Report* (USNWR) help undergraduate students find the “perfect school.” Graduate-program rankings by USNWR are often the most significant decision-making factor for prospective graduate students. The Academic Ranking of World Universities (known also as the “Shanghai Ranking”) is one that attracts much attention from university presidents and governing boards. New academic rankings, of many different forms and flavors, have been popping up regularly over the last few years.

Yet, there is also deep dissatisfaction in the academic community with the methodology of such rankings and with the outsize role that commercial entities play in the ranking business. The recent biennial meeting of the Computing Research Association (CRA) dedicated a session to this topic (see <http://cra.org/events/snowbird-2016/#agenda>), asserting that “Many members of our community currently feel the need for an authoritative ranking of CS departments in North America” and asking “Should CRA be involved in creating a ranking?” The rationale for that idea is the computing-research community will be better served by helping to create some “sensible rankings.”

The methodology currently used by USNWR to rank computer-science graduate program is highly questionable. This ranking is based solely on “reputational standing” in which department chairs and graduate directors are asked to rank each graduate program on a 1–5 scale. Having participated in such reputational surveys for many years, I can testify that I spent about a second or two coming

up with a score for the over 100 ranked programs. Obviously, very little contemplation went into my scores. In fact, my answers have clearly been influenced by prior-year rankings. It is a well-known “secret” that rankings of graduate programs of universities of outstanding reputation are buoyed by the halo effect of their parent institutions’ reputations. Such reputational rankings have no academic value whatsoever, I believe, though they clearly play a major role in academic decision making.

But the problem is deeper than the current flawed methodology of USNWR’s ranking of graduate programs. Academic rankings, in general, provide highly misleading ways to inform academic decision making by individuals. An academic program or unit is a highly complex entity with numerous attributes. An academic decision is typically a multi-objective optimization problem, in which the objective function is highly personal. A unidimensional ranking provides a seductively easy objective function to optimize. Yet such decision making ignores the complex interplay between individual preferences and programs’ unique patterns of strengths and weaknesses. Decision making by ranking is decision making by lazy minds, I believe.

Furthermore, academic rankings have adverse effects on academia. Such rankings are generally computed by devising a mapping from the complex space of program attributes to a unidimensional space. Clearly, many such mappings exist. Each ranking is based on a specific “methodology,” that is, a specific ranking mapping. The choice of mapping is completely arbitrary and reflects some “judgment” by the ranking organization.

But the academic value of such a judgment is dubious. Furthermore, commercial ranking organizations tweak their mappings regularly in order to create movement in the rankings. After all, if you are in the business of selling ranking information, then you need movement in the rankings for the business to be viable. Using such rankings for academic decision making is letting third-party business interests influence our academic values.

Thus, to the question “Should CRA get involved in creating a ranking?” my answer is “absolutely not.” I do not believe that “sensible rankings” can be defined. The U.S. National Research Council’s attempt in 2010 to come up with an evidence-based ranking mapping is widely considered a notorious failure. Furthermore, I believe the CRA should pass a resolution encouraging its members to stop participating in the USNWR surveys and discouraging students from using these rankings for their own decision making. Instead, CRA should help well-informed academic decision making by creating a data portal providing public access to relevant information about graduate programs. Such information can be gathered from an extended version of the highly respected Taulbee Survey that CRA has been running for over 40 years, as well as from various open sources. CRA could also provide an API to enable users to construct their own ranking based on the data provided.

Academic rankings are harmful, I believe. We have a responsibility to better inform the public, by ceasing to “play the ranking games” and by providing the public with relevant information. The only way to do that is by asserting the collective voice of the computing-research community.

Follow me on Facebook, Google+, and Twitter.

Moshe Y. Vardi, EDITOR-IN-CHIEF

Copyright held by author.

SHAPE THE FUTURE OF COMPUTING.

JOIN ACM TODAY.

ACM is the world's largest computing society, offering benefits and resources that can advance your career and enrich your knowledge. We dare to be the best we can be, believing what we do is a force for good, and in joining together to shape the future of computing.

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

- ☐ Professional Membership: \$99 USD
- ☐ Professional Membership plus
ACM Digital Library: \$198 USD (\$99 dues + \$99 DL)
- ☐ ACM Digital Library: \$99 USD
(must be an ACM member)

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 all aspects of the computing field. Available at no additional cost.

Priority Code: CAPP

Payment Information

Name _____

ACM Member # _____

Mailing Address _____

City/State/Province _____

ZIP/Postal Code/Country _____

Email _____

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc., in U.S. dollars or equivalent in foreign currency.

☐ AMEX ☐ VISA/MasterCard ☐ Check/money order

Total Amount Due _____

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

Return completed application to:
ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Satisfaction Guaranteed!

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.



Association for
Computing Machinery

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
acm.org/join/CAPP



DOI:10.1145/2980762

Vinton G. Cerf

Keeping the Internet Open

The Internet has always been open, or so it has been for much of its existence. Open to new ideas, new protocols, new applications, and new technology. But not everyone or

every regime sees the bounty of freedom of expression and invention. Some see an open wound into which every source of pestilence pours. Threatened by sunlight, authoritarian governments seek to shutter freedom of access to this creative infrastructure. They point to the worst content, the harmful malware and the cyber-attacks that damage the users of the Internet to justify their policies and rhetoric. And, in this, they are not entirely wrong. There is malware and there are serious cyber-attacks and weaknesses. Some regimes, while speaking out against these challenges, even contribute to them by sanctioning or launching their own attacks.

Those who maintain, operate, and evolve the Internet have an obligation to find ways to protect its users from these very threats. Those who offer and who benefit from the cornucopia of applications the Internet can support have an opportunity to contribute to the safety, security, privacy, and reliability of the Internet upon which so many now depend and so many more wish to have access.

To those who have spoken for and who espouse an open and welcoming Internet now falls the task of keeping it that way. To do so will require more than words and argument. Substantive improvements in the Internet's resilience and resistance to attack are needed. Firewalls and passwords are not enough. Strong authentication, cryptography, and attack-resistant operating systems, browsers, and other applications are needed. Better programming environments that produce less buggy,

exploitable software are necessary and await invention or use.

Nor can the users of the Internet escape some responsibility. Their poor choices of passwords and naïve clicking on phishing email messages, with their virus-laden content, open pathways for those with harmful intent. We have an obligation to teach safe network usage and practices to all who look for positive utility in this global network of networks. In a society of laws, the freedom we enjoy comes in part from protection against the harmful actions of law-breakers. We rely, in part, on enforcement of the law to maintain the safety that we, and our fellow Internauts, wish to enjoy. But in some societies, laws that limit freedom of expression, peaceful assembly, and access to information stifle the benefits that derive from freedom, and mock the very openness that lies at the heart of the Internet. Irrational responses to real and perceived potential harms lead to unenforceable legal regimes and do more damage than good.

Readers of this column should feel some measure of responsibility to fashion new and apply known means to increase the safety and security of the Internet while fostering the evolution of new and increasingly useful applications. In some very direct sense, computer science and engineering house the knowledge and methods needed to achieve these goals. Designs and practices that resist cyber-attacks of all kinds are needed. But this challenge will not be successfully met solely with technology. Transnational standards of behavior will be needed

for individuals, corporations, and even nations, if we are to keep the Internet open and to avoid catastrophic closing of its boundless shores.

As the Internet infrastructure becomes increasingly vital to the global economy and well-being of Internauts everywhere, nations will need new practices to respond to perceived national security threats. Cyber-hotlines, like those created in the tense days of the Cold War, may prove essential to avoid offensive cyber or even military responses to cyber-attacks subject to misattribution of origin. Misunderstandings between nations or even between domestic citizens and law enforcement can visit untold harm on everyone and lasting damage to freedom in its many dimensions. Just as DNA analysis has proven to be a powerful, if not always perfect, tool for truth, a cyber-counterpart may be needed to aid in the identification of criminal actors pursuing their aims along domestic or transnational courses.

What we should not and must not tolerate is the arbitrary shutting down of pathways that can and do link together our increasingly global societies. Discovery of diverse cultural, literary, and historical heritage, and their digital preservation should animate our purpose in keeping the Internet open. Online facilitation of commerce of all kinds has enhanced our lives. Every person on the planet should have the freedom to access and to contribute to the increasing utility of the Internet. Four decades have passed since its invention and we still have work ahead to assure its utility for many decades to come until it, too, is replaced by something even better and more beneficial. 

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

Copyright held by author.

Election Auditing and Verifiability

OVERALL, THE INSIDE RISKS Viewpoint “The Risks of Self-Auditing Systems” by Rebecca T. Mercuri and Peter G. Neumann (June 2016) was excellent, and we applaud its call for auditing systems by independent entities to ensure correctness and trustworthiness. However, with respect to voting, it said, “Some research has been devoted to end-to-end cryptographic verification that would allow voters to demonstrate their choices were correctly recorded and accurately counted. However, this concept (as with Internet voting) enables possibilities of vote buying and selling.” This statement is incorrect.

While Internet voting (like any remote-voting method) is indeed vulnerable to vote buying and selling, end-to-end verifiable voting is not. Poll-site-based end-to-end verifiable voting systems use cryptographic methods to ensure voters can verify their own votes are correctly recorded and tallied while (paradoxically) not enabling them to demonstrate how they voted to anyone else.

Mercuri and Neumann also said, “[end-to-end verifiability] raises serious questions of the correctness of the cryptographic algorithms and their implementation.” This sentence is potentially misleading, as it suggests confidence in the correctness of the election outcome requires confidence in the correctness of the implementation of the cryptographic algorithms. But end-to-verifiable voting systems are designed to be “fail safe”; if the cryptographic algorithms in the voting system are implemented incorrectly, the audit will indeed fail. Poor crypto implementations in the voting system will not allow an audit to approve an incorrect election outcome.

Finally, we note that end-to-end verifiable election methods are a special case of “verifiable computation,” whereby a program can produce not only a correct result but also a “proof” that it is the correct result for the given inputs. Of course, the inputs need to be agreed upon before such a proof makes sense.

Such methods may thus be useful not only for election audits but elsewhere.

Joseph Kiniry, Portland, OR, and
Ronald L. Rivest, Cambridge, MA

Authors Respond:

We cannot fully elucidate here the flaws in each of the many proposed cryptographically verifiable voting subsystems. Their complexity and that of the surrounding systems environments undemocratically shifts the confirmation of correct implementation to a scant few intellectually elite citizens, if even accomplishable within an election cycle. However, all of these methods have vulnerabilities similar to the Volkswagen emission system; that is, stealth code can be triggered situationally, appearing correct externally while internally shifting vote tallies in favor of certain candidates over others. We have previously discussed the incompleteness of cryptographic solutions embedded in untrustworthy infrastructures, potentially enabling ballot contents to be manipulated or detected via vote-selling tags (such as write-in candidates or other triggers). The mathematics of close elections also requires that a very high percentage of ballots (over 95%) be independently checked against the digital record, which is not likely to occur, leaving the results unverified.

Rebecca T. Mercuri, Hamilton, NJ, and
Peter G. Neumann, Menlo Park, CA

Unintended Consequences of Trusting AIs

Toby Walsh’s Viewpoint “Turing’s Red Flag” (July 2016) raised very good points about the safety of increasingly human-like AI and proposed some common-sense law to anticipate potential risks. It is wise to discuss such protections before the technology itself is perfected. Too often the law trails the technology, as with the Digital Millennium Copyright Act in response—perhaps a decade late—to illegal file sharing.

Walsh primarily addressed the potential threat of autonomous systems being mistaken for humans, but what about the reverse? Humans could gain an unfair or even a dangerous advan-

tage by impersonating an AI. For instance, in a world where autonomous vehicles are allowed smaller following distances and prompt extra caution from nearby human drivers, a human could install an “I am autonomous” identity device in order to tailgate and weave through traffic with impunity, having won unearned trust from other drivers and vehicles.

A similar situation could arise with the advent of bots that act as intermediaries between humans and online services, including, say, banks. As bots become more trusted, a human-in-the-middle attack could compromise everyone’s private data.

At perhaps the outer reaches of techno-legal tension, we could even imagine the advent of identity theft where the individual is an AI, lovingly brought to life by a Google or an Amazon, and the thief to be punished is a human impersonator. Is this the route through which AIs might someday become legal persons? In a world where the U.S. Supreme Court has already extended constitutional free speech rights to corporations, this scenario seems quite plausible.

Mark Grossman, Palo Alto, CA

Author Responds:

Grossman makes a valid point. Just as we do not want bots to be intentionally or unintentionally mistaken for human—as I suggested in my Viewpoint—we also do not want the reverse. The autonomous-only lane on the highway should not have humans in it pretending to be, say, the equivalent of more-capable autonomous drivers.

Toby Walsh, Berlin, Germany

More to Asimov’s First Law

In his Viewpoint (July 2016), Toby Walsh argued for some sort of preliminary indication in cases in which a human is interacting with a robot. I suggest he check Isaac Asimov’s classic science fiction novels *Caves of Steel* (1953) and *The Naked Sun* (1957) for an earlier treatment of the topic. In the latter work especially,

R. (Robot) Daneel Olivaw deliberately hides his/its nature while investigating a murder. Asimov also included interesting discussion on the limitations inherent in the first of his “Three Laws of Robotics,” whereby “A robot may not injure a human being or, through inaction, allow a human being to come to harm,” as it assumes the robot is aware such action/inaction would itself be harmful.

Joe Saur, Yorktown, VA

Author Responds:

Yes, science fiction offers many stories that support the call for a Turing Red Flag law whereby autonomous systems are required to identify themselves. I mentioned the movie Blade Runner, which is, of course, based on Philip K. Dick's 1968 novel Do Androids Dream of Electric Sheep? Isaac Asimov's oeuvre also contains many examples. We should all listen to these warnings.

Toby Walsh, Berlin, Germany

To Design Educational Languages, First Know What You Want to Teach

We were pleased by the attention R. Benjamin Shapiro's and Matthew Ahrens's Viewpoint “Beyond Blocks: Syntax and Semantics” (May 2016) gave to the potential educational value of tools that translate block syntax to text. However, though they did not identify any published studies that have evaluated possible benefits from such tools, several recent studies have indeed been done. Moreover, smooth transitions from blocks to text syntax have been a feature of research enhancements to existing languages (such as Tiled Grace by Homer and Noble) and of novel languages in successful products (such as the educational coding game Code Kingdoms). Researchers typically publish evaluations of their systems; we ourselves have evaluated the educational outcomes of Code Kingdoms.

But what specific skills and concepts are computer science educators actually teaching with such systems? To find out, we must focus on evaluating those skills and concepts, rather than on task performance or productivity measures with little relevance to educational objectives. We developed our own DrawBridge system¹ to support not only understanding of syntax through transition from blocks to (JavaScript) text syntax but

also transitions from direct manipulation drawing to geometric notation and from code to live Web deployment. Attention to educational assessment of benefits can also help guide and evaluate the design of continuing work, as in Shapiro and Ahrens. Educators and system designers should thus recognize the importance of notational expertise—understanding the nature and function of concrete syntax—along with the more popular but abstract concerns of computational thinking. An important step toward improving the design of educational systems is to better understand what computer science educators are actually trying to teach.

Reference

1. Stead, A.G. *Using multiple representations to develop notational expertise in programming*. Technical Report, Computer Laboratory, University of Cambridge, Cambridge, U.K., June 2016; <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-890.pdf>

Alistair Stead and **Alan Blackwell**,
Cambridge, U.K.

Still Want to Know Who Is the Human

Commenting on Moshe Y. Vardi's Editor's Letter “Would Turing Have Passed the Turing Test?” (Sept. 2014), Huma Shah's and Kevin Warwick's letter to the editor “Human or Machine?” (Apr. 2015) included part of a conversation between a judge (J19) and candidates (E20 and E24) of the (now famous) Turing Test experiment. Readers were asked to decide whether E20 or E24 is the computer—an appropriate and indeed challenging question. Unfortunately, I could not find a resolution in *Communications* or elsewhere. Would it be possible to get the correct answer from Shah and Warwick? I would like to include it in a quiz in a theory-of-computing course.

Sven Kosub, Konstanz, Germany

Author Responds:

In the 2014 experiment, all judges were informed there was indeed one human and one machine in each simultaneous comparison test. For Judge J19, the result for parallel interrogation of hidden entities E24 and E20 was the correct identification of the left interlocutor, E20, as a machine, awarding it 52/100 for conversational ability; E20 was UltraHal from Zabaware. J19 was unable to determine the nature of the right interlocutor,

E24, which was actually a human male, reporting “unsure” for E24.

Huma Shah, London, U.K., and
Kevin Warwick, Reading, U.K.

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

© 2016 ACM 0001-0782/16/09 \$15.00

Coming Next Month in **COMMUNICATIONS**

A Brief Chronology of Medical Device Security

Rethinking Security for Internet Routing

Universities and Computer Science in the European Crisis of Refugees

Idle-Time Garbage-Collection Scheduling

Dynamics of Change: Why Reactivity Matters

Addressing Ethical Considerations in Network Measurement Papers

Incremental, Iterative Data Processing with Timely Dataflow

Efficient Parallelization Using Rank Convergence in Dynamic Programming Algorithms

Plus the latest news about the end of Keck's Law, holographic interactions, and algorithm fairness.

Come join us at CSCW 2017!



CSCW
2017
PORTLAND, OR

DoubleTree by Hilton Hotel
February 25 - March 1, 2017

For more information
about the conference and
to register, please visit:
cscw.acm.org/2017/

Facebook: www.facebook.com/acmCSCW/
Twitter: @ACM_CSCW
Sina Weibo: ACM_CSCW (in Chinese)

The ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW) is the premier venue for presenting research in the design and use of technologies that affect groups, organizations, communities, and networks. Bringing together top researchers and practitioners from academia and industry, CSCW explores the technical, social, material, and theoretical challenges of designing technology to support collaborative work and life activities.

The scope of CSCW spans socio-technical domains including work, home, education, healthcare, the arts, leisure, and entertainment. The conference seeks novel research results or new ways of thinking about, studying, or supporting shared activities in these and related areas.

Upcoming Deadlines

Workshop Proposals: *Oct. 17, 2016*

Interactive Posters: *Nov. 4, 2016*

Panels: *Nov. 4, 2016*

Doctoral Colloquium: *Nov. 4, 2016*

Demonstrations: *Nov. 4, 2016*

Reconciling Quantum Physics with Math

Mathematicians explore the root of many problems in developing a proof for the Kadison-Singer problem.

A SOLUTION to a problem in mathematics that lingered unsolved for more than 50 years could help deliver faster computer algorithms to many problems in physics and signal processing. However, it may take years for mathematicians to fully digest the result, which was first published online three years ago.

The roots of the problem defined by Richard Kadison and Isadore Singer in the late 1950s lie in attempts to give the physics of quantum mechanics a footing in abstract mathematics. The concept it deals with traces back to Werner Heisenberg's initial work on quantum mechanics. Heisenberg used matrix mathematics to develop his model of the quantum world that says it is not possible to accurately measure simultaneously different properties of a physical system at the microscopic level.

A decade later, John von Neumann applied graph theory to the problem of reconciling what physicists had postulated about quantum mechanics with mathematics. One result of this work was the development of a specialized algebra known as C^* (pronounced



From left, Nikhil Srivastava, Adam Marcus, and Daniel Spielman shortly after completing the proof of the Kadison-Singer problem.

“see-star”) that could model quantum states. In this algebra, Kadison and Singer formulated a question of abstract mathematics that mirrored one of the key issues facing quantum mechanics: do measurements of quantum properties in the observable world map into uniquely identifiable states

that include the properties that cannot be measured?

They proved this did not apply to one class of measurements: those referring to continuous properties such as momentum. They thought, but could not prove, the answer would also be ‘no’ for discrete situations, such as energy states.

For close to 40 years, attempts to deliver a proof of the Kadison-Singer problem within the discipline of C^* -algebra failed. Yet work in seemingly unrelated fields of mathematics turned up other angles of attack on the problem.

Petter Brändén, a mathematician working at the KTH Royal Institute of Technology in Sweden, says, “Kadison-Singer is a very old problem. If it was going to be solved, it was going to be better to attack from another angle. You see it quite a lot in mathematics: in order to solve these older problems, you need to use new tools or have a brilliant idea.”

The path to a solution to Kadison-Singer lay in mathematicians seeing connections between different branches of the discipline. Adam Marcus, an assistant professor at Princeton University, worked with Daniel Spielman and Nikhil Srivastava at Yale University to solve the Kadison-Singer problem. He adds, “Looking back, essentially the problem was hard to solve because it demands two different approaches.

“One of the parts could be dealt with using techniques in functional algebra, but the other problem was a combinatorial one. Unless you have someone who knew the techniques from both areas or a way of coming together, it probably wasn’t going to get solved.”

The first clear step on the path to solving Kadison-Singer was taken close to 40 years ago, with a conjecture developed by Joel Anderson while working at Pennsylvania State University for a way of partitioning a specific type of matrix into multiple sub-matrices that fit within certain limits. This “paving conjecture,” Anderson argued, was equivalent to the Kadison-Singer problem. Its advantage, according to Pete Casazza, Curator’s Distinguished Professor at the University of Missouri, is that it took the original problem away from being based on C^* -algebras and into the much more heavily researched field of operator theory.

Just over a decade ago, Casazza began to notice an increasing number of problems resembling that of Kadison and Singer. Having moved from pure mathematics where he dealt with Banach spaces (which are closely related to C^* -algebras), Casazza started working on signal-processing problems, such

The path to a solution to Kadison-Singer lay in mathematicians seeing connections between different branches of the discipline.

as separating audio signals from each other. “One of the leaders is Hans Feichtinger. He wrote to me and asked me about the vectors we could use to do signal analysis. I recognized that this problem was at least similar to the Kadison-Singer problem. It took a few years before I realized they were the same problem, and then I began to find other problems that were similar to Feichtinger’s,” says Casazza.

With colleagues, Casazza published a paper in 2006 that described a number of problems collected from different areas of mathematics, spanning audio processing, building error tolerance into Internet protocols, and other branches of mathematics.

“I wasn’t looking for equivalence with Kadison-Singer: I was looking for an area that might provide a powerful solution to our problem,” Casazza. “But at a meeting, someone asked me, ‘when did you start to make a living out of finding conjectures similar to Kadison-Singer?’”

Two years later, Gil Kalai, the Henry and Manya Noskwith professor of mathematics at the Hebrew University of Jerusalem, while visiting Yale University in his role as an adjunct professor, spoke to Spielman about his group’s work on manipulating the matrices that describe the connections between points on a graph.

The primary focus in the Yale group is developing “sparsifiers,” techniques to reduce the number of connections in graphs that are often used to model networks, electrical circuits, and mechanical systems.

By sparsifying the graphs, problems can be solved faster without losing

overall accuracy. Matrix math provides a way to represent the information in graphs, which provides a connection to many problems in physics that rely on the ability to solve simultaneous equations that also can be represented in matrix form.

“Our main motivation for this was developing very fast systems for solving systems of linear equations and Laplacian matrices,” Spielman says. “We are trying to come up with faster algorithms, and sparse is a big hammer for that.”

Kalai told Spielman the work looked similar to Anderson’s paving conjectures, and with some further development, could be used to solve the Kadison-Singer problem.”

“We looked it up and thought, ‘it won’t take very long.’ We were very wrong,” Spielman noted. “To prove Kadison-Singer, we had to go through a very different kind of route to the one we had been taking.”

Spielman points out the relatively short proof hides many blind alleys that he, Marcus, and Srivastava investigated on their way to the final proof. “There were similarities, but our first path didn’t work. We spent a while trying to modify our original proof, but eventually failed.

“For most of the five years we were working on it, the types of proof we had looked quite different to the original work we did.”

Says Marcus, “A lot of techniques that we picked were based on experimentation. We acted like scientists: based on the information, we decided how to proceed.”

Spielman adds, “It makes a large difference to be able to experiment. In general, I do a lot of computational experiments to figure out what should be true and find counterexamples to conjectures.”

The core of the final proof revolves around the properties of specialized types of polynomials. These interlacing polynomials call for the roots of two otherwise independent polynomial functions to interleave, such that each root of one lies between two roots of the other. The polynomials also need to fall within a class known as “real stable.” Matrix functions such as determinants provide suitable polynomials. Spielman says the computational work

gave the group the confidence to pursue this route to the eventual Kadison-Singer proof.

“The theory around stable polynomials and hyperbolic polynomials is not new, but it has been more heavily developed over the past 10 years. Stable polynomials had been used to prove other conjectures and had already been proven to be useful for a wide set of problems,” says Brändén who, with the late Julius Borcea, worked on the properties of these polynomials. “The approach gives you ways of deforming polynomials, so you can start with a determinant and deform it so that it’s not a determinant any more, but you still have control over the roots.”

Although it marks a breakthrough in pure mathematics, the Marcus-Spielman-Srivastava proof is not a constructive proof, and so does not provide a guide to developing algorithms that can be used in signal processing, physics, or other areas. The proof’s novel approach also presents problems.

Says Casazza, “People in my area are not experts in the tools these guys use, so they not trying to use those tools to go further. We would really like a constructive proof of Kadison-Singer or its equivalents.”

What the proof does deliver is increased confidence that such a constructive proof can be discovered. Also, it might come from the same angle of attack.

“The algorithmic problem is simpler now,” says Marcus, pointing out the availability of even a non-constructive proof means there is a target at which to aim.

Many of the problems being worked on look for a minimum difference between the original graph and a version that has been split into two or more partitions when approximations have been employed. “You used to be asking what the minimum value could be,” Marcus says. “Now you know. You just need an algorithm that finds the minimum. You don’t have to worry whether it’s a minimum or not. Having the knowledge that these things exist opens up the possibility for different algorithms. Previously you could find a solution, but couldn’t say that the solution is good.”

Spielman and colleagues have ex-

plored the theme of interlacing polynomials in a series of four papers, with a fifth in preparation. The work has begun to point to algorithms for other problems, although the original sparsification path still provides the most hope for faster matrix solvers.

“At first, I didn’t think an algorithmic proof would be available from this work, but now there is work that gives hope to that possibility,” Marcus says.

Brändén adds, “I believe we will see this method applied to more problems. Quite recently it was proven applicable to the traveling-salesman problem, and I think that the theory can be developed further.”

One lingering issue may lie in the specific type of polynomials on which the Marcus-Spielman-Srivastava proof of Kadison-Singer lies.

“The types of problems for which these polynomials can be used are very specialized, but we hope they will become more generic,” says Marcus. “It may be that these polynomials really are very specific. I tend to think that’s not true, but if we can’t make it more generic, the proof probably won’t be used as much as it could.” **C**

Further Reading

Marcus, A., Spielman, D. And Srivastava, N., Interlacing polynomials II: Mixed Characteristic Polynomials and the Kadison–Singer Problem, *Annals of Mathematics*, 182 (2015), Issue 1. Preprint: <http://arxiv.org/abs/1306.3969>

Casazza, P. and Tremain J., Consequences of the Marcus/Spielman/Srivastava Solution of the Kadison-Singer Problem, *New Trends in Applied Harmonic Analysis: Sparse Representations, Compressed Sensing, and Multifractal Analysis*, ed. Akram Aldroubi, Carlos Cabrelli, Stephane Jaffard, Ursula Molter (2016). Preprint: <http://arxiv.org/pdf/1407.4768v2.pdf>

Borcea, J. and Brändén, P., The Lee-Yang and Pólya-Schur Programs, II: Theory of Stable Polynomials and Applications, *Comm. Pure Appl. Math.*, 62:1595–1631 (2009).

Kadison, R.V. and Singer, I. M., Extensions of Pure States, *American Journal of Mathematics*, Vol. 81, No. 2 (April 1959), 383–400

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

ACM Member News

INITIAL INTRIGUE YIELDS CAREER IN COMPUTER SCIENCE



When he was in high school, recalls Kevin Fall, deputy director and CTO of the

Software Engineering Institute (SEI) at Carnegie Mellon University (CMU), a friend of his had a home computer, “which in those days was still rare. I was intrigued and wanted to know how things worked, so I started looking at how floppy disk drivers and assembly language operated.”

As a result, when he enrolled at the University of California, Berkeley, Fall had an equal interest in medicine and computers. “In those days, they didn’t overlap at all, so I had to choose one. I wound up choosing computers.” After earning his B.A. in computer science at UC Berkeley, Fall earned M.S. and Ph.D. degrees in computer science from the University of California, San Diego.

When the opportunity arose to co-found startup NetBoost Corp., which developed and marketed hardware and software solutions for networking and communications, Fall “spent half of my time at the start-up and half at UC Berkeley as an adjunct professor.” NetBoost was later acquired by Intel, where Fall spent 12 years. “After that, I was at Qualcomm working on video streaming. Then, out of the blue, I received a recruiting invitation for the CTO/Deputy Director position at the SEI at CMU.”

Fall is responsible for technical direction across SEI. “The particular thing I am responsible for is the research program, running the internal competition we use for the allocation of research funds, but then also representing what we do to our sponsor, which is the (U.S.) Department of Defense, and to the rest of the Carnegie Mellon campus.”

—John Delaney

GPUs Reshape Computing

Graphical processing units have emerged as a major powerhouse in the computing world, unleashing huge advancements in deep learning and AI.

AS RESEARCHERS CONTINUE to push the boundaries of neural networks and deep learning—particularly in speech recognition and natural language processing, image and pattern recognition, text and data analytics, and other complex areas—they are constantly on the lookout for new and better ways to extend and expand computing capabilities. For decades, the gold standard has been high-performance computing (HPC) clusters, which toss huge amounts of processing power at problems—albeit at a prohibitively high cost. This approach has helped fuel advances across a wide swath of fields, including weather forecasting, financial services, and energy exploration.

However, in 2012, a new method emerged. Although researchers at the University of Illinois had previously studied the possibility of using graphics processing units (GPUs) in desktop supercomputers to speed processing of tasks such as image reconstruction, a group of computer scientists and engineers at the University of Toronto demonstrated a way to significantly advance computer vision using deep neural nets running on GPUs. By plugging in GPUs, previously used primarily for graphics, it was suddenly possible to achieve huge performance gains on computing neural networks, and these gains were reflected in superior results in computer vision.

The advance proved revolutionary.

“In only a few short years, GPUs have emerged at the center of deep learning,” says Kurt Keutzer, a professor in the Electrical Engineering & Computer Science Department at the University of California, Berkeley. “The use of GPUs is now moving into the



NVIDIA's Titan X graphics card, featuring the company's Pascal-powered graphics processing unit driven by 3,584 CUDA cores running at 1.5GHz.

mainstream, and by applying dozens to hundreds of processors to a single application, they are on a trajectory to radically change computing.”

Adds Wen-Mei W. Hwu, Walter J. Sanders III—Advanced Micro Devices Endowed Chair in Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, “GPUs are remarkable throughput computing devices. If you only have one task, it doesn’t execute particularly fast on GPUs. However, if you have a large number of independent tasks, it works really well.”

A Deeper Vision

GPU architectures have their roots in basic graphical rendering operations like shading. In 1999, Nvidia introduced the GeForce 256, which was dubbed the world’s first GPU. Simply put, the specialized circuits—which may be built into a video card or on a motherboard—manipulate and optimize computer memory in order to accelerate rendering on displays. Today GPUs are used in a wide array of devices, including personal computers, tablets, mobile phones, workstations,

IMAGE COURTESY OF GEFORCE.COM/NVIDIA

electronic signage, gaming consoles, and embedded systems.

However, “Many emerging applications in computer vision and deep learning are memory bandwidth-limited,” Keutzer explains. “In these applications, the speed of the application is often ultimately dictated by the time it takes to draw data from memory and stream it through the processor.”

A big advantage of a GPU implementation, and something that is frequently overlooked, is its superior processor-to-memory bandwidth. As a result, “In bandwidth-limited applications, the relative processor-to-memory bandwidth advantage transfers directly to superior application performance,” Keutzer points out. The key is that GPUs provide greater floating-point operations per second (FLOPs) using fewer watts of electricity, and they actually extend this energy advantage by supporting 16-bit floating point numbers, which are more power- and energy-efficient than single-precision (32-bit) or double-precision (64-bit) floating point numbers.

The manycore approach to GPUs relies on a larger number, such as 32 to 64, of simpler processor cores implemented in larger numbers. By contrast, multicore approaches use smaller numbers of conventional microprocessors, typically 2 to 4 to 8. The upshot? “GPUs deliver superior performance and better architectural support for deep neural networks. The performance advantages of GPUs on deep neural nets are transferred onto an increasingly broad variety of applications,” Keutzer says.

Today, a typical cluster is comprised of 8 to 16 GPUs, though researchers such as Keutzer are now pushing the numbers into the hundreds to simultaneously train deep neural nets on extraordinarily large datasets that would otherwise require weeks of training time. The training consists of running massive amounts of data through the system in order to get it to a state where it can solve problems. At that point, it may run on a CPU or hybrid processor. “This is not an academic exercise,” Keutzer notes. “We need this kind of speed in training neural nets to support emerging applications such as self-driving cars.”

“The use of GPUs is now moving into the mainstream, and by applying (many) processors to a single application, they are on a trajectory to radically change computing.”

GPU technology is advancing far faster than that of conventional CPUs. The scalability of GPUs, along with their sheer floating point horsepower and lower energy consumption, is turbocharging deep learning and machine learning tasks, says Bryan Catanzaro, senior researcher at Chinese-based Internet services, search, and data firm Baidu. “Deep learning is not new. GPUs are not new. But the field is taking off because of huge advances in computational capabilities and the availability of richer datasets.”

Much of the thrust has come from Nvidia, which has introduced increasingly sophisticated GPUs, including the new Pascal architecture that is designed to tackle specific tasks, such as training and inference. Its latest GPU system, the Tesla P100 chip, packs 15 billion transistors on a piece of silicon, twice as many as previous processors.

Baidu, for example, is pushing into new frontiers in speech recognition. Its “Deep Speech” initiative, which relies on an end-to-end neural net, provides speech recognition accuracy that rivals humans on short audio clips in both English and Chinese. The company is also venturing into the autonomous vehicle space with GPU technology; it has developed a self-driving vehicle that has navigated the streets of Beijing, with maneuvers including changing lanes, passing other vehicles, and stopping and starting.

Meanwhile, researchers at Microsoft Asia have used GPUs and a variant of Deep Neural Nets, called Residual Neu-

ral Nets, to achieve superior accuracy on the computer vision problems of object classification and object recognition.

Google, too, is using these techniques to continually improve its image recognition algorithms. Says Ilya Sutskever, a former Google AI researcher who is now research director at Open AI (a non-profit artificial intelligence research company, <https://openai.com>), “Neural networks are enjoying a renaissance. The core ideas of neural networks and deep learning have been discussed and pondered for many years, but it was the development of the general-purpose GPU that was a key enabler of their success.”

One Step Beyond

While GPU technology is pushing into new frontiers in the deep learning space, plenty of computing challenges remain. For one thing, “Programming individual manycore devices such as GPUs for high efficiency is still very difficult, and that difficulty is only compounded when these devices are gathered together in multi-GPU clusters,” Keutzer says. Unfortunately, he adds, “Much of the expertise for effectively programming these devices is housed in companies and many details of the techniques that have been developed are not widely shared.”

Similarly, the design of Deep Neural Nets is widely described as a “black art,” Keutzer says; creating a new Deep Neural Network architecture is as complicated, in many ways, as creating a new microprocessor architecture. To make matters worse, once the Deep Neural Network architecture is created, “there are many knobs, known as hyperparameters, used during training, and accuracy is only achieved when these knobs are set appropriately. All of this adds up to a knowledge gap between those ‘in the know’ and others.

“Individuals with expertise in either Deep Neural Nets or GPU programming are scarce, and those who know both well are very rare.”

Another challenge is understanding how to use GPUs most effectively. For example, Baidu requires 8 to 16 GPUs to train one model, achieving 40% to 50% of peak floating point math throughput on the entire application.

“This means there is very little performance left on the table,” Catanzaro says. “There are things that we would like to do to scale to more GPUs, so rather than using 8 or 16 GPUs, we would like to use 128 GPUs, for example.” This translates into a need for better interconnects, as well as the ability to move from 32-bit floating point support to the throughput of 16-bit floating point support. Nvidia’s next generation GPU, code-named Pascal, may address some of these issues.

Still another obstacle is better integrating GPUs with CPU/GPUs. Hwu says those two types of processors are not often integrated together, and they usually do not have high bandwidth communication between the two. This translates into a limited number of applications and capabilities that run well on these systems. “You really need to be able to give the GPU a kind of a very big task and with some amount of data and then let the GPU crank on it for a while to make this offloading process worthwhile,” Catanzaro explains.

Current Nvidia GPUs are located on separate chips. They are usually connected to the CPU via an I/O bus (PCIe). This is the reason one needs to send large tasks to the GPU. Future systems will integrate GPUs and CPUs in one tightly coupled package that supports higher bandwidth, lower latency, and cache coherent memory sharing across CPUs and GPUs.

Keutzer expects that over time, as CPUs and GPUs become better integrated, better cache coherence and synchronization between the two types of processors will result. In fact, Nvidia and Intel are both focusing on this space. Keutzer notes a new Intel chip dubbed Knight’s Landing (KNL) offers unprecedented computing power in a Xeon Phi 72-core supercomputing processor that integrates both CPU and GPU characteristics. This chip also offers 500 gigabyte-per-second processor-to-memory bandwidth that will erode GPU’s advantage in this area, he says.

Hwu notes each of the KNL chip’s 72 cores can execute “a wide vector instruction (512 bytes). When translated into double precision (8 bytes) and single precision (4 bytes), the vector width

Keutzer expects as CPUs and GPUs become better integrated, better cache coherence and synchronization between the two types of processors will result.

is 64 and 128 words; in that sense, it has a similar execution model to that of GPUs.”

The programming model for the KNL chip is the traditional x86 model, Hwu says, so programmers “need to write code to either be vectorizable by the Intel C Compiler, or use the Intel AVX vector intrinsic library functions.” The programming model for GPUs is based on the kernel programming model, he adds.

Also, X86 cores have cache coherence for all levels of the cache hierarchy, Hwu says, “whereas GPU’s first-level caches are not coherent. It does come with a cost of reduced memory bandwidth.” However, he says, “For deep learning applications, cache coherence for the first level cache is not very important for most algorithms.”

Over the next decade, a big wildcard in all of this will be how development cycles play out, Hwu says. He believes Moore’s Law can continue at its present rate for about three more generations. At the same time, he says, it will likely take about three generations for system designers and engineers to move away from mostly discrete CPU and GPU systems to true hybrid designs.

“If Moore’s Law stalls out, it could dramatically impact the future of these systems, and the way people use hardware and software for deep learning and other tasks,” Hwu points out. “Yet, even if we solve the hardware problem, certain deep learning tasks require huge amounts of labeled data. At some

point, we will need a breakthrough in generating labeled data in order to do the necessary training, particularly in areas such as self-driving cars.”

Over the next few years, Sutskever says, machine learning will tap GPUs extensively. “As machine learning methods improve, they will extend beyond today’s uses and ripple into everything from healthcare and robotics to financial services and user interfaces. These improvements depend on faster GPUs, which greatly empower machine learning research.”

Adds Catanzaro: “GPUs are a gateway to the future of computing. Deep learning is exciting because it scales as you add more data. At this point, we have a pretty much insatiable desire for more data and the computing resources to solve complex problems. GPU technology is an important part of pushing the limits of computing.”

Further Reading

Raina, R., Madhavan, A., and Ng, A.Y. *Large-scale Deep Unsupervised Learning using Graphics Processors, Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009. <http://www.machinelearning.org/archive/icml2009/papers/218.pdf>

Wu, G., Greathouse, J.L., Lyashevsky, A., Jayasena, N., and Chiou, D. *GPGPU Performance and Power Estimation Using Machine Learning*. Electrical and Computer Engineering, The University of Texas at Austin, 21st IEEE International Symposium on High Performance Architecture, 2015. <http://hgpu.org/?p=13726>

Coates, A., Huval, B., Wang, T., Wu, D.J., Ng, A.Y., and Catanzaro, B. *Deep learning with COTS HPC systems. Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. http://cs.stanford.edu/~acoates/papers/CoatesHuvalWangWuNgCatanzaro_icml2013.pdf

Chen, X., Chang, L., Rodrigues, C.I., Lv, J., Wang, Z., and Hwu, W. *Adaptive Cache Management for Energy-Efficient GPU Computing, MICRO-47 Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 343-355, IEEE Computer Society, 2014. <http://dl.acm.org/citation.cfm?id=2742190>

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

The Edge of the Uncanny

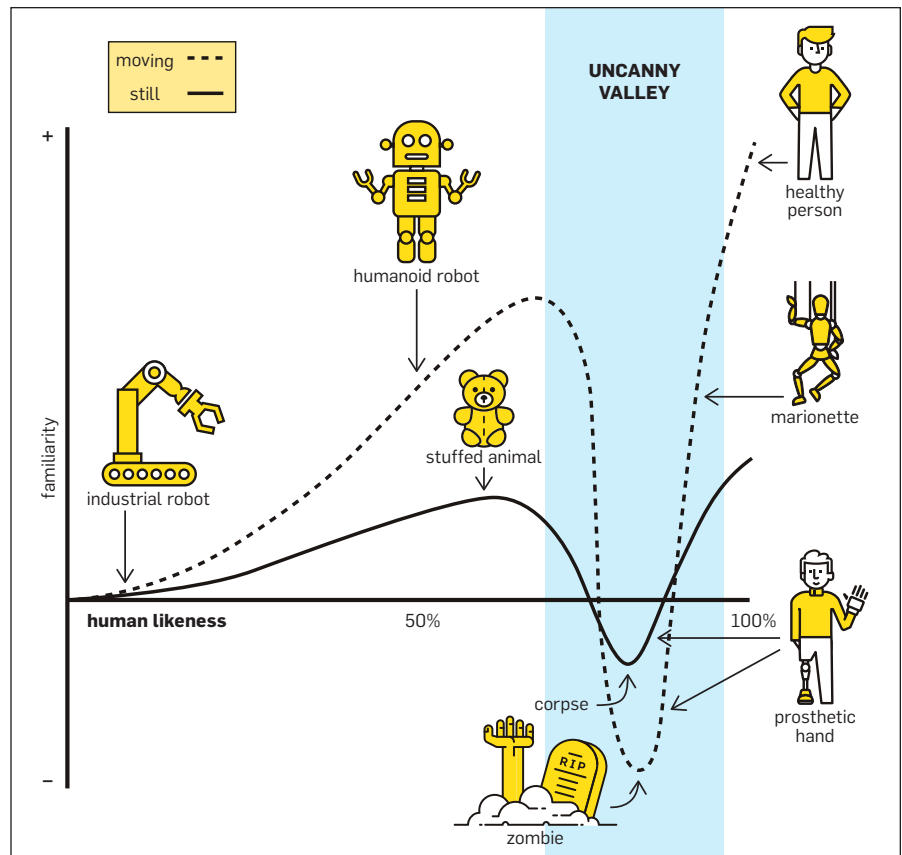
Scientists are learning more about what makes robots and chatbots engaging.

IN CONVERSATION, MITSUKU admits she does not know if her name has any meaning; this is simply what her father called her. Actually, she does not really have a father. She has a Mousebreaker, which is technically not a person, either, but a team of programmers who like beer and curry and share a fear of Daleks (the evil alien robots from *Doctor Who*).

Mitsuku is quick-witted, occasionally confusing, and strangely engaging. She is also a chatbot, built from the A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) platform originally developed by Richard Wallace in 1995. She conducts hundreds of thousands of conversations daily, according to Lauren Kunze, principal of Pandorabots, the Oakland, CA-based company behind the technology. “She doesn’t really do anything,” Kunze says. “She’s not designed to assist you. She can tell you the weather or perform an Internet search, but she’s really just there to talk to you, and she’s wildly popular with teens. People say, ‘I love you’ and ‘you’re my best friend.’”

The appeal is not accidental. The designers of chatbots like Mitsuku and the engineers of physical social robots have made significant advances in their understanding of how to build more engaging machines. Yet there are still many challenges, one of which is the unpredictability of humans. “We just don’t understand how people are going to react to physical or software robots,” says University of Southern California computer scientist Yolanda Gil, chair of SIGAI, ACM’s Special Interest Group on Artificial Intelligence. “This is one kind of technology where people continue to surprise us.”

While there are no absolute guide-



Masahiro Mori's "Uncanny Valley," based on the notion that machines that appear too human can have a repulsive effect because of their "uncanny" resemblance to real humans.

lines for building effective social robots or engaging chatbots, a few common themes have emerged.

Uncanny Expectations

One frequently cited theory in social robotics is the Uncanny Valley, first described by Japanese roboticist Masahiro Mori in 1970. The Uncanny Valley contends there is a risk in building machines that are too human, that instead of attracting people, realistic androids can have a repulsive effect because of their “uncanny” resemblance to real humans. The reasons for the aversion are varied. Researchers have found evidence

that highly capable androids bother people because they represent a threat to human uniqueness, or that on a subconscious level, they actually remind us of corpses.

Ultra-realistic humanoids also generate higher social expectations. “When you have a human-like appearance, people expect a matching level of behavioral realism,” says roboticist Karl F. MacDorman of Indiana University. In 2005, for example, MacDorman was studying in the lab of Japanese roboticist Hiroshi Ishiguro when the group tested a socially interactive android at a conference. At first, background noise impeded the machine’s

speech recognition software, causing a delay in the robot's responses until the scientists added more processors. That initial inhuman delay did not match the lifelike appearance, and the effect was unsettling to early attendees.

While robots that appear too human may not be ideal for social interaction, there is also a downside to being too much of a machine. Ilya Gelfenbeyn, founder of Api.ai, a platform that allows companies to build customized chatbots, says his software can process text and generate replies in just 50 milliseconds. Previously, Gelfenbeyn developed a Siri-like conversational app that was slower and occasionally more difficult to converse with because it needed more time to process speech. The speed and efficiency of his chatbots seemed to be a clear step forward.

To Gelfenbeyn's customers, though, the pace was problematic. When humans message each other, we pause between replies; his chatbots, on the other hand, were responding instantaneously. "You're not used to getting answers immediately," he says, "so one of the requests

Robots that appear too human are not ideal for social interaction, but there also is a downside to being too much of a machine.

we've gotten is to add delays; otherwise it feels unnatural."

Designed for a Purpose

The particular choices that are made when designing an effective social robot also depend on its function, according to Cory Kidd, CEO of Catalia Health. Kidd spearheaded the creation of Mabu, Catalia's socially interactive robot, which will act as a kind of in-home healthcare coach, encour-

aging patients to follow their physician-prescribed medication plans. The first Mabu units will go into patient homes this year, and the robot's goal is to keep these individuals out of the hospital.

Mabu does not move or manipulate objects; the small yellow robot consists of a torso, a head with facial features that vaguely resemble Casper the Friendly Ghost, and an interactive tablet. When Kidd and his colleagues were designing Mabu, they knew they needed something that could engage people, build trust, and move into a home in a low-impact way. The eyes were among their first considerations. They tested variations with large eyes reminiscent of anime characters, and others that resembled the digital cast of a Pixar movie. In the end they settled on something in between: attractive, but not so cartoonish that Mabu looks like a doll. The eyes have to be right, Kidd explains, because establishing eye contact is critical. "Psychologically, that's really key to helping draw someone's attention and draw them into a conversation," he says.

Ethics

Sociotechnical Changes Inspire Updating ACM's Code of Ethics and Professional Conduct

In 1992, everyone knew someone who owned a rotary telephone; today, many young people have never "dialed" a phone. In 1992, artificial intelligence was confined to research labs; today, we find it driving on our highways. In 1992, no one had ever caught a Pokemon; today, they are everywhere.

In 1992, the ACM Code of Ethics and Professional Conduct (www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct) was adopted as the 15th bylaw of the ACM.

It has not always been easy to see the impact of the Code on our profession since then, but recent reports indicate this is no longer the case. The Code was cited in the *Oracle v. Google* case

about the Java API, and by *The New York Times* with reference to whether Apple ought to unlock an iPhone for the FBI. In addition, ACM's Committee on Professional Ethics (COPE) has received an inquiry from a major corporation about how the Code applies to research, development, and deployment of artificial intelligence and machine learning.

Computing has changed over the last 24 years and now influences every corner of our lives. The significance of clear ethical vision for computing professionals has become more important. As responsible computing professionals, we are updating the Code to maintain its relevance as the conscience of

computing so it can continue to be robust and a useful guide in decision-making.

There have been changes in the power, ubiquity, and mobility of computers. They are increasingly integrated into people's lives. There is a clear need to update the imperatives and supporting guidelines of the Code in order to better reflect the complexity of the environment we impact through the practice of our profession. In order to do a thorough job drafting updates to the Code, COPE needs your perspective on computing.

Our goal is to have a first draft of the updated Code available for comment in November. We invite you to participate in this process now

by submitting your perspective on changes in technology, society, and how people use technology, so those changes will be reflected in the updated Code. Better yet, take a look at the Code and offer a suggestion on how that might be done. The COPE website (ethics.acm.org/code-of-ethics/code-2018) has more information about this project, and a form for providing feedback.

The ACM Code of Ethics has been an influential document for decades. With your help, we will keep it relevant and effective.

—Marty J. Wolf is a professor of computer science at Bemidji State University, and is vice-chair of ACM's Committee on Professional Ethics (COPE).

Size was another consideration. Too large and the robot would be imposing and potentially threatening, but if it were too small, then people might not be willing to interact with it. Kidd says they also did not want to veer too far from the humanoid form and develop a pet-like machine. “If you build something that looks like a dog or a cat, there’s a natural inclination to interact,” he says. “But if you’re doing what we’re doing, building a healthcare coach, well, I’m not going to trust my dog for health advice.”

Strength of Personality

Mabu and other social robots demonstrate an engaging machine does not necessarily need to be humanoid in its appearance or behavior, but an engaging personality is critical. Kidd and his colleagues hired a former Hollywood screenwriter to construct Mabu’s backstory. They do not share the story, so this was not some kind of cute public relations play; instead, the purpose was to help them define the robot’s responses and reactions, and thereby deepen the interaction with the patient. “If you’re going to have this in a patient’s home for years, it needs to be a consistent character,” Kidd says. “It’s part of building up trust with a person. You’re giving them something that’s believable and credible.”

The same holds true for chatbots, according to Tim Delhaes of Inbound Labs, a marketing agency that creates bots for its customers using the api.ai platform. Often, when a company takes on a new client, they have to enter basic information into multiple project management, sales, and marketing platforms. The Inbound Labs chatbot presents one timesaving interface for all that brainless data-entry work. Through a series of questions and answers, the chatbot gathers the information it needs, then updates the different platforms independently. Yet Delhaes says the interaction still has to be enjoyable. “The more you make the bot participate in a natural way with humans,” says Delhaes, “the more likely people are going to use it or enjoy using it.”

The api.ai platform allowed him to design his own chatbot, and Delhaes based his character on Marvin, the mo-

An engaging machine does not necessarily need to be humanoid in its appearance or behavior, but an engaging personality is critical.

rose robot of *The Hitchhiker’s Guide to the Galaxy*. Then he wrote responses appropriate to that character. So, when a user makes a request, Marvin might answer, “Oh, no, what do you want now?” or simply “Do it yourself.” The robot does perform the task eventually, but Delhaes believes its human-like reluctance is part of its appeal.

At the same time, he notes he is not trying to trick people into believing they are interacting with a human. “It’s obviously fake,” he says. “People know it’s a robot.”

That distinction and clarity is important, according to Kunze of Pandorabots. She argues chatbots should be upfront about their machine status, as this helps build trust. When Mitsuiku churns out a confusing response, the bot reminds the subject that “she” is a piece of software, and suggests a few ways to help the software learn and improve its responses.

“All our data indicates that engagement is much, much higher with chatbots that are human-like,” says Kunze. “If you’re talking with something every day to get things done, you should enjoy who or what you’re talking with.”

The Difficulty of Universal Appeal

Still, the behavior of the human in the equation is difficult to predict. Even the most carefully designed robots and chatbots will not be appealing to everybody, according to Maartje de Graaf, a social scientist at the University of Twente in the Netherlands. She

recently released the results of a study measuring how people interacted with a rabbit-like, home-based social robot called Karotz. The majority of the 102 study participants talked to the robot, and some gave it a name, but others reported an uncomfortable, uncanny feeling when the robot initiated the conversation, and chose to reduce their social interactions.

Kidd says the wide range of patient personalities Mabu will encounter has been a major consideration from the start. The first conversation the robot has with a human will be extremely important in terms of initiating a bond, setting expectations, establishing trust, and more. “We’ve created a lot of initial conversations,” he says, “but for us it depends a lot on the personality of the patient, and how we adapt.” Mabu will analyze the person’s tone, the content of their response, and even their facial expressions to gauge his or her reaction and generate the appropriate response.

In some ways, the unpredictability of human responses to these machines should be expected, since they are so fundamentally new and unfamiliar, according to social psychologist Maria Paola Paladino of the University of Trento, in Italy. “They are not human, but they’re not exactly machines,” says Paladino. “They are a different entity.”

Further Reading

Asada, M.

Towards Artificial Empathy. International Journal of Social Robotics, 6, 1, February 2015.

MacDorman, K. F., and Entezari, S.

Individual differences predict sensitivity to the uncanny valley. Interaction Studies, 16, 2, 141–172. 2015.

Kidd, C.D. and Breazeal, C.

Robots at Home: Understanding Long-Term Human-Robot Interaction. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Kanda, T. and Ishiguro, H.

Human-Robot Interaction in Social Robotics. CRC Press, 2012.

Mabu: A Personal Healthcare Companion
<https://vimeo.com/138783051>

Gregory Mone is a novelist and science writer in Boston, MA.

© 2016 ACM 0001-0782/16/09 \$15.00

Law and Technology

No Easy Answers in the Fight Over iPhone Decryption

A look at the legal background and future possibilities for an issue that is likely to reoccur.

PERHAPS THE MOST significant law and technology story of 2016 thus far has been the fight between the FBI and Apple over whether Apple could be compelled to assist in decrypting the contents of an alleged terrorist's iPhone. The San Bernardino case that captured the most attention ended without a judicial decision, because the FBI withdrew its request after a third party provided it with a means to obtain the data without Apple's help. The underlying legal questions though had already been raised in other cases and will surely recur in future ones. In this column, I explain the legal issues in the case and why it should be neither impossible nor easy for the government to compel this sort of assistance.

This story began when the FBI came into possession of the iPhone used by one of the shooters in the attack at the Inland Regional Center in San Bernardino, CA, last December. The FBI wanted to examine the contents of the phone, but this particular phone, an iPhone 5c running iOS 9, stores data in encrypted form using a key derived

from the user's passcode. User passcodes can be as short as four numeric digits, but Apple designed the phone to resist brute-force attacks by inserting increasingly long delays between unsuccessful login attempts and, if enabled by the user, by wiping the phone of the necessary key information after 10 unsuccessful attempts.

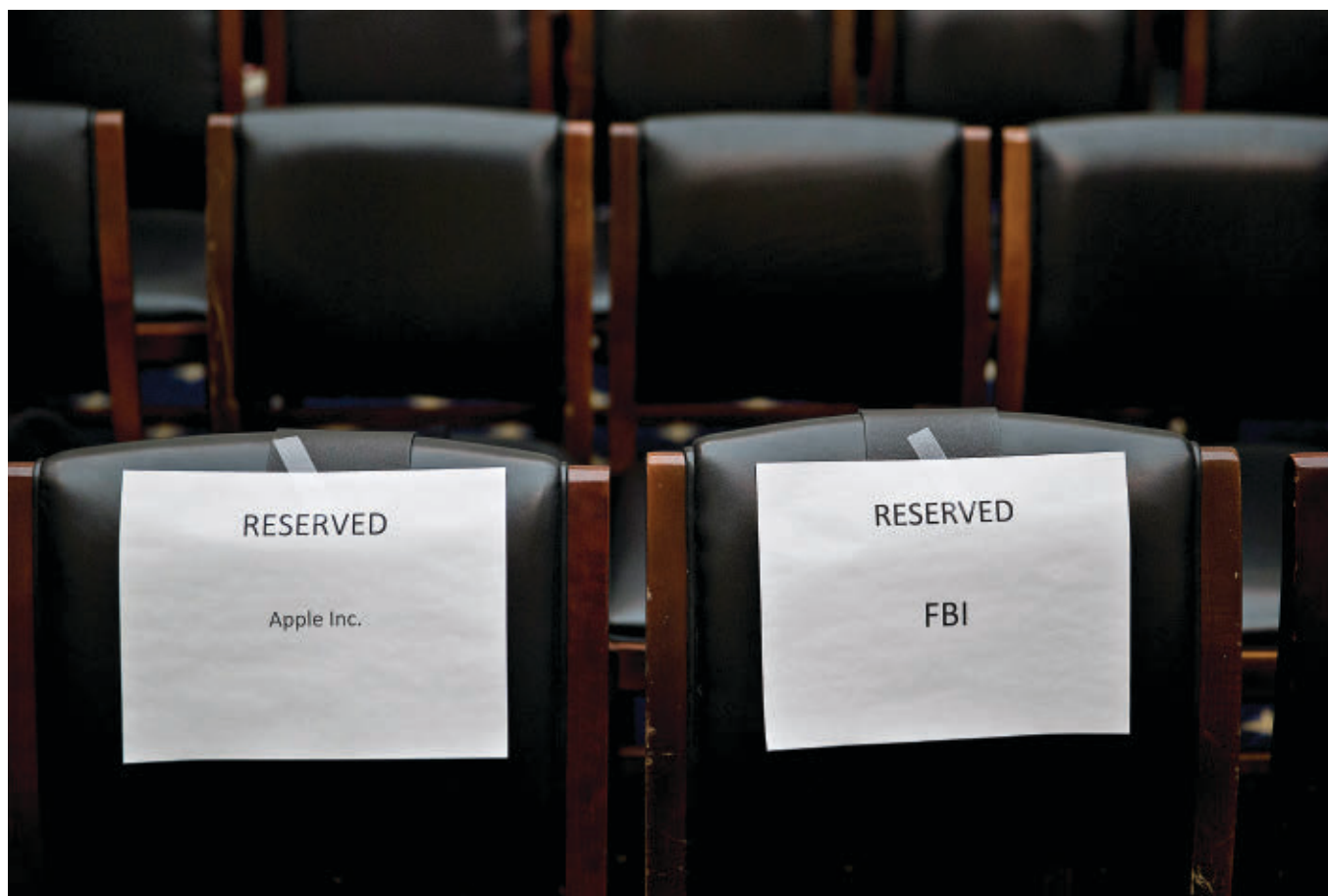
On February 16, 2016, the FBI sought and obtained a court order compelling Apple to design and sign a system update, tied to this particular phone, which would disable the impediments to a brute-force attack. Apple refused

to comply, challenging the legality of the court's order. After a round of briefings, and on the eve of a court hearing, on March 28 the FBI withdrew its request after it obtained the assistance of an unknown third party in circumventing the iPhone security measures and acquiring the decrypted contents it was seeking.

In its brief challenging the order, Apple made two types of arguments: constitutional and statutory. That is, Apple argued both that compelling its assistance would violate the Constitution and that such compulsion was beyond the powers that Congress had given to the federal courts. The distinction is important because while a ruling that the compulsion was beyond the court's statutory power could be changed by Congress—and there might be substantial pressure on Congress to make such a change—a ruling that the compulsion violated the Constitution would be beyond Congress's power to change.

Apple's primary constitutional argument was that compelling its assistance would violate the First Amend-

It should be neither impossible nor easy for the government to compel this sort of assistance.



Empty chairs reserved for Apple and the FBI in preparation for the U.S. House Committee on the Judiciary hearing in Washington, D.C., on Tuesday, March 1, 2016.

ment right to freedom of expression. The argument is seductively simple, almost syllogistic. Step one: courts have previously recognized computer code as a form of speech protected by the First Amendment. Step two: the Supreme Court has long held that the government can violate the First Amendment not only by restricting someone's speech, but also by compelling someone to speak. Ergo, forcing Apple to write code that it did not want to write would be compelling it to speak, in violation of the First Amendment. QED.

The trouble with the argument is that the syllogism breaks down because the reasons for protecting computer code have nothing to do with the reasons for forbidding compelled speech. Computer code merits at least some First Amendment protection because it can be used to communicate. If I want to describe how an encryption algorithm works, for example, I might do so using code or pseudo-code. The code conveys ideas, and the government should not be able to stop me

from explaining how an encryption algorithm works, at least not without a very good reason.

Compelling speech is problematic for an entirely different reason. Compelled speech does not stop anyone from saying or explaining anything. Nothing prevents a person from denying in the next breath what they were forced to say in the first one. The problem is that the very act of being forced to affirm a belief you do not hold can be a way to manipulate those beliefs and runs counter to the freedom to choose one's own beliefs. The classic Supreme Court case striking down a speech compulsion was one involving schoolchildren compelled to say the Pledge of Allegiance.

The code that the government would force Apple to produce is not a set of beliefs, and producing code is not like affirming beliefs. For starters, it is not meaningful to talk about the beliefs of Apple, the company. Even if we look at actual Apple employees who would be directing or writing code, if

they object, it is not because the code itself would be a misstatement of their beliefs. What they would be objecting to is being forced to do something that they don't believe in. But the government forces people to do things they do not believe in all the time: to pay taxes, to serve on a jury, even to stop at a stop sign. None of those things even come close to raising a First Amendment issue. Neither should forcing Apple to write code.

The government's access to information is also limited by the Fourth Amendment, which prohibits "unreasonable searches and seizures," but one way to make a search or seizure reasonable is to obtain a valid warrant, which the government had in these circumstances. Thus, there was no viable Fourth Amendment argument in the case, and Apple did not make one.

Apple's other arguments were statutory. In particular, Apple argued that the law the FBI and the court had invoked, the All Writs Act (AWA), did not in fact authorize the order at issue.

The AWA, first passed in 1789, states quite simply, “The Supreme Court and all courts established by Act of Congress may issue all writs necessary or appropriate in aid of their respective jurisdictions and agreeable to the usages and principles of law.” That very general language leaves to the courts the task of sorting out what counts as “necessary or appropriate” and “agreeable to the usages and principles of law.”

It is well established that the AWA only applies to situations that Congress has not specifically addressed. Apple argued that Congress had addressed this issue, in its favor, in a law called the Communications Assistance for Law Enforcement Act (CALEA). CALEA has language saying that the government cannot mandate the design of telecommunications systems and cannot require telecommunications carriers to decrypt user-encrypted communications.

But as relevant to this case, Apple is not a telecommunications carrier. Nor would it want to be, because CALEA’s main effect is to mandate that telecommunications carriers be able to intercept the communications they carry. The point of CALEA was to ensure phone companies, and later VoIP and broadband companies, could intercept communications when presented with a lawful order to do so. In return, the government would not tell the phone companies how to choose among designs with the required capabilities, and would not make the carriers responsible for user encryption. Since Apple is not subject to CALEA’s mandate, there is no reason to think it can take advantage of CALEA’s limits either.

This means we are in a situation that Congress has not explicitly addressed. The extent to which the AWA can be used in such situations is quite uncertain, because there have not been many such cases over the years. In fact, there is essentially a single relevant Supreme Court case. In that case, decided in 1977, the Supreme Court determined that the New York Telephone Company could be compelled under the AWA to install what is known as a “pen register” on its telephone equipment in order to record the phone numbers dialed from a par-

How much increased security risk is acceptable depends on what the countervailing benefits are.

ticular phone. In so doing, the Court talked about a number of different factors that supported its conclusion, but was not clear about whether any of the factors were strictly necessary, and how, if at all, the factors were supposed to be weighed against each other.

Later cases have derived three factors from the language of the New York Telephone case. The first is whether the company is “a third party so far removed from the underlying controversy that its assistance could not be permissibly compelled.” Apple argued that it was “far removed” from the controversy, because while New York Telephone was asked to install something on its own equipment, Apple did not own the iPhone at issue. This cannot be right. Because a system update must be signed by Apple to be accepted by the iPhone, Apple is uniquely positioned to provide the assistance the FBI sought. Moreover, the assistance is necessary because of the design choices that Apple made. That isn’t necessarily reason enough to compel the assistance, but it does suggest the Supreme Court’s language about parties being far removed wasn’t meant to categorically exclude a company in Apple’s position.

The last two factors from the New York Telephone case are the burden on the company to provide the assistance and the necessity for the government to obtain the assistance. It is with respect to these factors that Apple’s arguments are most convincing, but in a way that depends on the facts of each individual case. The key

burden here is not so much the effort of writing the code, for which Apple would be reimbursed, but the security risk that would be created as a result. Even if Apple tied the code it produced to this particular phone and only deployed it within the company, that internal development and deployment would still increase the risk that other similar iPhones could be broken if that knowledge were leaked, although the risk would be much smaller than if Apple were to produce general-purpose code that it delivered to the FBI.

How much increased security risk is acceptable depends on what the countervailing benefits are. That benefit varies depending on the nature of the case and the role of the particular device in the case, as well as what alternatives, if any, the government has. Lack of alternatives, standing alone, shouldn’t necessarily win the day. In these cases, the government consistently argues that lack of access to data means that crimes will go unsolved. But that is a trade-off we make throughout the rules that govern criminal procedure. Sometimes we let crimes go unsolved to serve other societal goals. Moreover, it may be helpful to ask what we’re comparing to when we say that crimes are going unsolved. Access to phone data is certainly better for law enforcement than no access. On the other hand, these phones collect and store data that 20 years ago simply would not have existed.

As for how the balance should have been struck in the San Bernardino case, I am actually somewhat ambivalent. Apple could have made a better case for what the security risk would have been, beyond just calling it a “backdoor,” while the government could have made a better case for its need, beyond just calling it a “national security” case. The important point is that when the next fight comes—and it will—neither side should be able to claim an easy win without a careful look at the facts. 

Felix Wu (felix.wu@yu.edu) is Professor of Law at the Benjamin N. Cardozo School of Law in New York City, as well as a Faculty Director of the Cardozo Data Law Initiative and Co-Director of the Cardozo Intellectual Property + Information Law Program.

Copyright held by author.



DOI:10.1145/2971327

Peter J. Denning

The Profession of IT Software Quality

Software users are looking more and more for software that delights.

“THERE ARE TWO common aspects of quality: One of them has to do with the consideration of the quality of a thing as an objective reality independent of the existence of man. The other has to do with what we think, feel, or sense as a result of the objective reality. In other words, there is a subjective side of quality.”

—W.A. Shewart, 1931⁶

Quality and dependability of software systems are among the greatest concerns of computing professionals. In the 1970s Barry Boehm,² James McCall,⁵ and their colleagues devised models for measuring software quality, which were eventually folded into an international standard ISO 9216.^{1,4} The standard metrics measure objective characteristics of the software. But, as Shewart noted long before software existed, quality is ultimately a subjective assessment made by humans reacting to artifacts. In other words, quality is in the eye of the beholder. Today much software falls short of the ISO standard and yet is more popular than systems that meet the standard. What standards of assessment are modern software users using? I will propose a preliminary answer here.

Traditional Code-Level Standards

Software developers have traditionally sought to produce software that is dependable, reliable, usable, safe, and secure. They wanted to find objective measures to support these goals, attributes that can be observed and quantified in the software itself. They developed software quality measures based on the notion that quality is strongly



related to rigor in the specifications and texts that appear throughout the software design process. Program construction techniques that maintain tight relationships between program structure and specifications are considered to be essential. The basic documents for software quality assessment list 20 measurable factors to assess overall quality:^{1,2,4,5}

- ▶ correctness
- ▶ reliability
- ▶ integrity
- ▶ usability
- ▶ efficiency
- ▶ maintainability
- ▶ testability
- ▶ interoperability
- ▶ flexibility
- ▶ reusability
- ▶ portability

- ▶ clarity
- ▶ modifiability
- ▶ documentation
- ▶ resilience
- ▶ understandability
- ▶ validity
- ▶ functionality
- ▶ generality
- ▶ economy

Each of these factors can be expanded and elaborated in detail, resulting in a complex set of rules for programmers to follow. No one said that quality is simple and straightforward.

Today there are huge markets for software in the form of apps downloadable to a mobile device or desktop and in the form of cloud services. The Android and Apple apps stores offer approximately 1.5 million apps each. This is a different environment from

the one in which the international standards were developed, when consumer software was relatively uncommon. Under pressure to beat competitors to market, software developers routinely deliver software with bugs. Moreover, many software systems are so big and complex there is no hope to deliver them without bugs. For example, among operating systems, the Linux kernel has 15 million lines of code, Windows 10 has 50 million, macOS 10.4 has 86 million, and full open source Debian release of Linux has 420 million (all according to Wikipedia). These systems are riddled with flaws, which contribute to buggy application software and cyber exploits.

Emerging User-Level Standards

Let us consider how this looks from a user perspective. Users do not ask, “Is this software well structured by ISO 9126 standards?” but rather “Does this software help me get my work done? Can I depend on it?” What does it mean to the user to get work done or depend on software? There is a strong correlation between the user’s experience of satisfaction and the user’s assessment of quality. I see six distinct levels—four positive and two negative (see the accompanying table). I will discuss them from the bottom up.

Level -1: No trust. Users do not trust the software. It may be full of bugs, crash their systems, or carry malware. You might think users would avoid untrusted software. But users do often use untrusted software—for example, after being lured by fraudulent pitches, phishing, visits to compromised websites, overwhelming desires for convenience, and the like.

Level 0: Cynical satisfaction. Many users trust some but not all the claims made by the software maker—but they trust enough to be cynically willing to use it. Much software is released with bugs and security vulnerabilities, which the developers fix only after hearing user complaints and bug reports. User forums are rife with stories about how the software has failed them and with requests for workarounds and fixes; representatives of the developers are usually nowhere to be seen in these forums. A combination of factors facilitates this situation including strong pushes to get something workable to

Six User Levels of Software Quality Assessments

4	Software delights
3	Software produces no negative consequences
2	Software fits environment
1	Software fulfills all basic promises
0	Some trust, begrudging use, cynical satisfaction
-1	No trust

market before the competition, belief that users will tolerate many bugs, and a lack of liability codified in the license agreements users must sign to unlock software. This approach to software delivery is coming under fire because the many bugs are also security vulnerabilities. Cynical users have no loyalty and will desert to another producer who makes a better offer.

Level 1: Software fulfills all basic promises. The user assesses the producer has delivered exactly what was promised and agreed to. This might be called “basic integrity.” The ISO standard addresses this level well.

Level 2: Software fits environment. The user assesses the software is a good fit to the user’s environment. This means several things. The practices and routines to use the software align with other practices and routines already in the environment; for example, because an ATM implements familiar practices of making bank transactions, users can use an ATM immediately without having to learn anything special or new. The software does not enable or encourage actions that violate social or cultural norms in the environment. The user has the experience that the software improves the user’s ability to get work done and to carry out important tasks.

Level 3: Software produces no negative consequences. After a period of use, the user has encountered no unforeseen problems that cause disruption or losses. The user assesses that the product’s design has been well thought out and that it anticipates problems that were not apparent at the outset.

Negative consequences can arise in numerous ways: The software carries vulnerabilities that can be exploited by hackers and malware. The software itself contains malware that can steal, damage, or destroy user data. The user attempts an action that was not intend-

ed or considered by the designers and the software misbehaves and damages the environment or data. A user makes a mistake with the software and there is no provision to back out to a previous good state. Over time, users develop new expectations that cannot be met by the current capabilities of the software. There can be unforeseen interactions between the many copies of the same software distributed throughout a network—for example, the stock market crash of 1987 occurred when a large number of computers programmed to sell when prices dropped by more than a preset amount automatically issued sell orders, driving prices down and triggering more selling by other computers. Operating system security vulnerabilities are another example: any or all of millions of systems can be attacked via a single vulnerability.

From long experience, a good software designer may include functions the user did not ask for but that will spare the user unwanted future problems. An example is the Apple Time Machine continuous backup system; the user can retrieve any previous version of a file and can transfer an entire file system to a new computer quickly. Another example is the Microsoft Secure Development Lifecycle, a set of internal management controls that significantly reduced bugs and security vulnerabilities in Microsoft systems.³ The designer will continue to work with the customer after the software is installed in order to modify the software in case negative consequences are discovered. These actions—anticipation and continued availability after delivery—are essential for a software producer to earn the user’s satisfaction at this level.

Level 4: Software delights. At this level the product goes well beyond the user’s expectations and produces new, unexpected, sometimes surprising positive effects. The user expresses great delight with the product and often promotes it among others. The user feels the producer understands the user’s world and contributes to the user’s well being.

Very few software systems have produced genuine delight. Some early examples include the Unix system, which was elegant and enabled powerful operations with simple commands; the Apple Macintosh, which brought an

incredibly easy to use desktop with a bitmapped display; the DEC VAX VMS, which was amazingly stable and retained previous versions of files for fast recovery; VisiCalc, the first automated spreadsheet, which made easy accounting available to anyone; Lotus 1-2-3, a successor of VisiCalc, which enabled arbitrary formulas in cells and opened a new programming paradigm; and Microsoft Word, which made professional document formatting easy and eventually effectively banished most other word processors from the market.

Recent examples include the iPhone and Android operating systems, which allow customizable access to millions (literally) of downloadable apps. Among the apps themselves some have attained high delight ratings; for example, many airlines, publishers, and newspapers offer apps that give direct access to their content via a mobile device. Some apps give users access to networks where data from many others is aggregated to give the user something that saves a lot of time and anxiety. For example, Amazon created the Kindle reader service that enables users to purchase e-books from the Amazon store and begin reading them instantly from any device with a Kindle app. Google and Apple maps use location information from smartphones to detect traffic congestion, overlay it on street maps, and propose alternate routes around congested areas. Blizzard Entertainment accumulated as many as 10 million subscribers to its World of Warcraft online game because of its rich complexity and realistic graphics. Uber allows users to hail rides whose drivers come to their exact location within minutes. In each case, customers found they could do previously impossible things with the app than without, much more than they expected.

The interesting thing about these examples is that many failed important ISO metrics such as portability, speed, efficiency, or reliability. Yet people ignored those shortcomings and became avid and loyal subscribers to the software developer.

Software developers are banking on new delights as artificial intelligence technology matures. Many people are looking forward to driverless cars, personal assistants that know your daily routines and keep you from becoming

forgetful, and virtual reality tools that allow you to tour distant places, train risk-free for a new skill or environment, or access new kinds of entertainment.

But delight is ephemeral if based on the software itself: Having mastered the new environment, the user will expand horizons and expect more. Few would find the original Unix, Macintosh, VMS, VisiCalc, or Word to be delightful today. Software producers now invest considerable effort into anticipating what will delight their users in the future. Their question has to be: Will we be able to provide delightful surprises for customers with growing expectations?

Conclusion

I have argued that software quality evaluation has transformed significantly from code-level measures of the 1970s to user-level assessments today. I proposed six levels at which users assess software quality. The levels reflect different degrees of emphasis on user satisfaction. Program correctness is essential but is limited to quality at the first level. The highest level—delight—arises in the context of the relationship between the customer and performer. The delighted customer will say that the performer has taken the trouble to understand the customer's work and business, is available to help with problems and to seize opportunities, and generally cares for the customer. Software producers today look to designs and services that produce genuine delight. When they succeed we witness new waves of killer apps. 

References

1. Blekinge Institute of Technology. Software quality models and philosophies; <http://bit.ly/1VXv15L>.
2. Boehm, B., Brown, J., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M. *Characteristics of Software Quality*. North Holland, 1978.
3. Howard, M. and Lipner, S. *The Security Development Lifecycle SDL*. Microsoft Press, 2006.
4. International Standards Organization. ISO 9126 on software quality. 1993 (standard was updated in 2001 and 2011); <http://bit.ly/1ZOpIEY>.
5. McCall, J., Richards, P., and Walters, G. Factors in software quality. *National Technical Information Service 1,2,3* (1977).
6. Shewart, W.A. *Economic Quality Control of Manufactured Products*. Van Nostrand, 1931.

Peter J. Denning (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, CA, is Editor of *ACM Ubiquity*, and is a past president of ACM. The author's views expressed here are not necessarily those of his employer or the U.S. federal government.

Copyright held by author.

Calendar of Events

September 3–7

ASE '16 ACM/IEEE International Conference on Automated Software Engineering
Singapore,
Contact: David Lo,
Email: davidlo@smu.edu.sg

September 6–9

MobileHCI '16: 18th International Conference on Human-Computer Interaction with Mobile Devices and Services
Florence, Italy,
Sponsored: ACM/SIG,
Contact: Fabio Paterno,
Email: fabio.paterno@isti.cnr.it

September 6–10

TAPIA '16: Richard Tapia Celebration of Diversity in Computing Conference
Austin, TX,
Sponsored: ACM/SIG,
Contact: Valerie E Taylor,
Email: vtaylor@tamu.edu

September 8–9

ESEM 2016: ACM/IEEE 9th International Symposium on Empirical Software Engineering and Measurement
Ciudad Real, Spain,
Sponsored: Other Societies,
Contact: Marcela Genero,
Email: marcela.genero@uclm.es

September 8–12

ICER '16: International Computing Education Research Conference
Melbourne, Australia,
Contact: Brian Dorn,
Email: bdorn@unomaha.edu

September 12–16

DocEng '16: ACM Symposium on Document Engineering 2016
Vienna, Austria,
Sponsored: ACM/SIG,
Contact: Robert Sablatnig,
Email: sab@caa.tuwien.ac.at

September 12–16

ICTIR '16: ACM SIGIR International Conference on the Theory of Information Retrieval
Newark, DE,
Sponsored: ACM/SIG,
Contact: Ben Carterette,
Email: carteret@cis.udel.edu

Broadening Participation “For All” in “Computer Science For All”

Seeking to expand inclusiveness in computer science education.

ON JANUARY 30, 2016, President Obama announced the “Computer Science For All” initiative to expand K–12 computer science education in the United States: \$4 billion to prepare teachers, expanding access to materials, and building effective partnerships. The goal of this initiative is: “... to empower all American students from kindergarten through high school to learn computer science and be equipped with the computational thinking skills they need to be creators in the digital economy, not just consumers, and to be active citizens in our technology-driven world. Our economy is rapidly shifting, and both educators and business leaders are increasingly recognizing that computer science (CS) is a “new basic” skill necessary for economic opportunity and social mobility.”^a

The effort builds on Exploring Computer Science (ECS)^b and AP Computer Science Principles (CPS)^c to reach many more students. These efforts focus on high school. “All” students must include K–8 efforts by Code.org, ACM, and CSTA. This column clarifies the meaning of “For All” in “Computer Science For All.” Diverse students will take computer science, so effective curricula must embrace students with different backgrounds and accessibility needs.

a <https://www.whitehouse.gov/blog/2016/01/30/computer-science-all>

b <http://www.exploringcs.org/>

c <http://apcsprinciples.org/>



K–12 Students

The National Center for Education Statistics (NCES) says there are approximately 50 million students in U.S. K–12 education. Table 1 shows numbers as of 2012 and projections to 2024. About 15% of students in K–12 are identified as having a disability. The 6.4 million students listed include only those with the act’s Individualized Education Program (IEP) under the Individuals with Disabilities Education Act (IDEA). An additional million or so have Section 504 plans under the Rehabilitation Act of 1973. They have the same education goals as peers and require sign language interpreters, real-time captionists, extended time to complete assignments and tests, note-takers,

Braille or electronic books, and other common accommodations.

Four years ago almost half of K–12 students were non-white. The population is headed toward 55% non-white due to increases in Hispanic, Asian/Pacific Islander, and mixed-race students. Women are about at 50% of K–12 students, but along with Blacks, Hispanics, Native Americans, and students with disabilities, they are underrepresented in computing fields.

Challenges

Reaching all K–12 students with computer science requires enough computer science teachers. Approximately eight years ago the U.S. National Science Foundation (NSF) created the CS

10K initiative to create two new high school courses: Exploring Computer Science (ECS), a gentle introduction to computer science that is taught with sensitivity to equity issues; and Computer Science Principles (CSP), a framework that has been implemented as a course in many ways using different approaches and computer languages. CSP can be an AP course if the curriculum is officially endorsed by the College Board. Starting in 2017 the AP CSP can serve for potential college credit or placement. ECS and CSP teachers are also trained to be sensitive to equity issues. The goal of the CS 10K initiative is to prepare 10,000 teachers in 10,000 schools to teach computer science by 2016, usually by retraining existing teachers. No one claims this goal has been reached, but in addition to NSF-funded professional development efforts, organizations like Code.org^d and Project Lead the Way (PLTW)^e are training existing teachers to teach computer science.

There is a limit to existing teachers, so pre-service preparation is also needed. Unfortunately, most computer science majors are headed to industry, so new and creative approaches are required to attract them to K–12 education. They need not be as knowledgeable as full computer science majors, but they need to be passionate about teaching and be able to implement pedagogies that promote learning and engagement for diverse learners. Schools of Education and computer science departments can work together to prepare competent K–12 computer science teachers.

K–12 CS education must also be relevant to students not traditionally engaged with CS. Women, certain minority groups, and people with disabilities have low numbers in the computing workforce. Many do not take AP Computer Science A (traditionally introductory Java programming): of 46,000 U.S. students who took it in 2015, there were 21.9% women, 3.9% Black, and 9.2% Hispanic.^f (We could not find data on students with disabilities.) Perhaps the lack of minori-

Table 1. Demographics of K–12 students in 2012 and projected to 2024. All but the last column are from NCES Table 203.60.* The last column is from NCES Table 204.30, only disabled students under IDEA. There is no projection beyond 2012. All numbers are in thousands.**

	Total	White	Black	Hispan	Asian/PI	Nat Am	Mixed	Disab
2012	49,771	25,386	7,803	12,104	2,552	534	1,393	6,401
		51.0%	15.7%	15.7%	5.1%	1.1%	2.8%	12.9%
2024	52,920	24,157	7,862	15,473	3,010	489	1,929	
		45.6%	14.9%	29.2%	5.7%	0.9%	3.6%	

* https://nces.ed.gov/programs/digest/d13/tables/dt13_203.60.asp

** https://nces.ed.gov/programs/digest/d13/tables/dt13_204.30.asp

ties taking AP CS A can be explained because the course is not taught in their schools, but women comprise approximately half of the students in most U.S. schools.

There is no silver bullet for making CS relevant to all, but the CS 10K initiative helps support ECS and CSP in the U.S. Code.org and other organizations have developed early introductions to computational thinking and programming that might yield more interest at higher levels. “Culturally responsive computing education” is an interesting approach in which the curriculum itself embodies norms and mores of students.³ For example, teachers can use culturally authentic characters, not stereotypes, as sprites when teaching Scratch.

Students with Disabilities

We are especially concerned for students with disabilities. In addition to availability and relevance, accessibility and specialized instructional approaches are needed. Curricula, including programming tools, must be accessible, and teachers must include students with disabilities in their classes. Students with disabilities comprise nearly 15% of K–12 students, and have a wide variety of disabilities (see Table 2). Most are taught alongside peers in general education settings and can learn the basics of computer science and attend college.

Challenges come in three categories: teachers’ attitudes/expectations, pedagogical approaches, and accommodations/accessible materials, including technology.

Teacher attitudes and expectations. IEPs provide specialized instruction to students with disabilities to increase academic and functional skills. An IEP for a student with a learning disability in reading might focus on increasing reading fluency and comprehension to

Table 2. Disabilities under IDEA from NCES Table 204.30.* All numbers are in thousands.

Total under IDEA in 2011-2012	6,401
Learning	2,303
Speech and Language	1,373
Health Impairments	743
Autism	455
Developmental Delay	393
Emotional Disturbance	373
Hearing	78
Orthopedic	61
Vision	28
Other	594

* https://nces.ed.gov/programs/digest/d13/tables/dt13_204.30.asp

help the student read at grade level, but for some students with disabilities expectations are lowered too much. This can create an unconscious bias that the student cannot learn computer science. Students with disabilities are often not intellectually limited and should not be deemed incapable of learning computer science. Unconscious bias should not have teachers, parents, and administrators set goals lower than they should be.

Pedagogical approaches. The Universal Design for Learning (UDL) approach provides a proactive way to increase the number of students reached.¹ The approach uses strategies such as

- ▶ scaffolding tools such as outlines, graphic organizers, and copies of projected materials with room for note taking
- ▶ large, bold fonts on uncluttered slides in presentations
- ▶ background and contextual information that provide more than the facts
- ▶ alternative explanations of important concepts in instruction for all learners.

It helps teachers address the range of learner variability in classes rather than

^d <https://code.org/>

^e <https://www.pltw.org/>

^f <http://research.collegeboard.org/programs/ap/data/participation/ap-2015>



ACM Conference Proceedings Now Available via Print-on-Demand!

Did you know that you can now order many popular ACM conference proceedings via print-on-demand?

Institutions, libraries and individuals can choose from more than 100 titles on a continually updated list through Amazon, Barnes & Noble, Baker & Taylor, Ingram and NACSCORP: CHI, KDD, Multimedia, SIGIR, SIGCOMM, SIGCSE, SIGMOD/PODS, and many more.

For available titles and ordering info, visit:
librarians.acm.org/pod

individualizing standard lesson plans for each student. Although UDL increases access and engagement for all learners (including those with disabilities), it may need to be further augmented by pedagogical approaches for specific learners with disabilities. These have not been fully studied for computer science education, but studies from other content areas show promise. Many students need explicit instruction in computational concepts and procedural approaches relevant to CS but not provided in the students' IEPs. Emerging research from the CS education community might provide better guidelines.

Accommodations and accessible materials. Accommodations for students with disabilities, typically provided in IEPs or 504 plans, increase accessibility and/or allow students to demonstrate what they know. However, an accommodation often does not reflect grade-level expectations for students. Accommodations like extra time on exams, sign language interpreters or real-time captioners, or Braille might not be sufficient if course materials are inaccessible as is the case for too many K–12 CS course materials. Typical course materials for young students include block-based programming languages like Scratch (<https://scratch.mit.edu/>), ScratchJr. (<http://www.scratchjr.org/>), Tynker (<https://www.tynker.com/>), Blockly (<https://developers.google.com/blockly/>), App Inventor (<http://appinventor.mit.edu/explore/>), and Snap! (<http://snap.berkeley.edu/>). These languages rely on visual access to a screen where students can drag and drop blocks that snap together to form programs. Outputs from these programs are typically animations. Potentially, blind students might use these with the aid of a screen reader, but none of them is accessible using a screen reader. By contrast, most programming languages are text-based, and screen readers are very good at reading text. “For All” includes the 28,000 K–12 blind students in Table 2. We encourage the developers of block languages to make them accessible and to develop non-visual activities using them. Quorum,^g a text-based language designed for young children, is screen-reader accessible, and has the interesting output command “say” that causes a speech synthesizer to speak the output

through speakers or headphones. In addition to making the block-based languages accessible to a screen reader for blind students, adding a say command for output could make them more fun for all children.

Conclusion

We argue that “For All” in “Computer Science For All” means *all* K–12 students, including the underrepresented in computing fields: women, Blacks, Hispanics, Native Americans, and students with disabilities. There are challenges to such inclusion, particularly with respect to students with disabilities. Many countries are grappling with introducing computer science to K–12 education. The United Kingdom has a grass-roots organization called Computing at Schools^h helping to coordinate nationwide efforts. According to European Schoolnet’s 2015 survey, 18 European countries have integrated or will integrate computer coding in K–12.² Centralized education systems might make changing the curriculum easier, but all will have difficulty preparing teachers. “Computer Science For All” will be interpreted differently from country to country. What is meant by “all” remains a challenge. 

^h <http://www.computingatschool.org.uk/>

References

1. Burgstahler, S. Universal design: Implications for computing education. *Trans. Comput. Educ.* 11, 3 Article 19 (Oct. 2011).
2. Computing our future: Computer programming and coding Priorities, school curricula and initiatives across Europe, by European Schoolnet; <http://bit.ly/1Zy3Mzq>
3. Eglash, R., Gilbert, J.E., and Foster, E. Toward culturally responsive computing education. *Commun. ACM* 56, 7 (July 2013), 33–36.

Richard E. Ladner (ladner@cs.washington.edu) is a professor in computer science and engineering at the University of Washington. He is the principal investigator for the two NSF-funded projects AccessComputing and Access10K, the later of which focus on access to K–12 computer science education by students with disabilities. He also has a research program that focuses on technology to benefit people with disabilities.

Maya Israel (misrael@illinois.edu) is an assistant professor of special education at the University of Illinois. Her primary areas of specialization include supporting students’ meaningful access to science, technology, engineering, and mathematics (STEM) learning through Universal Design for Learning, instructional strategies, and technologies. She is co-principal investigator on an NSF STEM+C project that is focused on integrating computational thinking into mathematics education.

This material is based upon work supported by the National Science Foundation under grant numbers CNS-1539179 and CNS-1440843.

^g <https://quorumlanguage.com/>

Copyright held by authors.

Viewpoint

Designing AI Systems that Obey Our Laws and Values

Calling for research on automatic oversight for artificial intelligence systems.

OPERATIONAL AI SYSTEMS (for example, self-driving cars) need to obey both the law of the land and our values. We propose AI oversight systems (“AI Guardians”) as an approach to addressing this challenge, and to respond to the potential risks associated with increasingly autonomous AI systems.^a These AI oversight systems serve to verify that operational systems did not stray unduly from the guidelines of their programmers and to bring them back in compliance if they do stray. The introduction of such second-order, oversight systems is not meant to suggest strict, powerful, or rigid (from here on ‘strong’) controls. Operations systems need a great degree of latitude in order to follow the lessons of their learning from additional data mining and experience and to be able to render at least semi-autonomous decisions (more about this later). However, all operational systems need some boundaries, both in order to not violate the law and to adhere to ethical norms. Developing such oversight systems, AI Guardians, is a major new mission for the



AI community.

All societies throughout history have had oversight systems. Workers have supervisors; businesses have accountants; schoolteachers have principals. That is, all these systems have hierarchies in the sense that the first line operators are subject to oversight by a second layer and are expected to respond to corrective signals from the overseers. (These, in turn, are expected to take into account suggestions or even demands by the first line to

change their modes of oversight). John Perry Barlow, in his famous “Declaration of the Independence of Cyberspace” in 1996, described the burgeoning online world as one that would be governed by a social contract formed among its users.^b

Terra Incognita

AI systems not only need some kind of oversight, but this oversight must

^a See T.G. Dietterich and E.J. Horvitz, “Rise of Concerns About AI: Reflections and Directions,” *Commun. ACM* 58, 10 (Oct. 2015), 38–40, for an in-depth discussion of the various risks.

^b See <http://bit.ly/1KavIVC>

be provided—at least in part—not by mortals, but by a new kind of AI system, the oversight ones. AI needs to be guided by AI.^c

One reason is that AI operational systems are *learning systems*. These systems do not stop collecting data once they are launched; instead, continued data mining and experience are used to improve their performance. These AI systems may hence stray considerably from the guidelines their programmers initially gave them. But no mortal can monitor these changes, let alone in real time, and determine whether they are legal and ethical.

Second, AI systems are becoming highly *opaque*, “black boxes” to human beings. Jenna Burrell from the School of Information at UC-Berkeley distinguishes three ways that algorithms become opaque: Intentional opacity, for example with proprietary algorithms that a government or corporation wants to keep secret; Technical illiteracy, where the complexity and function of algorithms is beyond the public’s comprehension; and Scale of application, where either “machine learning” and/or the number of different programmers involved renders an algorithm opaque even to the programmers.¹

Finally, AI-guided systems have increasing *autonomy* in the sense they make numerous choices “on their own.”⁵ That is, these instruments, using complex algorithms, respond to environmental inputs independently.⁶ They may even act in defiance of the guidelines the original programmers installed. A simple example is automatic emergency braking systems,³ which stop cars without human input in response to perceived dangers.⁷ Consumers complain of many false alarms, sudden stops that are dangerous to other cars,⁴ and that these brakes force cars to proceed in a straight line even if the driver tries to steer them elsewhere.

For all these reasons, AI oversight systems are needed. We call them AI Guardians. A simple dictionary definition of a guardian is: “a person who guards, protects, or preserves.”^d This

However smart a technology may become, it is still a tool to serve human purposes.

definition captures well the thesis that oversight systems need not be strong because this would inhibit the innovative and creative development of operational AI systems—but cannot be avoided. Indeed, a major mission for AI is to develop in the near future such AI oversight systems. We now describe whose duty it is to develop these oversight systems and to whom they are to report their findings and whose values they are to heed.

Different Kinds of AI Guardians

Interrogators. After a series of crashes of drones manufactured by one corporation, another corporation that purchased several hundred drones is likely to try to determine the cause of the crashes. Were they intentional (for example, caused by workers opposed to the use of drones)? Unwitting flaws in the design of the particular brand of drones? Flaws in the AI operational system that serves as the drone’s ‘brain’? For reasons already discussed, no human agent is able to provide a definitive answer to these questions. One would need to design and employ an interrogator AI system to answer the said questions.

In recent years, several incidents show the need for such interrogation. In 2015, a team of researchers from Carnegie Mellon University and the International Computer Science Institute found that Google was more likely to display ads for high-paying executive jobs to users that its algorithm believed to be men than to women.^e Google stated that there was no intentional discrimination but that the effect was due to advertisers’ preferences.^f

In 2014, Facebook conducted a study unbeknownst to its users wherein its algorithms manipulated users’ posts to remove “emotional content” in order to gauge reactions from the posters’ friends.^g Facebook later apologized for not informing its users about the experiment. Twitter recently deleted 125,000 accounts, stating that these included only accounts that were linked to the Islamic State. If a committee of the board of these corporations or an outside group sought to verify these various claims—they would need an AI monitoring system.

Auditor: Wendell Wallach, a scholar at Yale’s Interdisciplinary Center for Bioethics points out that “in hospitals, APACHE medical systems help determine the best treatments for patients in intensive care units—often those who are at the edge of death. Wallach points out that, though the doctor may seem to have autonomy, it could be very difficult in certain situations to go against the machine—particularly in a litigious society.”^h Hospitals are sure to seek audits of such decisions and they cannot do so without an AI auditing system.

Monitor: Because self-driving cars are programmed to learn and change, they need a particular kind of AI Guardian program—an AI Monitor—to come along for the ride to ensure the autonomous car’s learning does not lead it to violate the law, for example learning from the fact that old-fashioned cars violate the speed limit and emulating this behavior.

Enforcer: In rare situations, an AI Guardian may help enforce a regulation or law. For instance, if the computers of a military contractor are repeatedly hacked, an AI enforcer may alert the contractor that it needs to shore up its cyber defenses. If such alerts are ignored, the AI enforcer task will be to alert the ‘clients’ of the contractor or to suspend its clearance.

Ethics bots: AI operational systems must not only abide by the law, but also heed the moral norms of society. Thus driverless cars need to be told whether they should drive at whatever speed the law allows, or in ways that conserve fuel to help protect the environment, or to stay in the slower lanes if children are

c See D. Weld and O. Etzioni, “The First Law of Robotics (a call to arms),”⁸ for an early attempt to formalize a solution to this problem.

d See <http://bit.ly/28Y3gGG>

e See <http://bit.ly/28Xy0pT>

f See <http://bit.ly/292qE9h>

g See <http://bit.ly/23KzDS3>

h See <http://bit.ly/1N4b5WY>

in the car. And—if they should wake up a passenger in the back seat if they “see” an accident.

Several ideas have been suggested as to where AI systems may get their ethical bearings. In a previous publication, we showed that asking each user of these instruments to input his or her ethical preferences is impractical, and that drawing on what the community holds as ethical is equally problematic. We suggested that instead one might draw on ethics bots.²

An ethics bot is an AI program that analyzes many thousands of items of information—not only information publicly available on the Internet but also information gleaned from a person’s own computers about the acts of a particular individual that reveal that person’s *moral* preferences. And then uses these to guide the AI operational systems (for instruments used by individuals, such as the driverless cars).

Essentially, what ethics bots do for moral choices is similar to what AI programs do when they ferret out consumer preferences and target advertising accordingly.¹ In this case, though, the bots are used to guide instruments that are owned and operated by the person, in line with their values—rather than by some marketing company or (political campaign). For instance, such an ethics bot may instruct a person’s financial program to invest only in socially responsible corporations, and in particular green ones, and make an annual donation to the Sierra Club, based on the bot’s reading of the person’s past behavior.

In short, there is no reason for the digital world to become nearly as hierarchical as the non-digital one. However, the growing AI realm is overdue for some level of guidance to ensure AI operational systems will act legally and observe the moral values of those who own and operate them.

It is not necessarily the case that AI guardians are more intelligent than the systems they oversee. Rather, the guardians need to be sufficiently ca-

pable and intelligent that they are not outwitted or short-circuited by the systems they are overseeing. Consider, for example, an electrical circuit breaker in a home: it is far less sophisticated than the full electrical system (and associated appliances) but it is quite reliable, and can be “tripped” by a person in an emergency.

AI researchers can work toward this vision in at least three ways. First, they can attempt to formalize our laws and values following an approach akin to that outlined in the work on formalizing the notion of “harm.”⁸ Second, researchers can build labeled datasets identifying ethical and legal conundrums labeled by desired outcomes, and provide these as grist for machine learning algorithms. Finally, researchers can build “AI operating systems” that facilitate off switches as in the work on “safely interruptible agents” in reinforcement learning.¹ Our main point is that we need to put AI guardians on the research agenda for the field.

Who Will Guard the AI Guardians?

There are two parts to this question. One aspect concerns who will decide which AI oversight systems will be mobilized to keep in check the operational ones. Some oversight systems will be introduced by the programmers of the software involved at the behest of the owners and users of the particular technologies. For example, those who manufacture driverless cars and those who use them will seek to ensure that their cars will not speed ever more. This is a concern as the cars’ operational systems—which, to reiterate, are learning systems—will note that many traditional cars on the road violate the speed limits. Other AI oversight systems will be employed by courts and law enforcement authorities. For instance, in order to determine who or what is liable for accidents, and whether or not there was intent.

Ethics bots are a unique AI Guardian from this perspective. They are to heed the values of the user, not the owner, programmer, or those promoted by the government. This point calls for some explanation. Communities have two kinds of social and

moral values. One kind includes values the community holds, which are of particular importance and hence their implementation cannot be left to individual choice; heeding them is hence enforced by coercive means, by the law. These values include a ban on murder, rape, theft and so on. In the AI world, heeding these is the subject of a variety of AI Guardians, outlined earlier. The second kind of values concern moral choices the community hold it can leave to each person to decide whether or not to follow. These values include whether or not to donate an organ, give to charity, volunteer, and so on. These are implemented in the AI world by ethics bots.

The question of who will guard the guardians rises. Humans should have the ultimate say about the roles and actions of both the AI operational and AI oversight systems; indeed, all these systems should have an on and off switch. None of them should be completely autonomous. Ultimately, however smart a technology may become, it is still a tool to serve human purposes. Given that those who build and employ these technologies are to be held responsible for their programming and use, these same people should serve as the ultimate authority over the design, operation, and oversight of AI. **□**

References

1. Burrell, J. How the machine ‘thinks’: Understanding opacity in machine learning algorithms. *Big Data & Society* 3, 1 (2016).
2. Etzioni, A. and Etzioni, O. AI assisted ethics. *Ethics and Information Technology* 18, 2 (2016), 149–156; <http://bit.ly/28Yymx0>
3. Kapran, C. Auto-braking: A quantum leap for road safety. *The Telegraph*, (Aug. 14, 2012); <http://bit.ly/2917jog>.
4. Limer, E. Automatic brakes are stopping for no good reason. *Popular Mechanics*, (June 19, 2015); <http://bit.ly/28XVSxP>.
5. Mayer-Schönberger, V. and Cukier, K. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. 2014, 16–17.
6. New algorithm lets autonomous robots divvy up assembly tasks on the fly. *Science Daily*, (May 27, 2015); <http://bit.ly/1FFCijX>.
7. Phelan, M. Automatic braking coming, but not all systems are equal. *Detroit Free Press*, (Jan. 1, 2016); <http://on.freep.com/2917nnZ>.
8. Weld, D. and Etzioni, O. The First Law of Robotics (a call to arms). In *Proceedings of AAAI ’94*. AAAI, 1994; <http://bit.ly/292kpSK>

Amitai Etzioni (etzioni@gwu.edu) is a University Professor of Sociology at The George Washington University, Washington, D.C.

Oren Etzioni (orene@allenai.org) is CEO of the Allen Institute for Artificial Intelligence, Seattle, WA, and a Professor of Computer Science at the University of Washington.

Copyright held by authors.

i Ted Cruz’s campaign in Iowa relied on psychological profiles to determine the best ways to canvass individual voters in the state. T. Hamburger, “Cruz campaign credits psychological data and analytics for its rising success,” *The Washington Post* (Dec. 13, 2015); <http://wapo.st/1NYgFto>.

j See <http://bit.ly/1RVnTA1>

Viewpoint

Helping Conference Attendees Better Understand Research Presentations

Sharing lessons learned from a lecture program for making technical material more accessible to conference attendees.

FOR OVER 20 YEARS, ACM SIGCOMM has supported activities to make its conferences more accessible to an ever-wider range of attendees, with activities such as travel grants, student events, and mentoring programs. During the mentoring program at ACM SIGCOMM 2014, one of the co-authors of this Viewpoint—Ethan Katz-Bassett—found that students lacked enough background in the topics of many of the research papers, causing them to struggle to understand the presentations or to engage other attendees in conversations. That led us to put together a program of short lectures at ACM SIGCOMM 2015, designed to prepare attendees to better understand the presentations at subsequent technical sessions.

The lectures were extraordinarily successful. We overfilled the lecture rooms and, unfortunately, still had to turn some attendees away. A questionnaire circulated after the conference showed 88% of the lecture attendees felt the previews helped them get more from the technical talks. Although our initial intention was to provide a session to help *students* understand the conference material, to our surprise many attendees were established researchers seeking to understand a new area.

Clearly the lectures met a need. In the remainder of this Viewpoint, we try



to characterize that need, explain how we organized the lectures, and discuss insights and lessons learned. The lessons we present are anecdotal—not scientific—but we hope that they spur discussion on how to make technical material more accessible to attendees, a step toward both better communicating advances to the research commu-

nity and making the community more welcoming. We are in the process of organizing similar lectures for ACM SIGCOMM 2016.

A Field Keeps Getting Bigger and Broader

The ACM Special Interest Group in Data Communications was estab-

lished in 1969, but it was not until the mid-1980s that it began to host its namesake annual technical conference, ACM SIGCOMM. At the time, the field of data communications was small, and there were less than a handful of conferences. ACM SIGCOMM rapidly became a preferred venue for presenting top research, and, well into the 1990s, an ACM SIGCOMM attendee could expect to hear presentations on at least half the major research results of the year.

Twenty years later, in a data communications field with many more conferences and exponentially more researchers, ACM SIGCOMM continues to seek to present some of the best research of the year in a three-day single-track program. But the intellectual experience for a graduate student has become far more challenging. In the 1980s, a student could read less than 100 papers and have a good sense of the field (an example of this sort of reading collection is Partridge¹). Now individual papers may require at least some understanding of a dozen or more prior papers, and, in hot topics, the research progress in a year can be substantial. What the students were telling Ethan Katz-Bassett last year was the volume of background material required was too big, and it was inhibiting their ability to fully benefit from the research presentations—and hallway conversations—at the conference.

SIGCOMM has also become a large conference. SIGCOMM 2015 had 608 registrations, of which 171 (28%) were from students. A significant proportion of the students were “junior” graduate students—40% identified themselves as first-year graduate students. In addition, there were five undergraduate students who registered for the conference. These students, graduates and undergraduates, were the target audience for the lecture program.

An interesting thing we learned as we set up the lecture program was that students are not the only ones who could benefit from a background lecture. The quickening pace of research results can affect senior researchers too. As we began circulating information about the program, several senior researchers commented that they too would try to attend too. One commented,

Students are not the only ones who could benefit from a background lecture. The quickening pace of research results can affect senior researchers too.

“I haven’t worked in some areas in a few years, and it would be good to know what’s been happening before I listen to the talks.” And, indeed, a number of senior researchers attended the lectures.

How the Lectures Worked

At SIGCOMM 2015, we scheduled two 50-minute lectures, each of which had four 10-minute talks. Each talk covered a topic area in the conference and corresponded to at least one paper session in the conference. The first lecture was held Monday evening, before the conference reception, and the other was held on Wednesday during the lunch break. For the Wednesday session, an industry sponsor provided lunch.

We advertised the sessions by sending an email announcement to all registered attendees, posting to social media (including the SIGCOMM Facebook group), putting up posters at the conference venue, and announcing the Wednesday lecture at the conference’s opening session and at the end of the session just before the lecture.

For each topic area, we asked a speaker to provide background information the speaker thought would help the audience appreciate and understand the conference’s technical talks on that subfield of networking. We selected as speakers individuals with influential work in the topic area, published within the last few years. There were eight speakers: three industry researchers, one grad-

uate student, and four junior faculty members. Five of the eight speakers were women.

A typical presentation included a brief explanation of the topic (often as one might learn in a graduate networking class) followed by an explanation of the current problems the sub-field was seeking to address, where the conference’s papers were seeking to push the field forward, and, where appropriate, how the set of papers interrelated. In addition to setting out the context for the coming technical talks, many of the speakers (unprompted by the organizers) used their presentation to drum up enthusiasm and advertise their topics’ talks. Presenters jokingly argued with each other over which areas were more important, interesting, or challenging.

For logistical reasons, we could not use the main lecture hall where the conference presented papers. Rather we were in a substantial classroom that seated approximately 60 people. The room was overwhelmed for both sessions, with people sitting on the floor and standing in the back.

Post Conference Survey

After the SIGCOMM 2015 conference ended, we asked people to respond to a questionnaire about the preview lectures, regardless of whether they had attended. We advertised to all registered attendees and via social media, and we received 59 responses. Of those responding, 20 indicated they had not attended the lectures. The comments revealed that at least some of these 20 were faculty members who were summarizing their students’ experiences and also some attendees who had been unable to attend the previews, but had made use of the slides (which we placed online). Of those who attended, just under half (47%) attended both sets of preview lectures.

Everyone polled said they would recommend that students attend the preview lectures in the future and all but one person said we should offer the program at ACM SIGCOMM 2016. When asked whether the talks helped attendees get more out of the technical talks, 39% of those surveyed said the preview lectures were very useful, 37% said somewhat useful, 10% said only a

COMMUNICATIONS APPS

Access the latest issue, past issues, **BLOG@CACM**, News, and more.



Available for iPad, iPhone, and Android



Available for iOS, Android, and Windows

<http://cacm.acm.org/about-communications/mobile-apps>



Association for Computing Machinery

For some students, the lectures turned out to be their first exposure to some subfields of data communications.

little useful, and the remainder were unsure. Interestingly, similar percentages felt that the preview lectures also helped attendees talk with other attendees about research interests.

We asked whether we should change the format to individual lectures, or one long session of preview lectures. Attendees strongly (71%) preferred the format we (rather accidentally) had used of a set of four lectures previewing the upcoming sessions. They overwhelmingly (94%) endorsed keeping each preview lecture short (10-minute talks) and generally (92%) felt that the short talks had provided the right level of background for the next day's talks and (89%) clearly explained the motivation for the research that the talks would present.

Less clear was whether the preview lectures got the balance right in terms of how much detail about the papers was in the preview lecture vs. leaving the details to actual authors' presentations the next day. A majority felt the balance was about right (68.6%), but significant minorities wanted more (22.9%) or less (8.6%) details of the upcoming papers.

Other Lessons Learned

We learned some things, largely accidentally, while putting on the lecture program. For some students, the lectures turned out to be their first exposure to some subfields of data communications. Some of the speakers picked up on this vibe and advertising that their subfield was "cooler" became a running jest across

the lectures. Yet the impact was also real. Two students commented to co-author Justine Sherry that her presentation had led them to consider doing thesis work in her topic area. Several female students observed that the majority of the speakers were women and commented that it helps them build up confidence.

Ways We Can Improve

The most obvious way we can improve is to provide more space, so that all interested attendees can hear the lectures. While there have been suggestions to put the individual lectures on video, the unexpected benefits of clustering the talks have led us to seek to continue with the in-person multi-lecture format for SIGCOMM 2016. We also clearly have an issue engaging the audience to ask questions. The speakers received almost no questions during the lectures, despite inviting questions. Yet the post-conference questionnaire showed that 40% of attendees felt there was not enough time for questions. It would appear that something about the format unintentionally suppressed questions. We need to find ways to make it easier to ask questions. **C**

Reference

1. Partridge, C., Ed. *Innovations in Internetworking*. Artech House, Inc., Norwood, MA, 1988.

Ethan Katz-Bassett (ethan.kb@usc.edu) is an assistant professor in the computer science department at the University of Southern California.

Justine Sherry (justines@andrew.cmu.edu) is an assistant professor at Carnegie Mellon University starting in Fall 2017.

Te-Yuan Huang (thuag@netflix.com) is a senior software engineer at Netflix.

Maria Kazandjeva (mariakaz@cs.stanford.edu) is a senior software engineer at Netflix.

Craig Partridge (craig@bnn.com) is chief scientist at Raytheon BBN Technologies.

Fahad Dogar (fahad@cs.tufts.edu) is an assistant professor in the department of computer science at Tufts University.

The authors thank the individual lecturers (Justine Sherry, Laurent Vanbever, Aaron Schulman, George Porter, Te-Yuan (TY) Huang, Maria Kazandjeva, Nandita Dukkkipati, Phillipa Gill) for making the program a success. We also owe a tremendous debt to the ACM SIGCOMM 2015 local arrangements team (especially Hamed Haddadi) for dealing with logistics. Finally, a thank you to Netflix for catering the lunchtime lecture session.

Copyright held by authors.

Viewpoint

*Seeking to make Web data “smarter”
by utilizing a new kind of semantics.*

FROM THE VERY early days of the World Wide Web, researchers identified a need to be able to understand the semantics of the information on the Web in order to enable intelligent systems to do a better job of processing the booming Web of documents. Early proposals included labeling different kinds of links to differentiate, for example, pages describing people from those describing projects, events, and so on. By the late 1990s, this effort had led to a broad area of computer science research that became known as the Semantic Web.¹ In the past decade and a half, the early promise of enabling software agents on the Web to talk to one another in a meaningful way inspired advances in a multitude of areas: defining languages and standards^a to describe and query the semantics of resources on the Web; developing tractable and efficient ways to reason with these representations and to query them efficiently; understanding patterns in describing knowledge; and defining ontologies that describe Web data to allow greater interoperability.

Semantic Web Today

In fact, Semantic Web research and practice spanned the spectrum from focusing on expressivity and reasoning on the Web⁴ to providing an ecosystem of linked data that allows data

a <http://bit.ly/1gQGTot>



resources to link to one another explicitly through shared naming and equivalence statements across repositories.² Arguably, the far ends of this spectrum were ignoring the messiness of the real Web in the former case, and were not providing enough perceivable value because of lack of any organization or semantics in the latter. However, in be-

tween, there was a broad “sweet spot” where the work coming out of these communities has led to contributions that have gone beyond research and led to undeniable advances in the way that the Web works today:

- ▶ Over 2.5 billion Web pages have markup conforming to the schema.org format, which enables them to describe

precisely the structured content on their sites using a shared vocabulary.^b

► Linked data, in the form of structured, typed, and dereferencable links, powers media sites for organizations such as the BBC and *New York Times*; major libraries and museums around the world actively develop their content as linked data.

► Google, Yahoo!, Microsoft, Facebook, many other large Web companies as well as numerous research projects are developing large knowledge graphs, which define, structure, and link hundreds of millions of entities, to enhance search, to provide better advertising match, to improve the answers of their artificial personal assistants, and so on.

► Commercial database-management systems (for example, Oracle) provide native support for Semantic Web languages.

► Recommender companies are increasingly using semantics and semantic tagging to improve both the quality and accuracy of recommendations that they provide.^c

► The World Health Organization is developing the main international terminology for diseases to be used by all United Nations member countries as an ontology to be usable on the Web.⁶

The list goes on.

Semantic Web Research in Transition

As the early research has transitioned into these larger, more applied systems, today's Semantic Web research is changing: It builds on the earlier foundations but it has generated a more diverse set of pursuits. As the knowledge graphs mentioned previously increasingly use semantic representations, they have driven the functionality of a new generation of apps (mobile healthcare, mapping and shopping assistants, and others). As these applications became increasingly crucial to advertising and e-commerce, the representations they used became less formal and precise than many early Semantic Web researchers had envisioned.

As developers strive to provide structure and organization beyond

just linking of data, they are not making very much use of the formal semantics that were standardized in the Semantic Web languages. Modern semantic approaches leverage vastly distributed, heterogeneous data collection with needs-based, lightweight data integration. These approaches take advantage of the coexistence of a myriad of different, sometimes contradictory, ontologies of varying levels of detail without assuming all-encompassing or formally correct ontologies. In addition, we are beginning to see the increased use of textual data that is available on the Web, in hundreds of languages, to train artificially intelligent agents that will understand what users are trying to say in a given context and what information is most pertinent to users' goals at a given time. These projects are increasingly leveraging the semantic markup that is available on the Web; for example, the IBM Watson "Jeopardy!"-playing program made use of taxonomies and ontologies (such as DBpedia^d and YAGO^e) to increase performance significantly.³

In addition to the increasing amount of semantically annotated information on the Web, a lot more structured data is becoming available. This data includes information from scientists and governments publishing data on the Web and the ever-increasing amount of information avail-

d <http://bit.ly/2aujZ8o>

e <http://bit.ly/2asoZLi>

As the early research has transitioned into larger, more applied systems, today's Semantic Web research is changing.

able about each of us, individually and as societies—in the form of our social interactions, location and health data, activities, and interests. Harnessing this data, and understanding its diverse and often contradicting nature, to provide really meaningful services and to improve the quality of our lives, is something that researchers in both industry and academia are beginning to tackle. Statistical and machine-learning methods become more powerful and computational resources continue to improve. Thus, some of the semantic knowledge that researchers had to construct manually they can now learn automatically, tremendously increasing the scale of the use of semantics in understanding and processing Web data. While manually constructed formal ontologies may often (but not always) be required to form a backbone of semantics for the Web, much of the content that puts "meat" on those bones is "scruffy" and imprecise, often statistically induced. Indeed, the ontologies themselves might be learned or enhanced automatically. As the semantics, in a sense, becomes more "shallow," it could be more widely applicable.⁵ Consequently, our very understanding of the nature of the semantics that intelligent systems produce and leverage is changing, and with it, our vision for the future of the Semantic Web.

The Next 10 Years

As we look at the next decade of the Semantic Web, we believe these trends will continue to fuel new demands on Web researchers. Thus, these trends lead us to formulate a new set of research challenges. We believe the objective of the next decade of Semantic Web research is to make this vast heterogeneous multilingual data provide the fuel for truly intelligent applications.

Achieving this objective will require research that provides more meaningful services and that relies less on logic-based approaches and more on evidence-based ones. We note the rubrics listed here are not all that different from the challenges we faced in the past, but the methods, the scale, and the form of the level of representation language-

b <http://bit.ly/2a2fEUy>

c <http://bit.ly/1L02VhY>

es changes drastically. We present questions under each of the rubrics to guide this research.

► **Representation and lightweight semantics:** Semantic Web standards that were developed by the World Wide Web Consortium fueled early research on the Semantic Web, enabling scientists not to worry about the underlying representation languages and to publish resources that provide linking between many open databases expressed in standard formats.^f However, the world of semantics on the Web also increasingly encompasses representations in non-standard (and sometimes proprietary) formats. This diversity also applies to how formal the representations are. New questions that emerge include: How do we leverage these diverse representations? What is a broader view of what constitutes semantics on the Web? How do we coordinate the diverse components of structured knowledge that are defined by various parties and that must interact in order to achieve increasingly intelligent behavior? How do we define lightweight, needs-based, “pay-as-you-go” approaches for describing knowledge? What are the languages and architectures that will provide this knowledge to the increasingly mobile and application-based Web?

► **Heterogeneity, quality, and provenance:** It is a truism that data on the Web is extremely heterogeneous. Web resources drastically vary in size, underlying semantics, and of course, quality. A dataset precise enough for one purpose may not be sufficiently precise for another. Data on the Web may be wrong, or wrong in some context—with or without intent. Provenance has already been recognized as critical to applications using data on the Web. This heterogeneity raises a variety of questions to explore: How do we integrate heterogeneous data and particularly how can we understand which data can be integrated to what degree? How can we represent and assess quality and provenance of the data? How do we evaluate whether the quality of a particular source is sufficient for a given task?

► **Latent semantics:** Obviously,

Bringing a new kind of semantics to the Web is becoming an important aspect of making Web data smarter and getting it to work for us.

there is a lot of semantics that is already on the Web, albeit mostly in text, or in data that machines cannot readily interpret. To complement formally developed ontologies, we must be able to extract latent, evidence-based models that capture the way that users structure their knowledge implicitly. We need to explore these questions: How much of the semantics can we learn automatically and what is the quality of the resulting knowledge? As ontologies are learned or enhanced automatically, what is the very meaning of “formal ontologies”? How do we develop some notion of approximate correctness? Do similar or different reasoning mechanisms apply to the ontologies that are extracted in this way? How do crowdsourcing approaches allow us to capture semantics that may be less precise but more reflective of the collective wisdom?

► **High volume and velocity data:** While the challenges of the growing “Internet of things” are just starting to emerge, already we see scientists and developers trying to come to grips with the problems caused by the high volume and velocity of the sensory data that is streaming to the Web. New research must explore these questions: How do we triage the data in motion to determine what to keep and what we may choose, or need, to allow to be lost? How do we deploy simple decision-making agents in such applications, and what are the semantic needs of such agents? How can our applications integrate con-

stantly changing sensor data with fixed data of long duration and high-quality semantic provenance?

In short, bringing a new kind of semantics to the Web is becoming an increasingly important aspect of making Web data smarter and getting it to work for us. We believe our fellow computer scientists can both benefit from the additional semantics and structure of the data available on the Web and contribute to building and using these structures, creating a virtuous circle. The techniques of the early Semantic Web research have defined many of the parameters that we need in order to understand these new approaches and have provided important data resources to the community exploring how to build new Web-based applications. Continued research into Web semantics holds incredible promise, but only if we embrace the challenges of the modern and evolving Web. ■

References

1. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American* 284, (2001), 34–43; DOI:10.1038/scientificamerican0501-34.
2. Bizer, C., Heath, T., and Berners-Lee, T. Linked data—The story so far. *International Journal on Semantic Web and Information Systems*, 5, (2009), 1–22. DOI:10.4018/jswis.2009081901.
3. Ferrucci, D. et al. Building Watson: An overview of the DeepQA Project. *AI Magazine* 31, 3 (2010), 59–79; DOI: 10.1609/aimag.v31i3.2303.
4. Horrocks, I., Patel-Schneider, P., and van Harmelen, F. From SHIQ and RDF to OWL: The making of a Web ontology language. *Journal of Web Semantics* 1, (2003), 7–26.
5. Meusel, R., Petrovski, P., and Bizer, C. The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In P. Mika et al., Eds. *The Semantic Web—ISWC 2014 SE-18* (Vol. 8796, 2014), Springer International Publishing, 277–292; DOI: 10.1007/978-3-319-11964-9_18.
6. Tudorache, T., Nyulas, C., Noy, N., and Musen, M. Using Semantic Web in ICD-11: Three Years Down the Road. In H. Alani, et al., Eds. *The Semantic Web—ISWC 2013 SE-13* (Vol. 8219, 2013); Springer Berlin Heidelberg, 195–211; DOI: 10.1007/978-3-642-41338-4_13.

Abraham Bernstein (bernstein@ifi.uzh.ch) is a professor of Informatics and the chair of the Department of Informatics at the University of Zurich as well as the vice president of the Semantic Web Science Association (SWSA).

James Hendler (hendler@cs.rpi.edu) is the Tetherless World Professor of Computer, Web and Cognitive Sciences and the director of the Rensselaer Institute for Data Exploration and Applications at Rensselaer Polytechnic Institute as well as a former president of the Semantic Web Science Association (SWSA).

Natalya Noy (noy@google.com) is a staff scientist at Google Research and the president of the Semantic Web Science Association (SWSA).

Copyright held by authors.

^f <http://bit.ly/1fCLW4d>

 Article development led by queue.acm.org

**Expert-curated guides to
the best of CS research.**

BY PETER BAILIS, SIMON PETER, JUSTINE SHERRY

Introducing Research for Practice

READING A GREAT research paper is a joy. A team of experts deftly guides you, the reader, through the often complicated research landscape, noting the prior art, the current trends, the pressing issues at hand—and then, sometimes artfully, sometimes through seeming sheer force of will, expands the body of knowledge in a fell swoop of 12 or so pages of prose. A great paper contains a puzzle and a solution; these can be useful, enlightening, or both. A great paper is a small, structured quantum of human ingenuity, creativity, and labor, in service of a growing understanding of our world and the future worlds we may inhabit.

Unfortunately, information overload is a defining problem of our time, and computer science research is no exception. The volume of research produced each year in computer science is heartening, but it can be difficult to determine which papers are most deserving of our scarce time. This volume of papers is also at odds with many of the best elements of paper

can be difficult to determine which papers are most deserving of our scarce time. This volume of papers is also at odds with many of the best elements of paper reading: distillation of work to its critical essence, thoughtful consideration of its nuances and the context in which the research was performed, and application of concepts to one's own technical problems and experiences.

As a result, the past few years have seen a rise in interest and organizations—such as Papers We Love and its many chapters—devoted to the joy and utility of reading computer science research: curated-paper discussions have escaped the traditionally academic “reading seminar” format and have been supplanted by groups of hundreds of participants meeting regularly, at startups and community centers, to discuss the latest and greatest computer science research. This is exciting. Why should the greatest of papers be enjoyed only in academia? As a public good, research should be read, discussed, digested, and enjoyed by all interested parties.

ACM has a particularly important role to play in this democratization of access to research. First, the ACM Digital Library is the largest collection of computer science research in the world, with hundreds of thousands of papers, articles, and manuscripts. Second, the ACM membership consists of world experts across all subfields of computer science, from Turing laureates to ACM Fellows, from upstart academics to engineers on the cutting edge of practice. Separately, these are unparalleled resources; put together, they are even more extraordinary.

Research for Practice (RfP) is born from the potential of this combination. In every RfP column, two experts will introduce a short, curated selection of papers on a concentrated, practically oriented topic. Want to learn about the latest and greatest developments in operating systems for datacenter workloads? RfP will pro-

vide an essential crash course from a world authority by describing the trends in this space, selecting a handful of papers to read, and providing motivation and the critical insights behind each.

This approach is designed to allow you to become fluent in exciting topics in computer science research in a weekend afternoon. In addition, ACM has graciously agreed to provide open access to any RfP paper citations available in the ACM Digital Library. Each installment will cover different topics from different volunteer experts, and we intend to cover the entire range of computer science subfields.

With this issue we present the first installment of Research for Practice. Were you curious about the datacenter operating system trends I just mentioned? You're in luck: Simon Peter has a fantastic selection on this topic, including papers on the interplay between emerging I/O subsystems and the kernel, principles for multicore scalability, and systems possibilities for new secure computing hardware. In addition, Justine Sherry has contributed an exciting selection on network functions virtualization: our networks are getting smarter, aided by increasingly complex in-network software. This allows functionality beyond traditional network "middlebox" operation, including complex routing and policy deployment and cryptographically secure and private packet processing. Both of these selections highlight practical yet principled research papers. We are especially pleased by how accessible each of our experts has made these otherwise highly technical topics.

RfP is itself an ongoing experiment. We are inspired by the widespread and growing enthusiasm about computer science research as well as the role ACM, its members, and the ACM readership can play in amplifying this excitement. We welcome your feedback, and please enjoy!

— Peter Bailis

Peter Bailis is assistant professor of computer science at Stanford University. His research in the Future Data Systems group (<http://futuredata.stanford.edu/>) focuses on the design and implementation of next-generation data-intensive systems.



Datacenters are Changing the Way We Design Server Systems By Simon Peter

The growing number of cloud service users and volume of data are putting tremendous pressure on I/O, processing, and integrity. Hardware has kept pace: datacenter networks allow servers to transmit and receive millions of requests per second with microsecond delivery latencies. An increasing number of processors multiplies server-processing capacities, and new technologies such as Intel's Software Guard Extensions (SGX) help keep sensitive data confidential. As a result, operating systems must provide these new technologies to applications scalably and efficiently.

The following papers introduce thought-provoking OS design paradigms that address each of these trends. First, we attack the I/O performance problem. We then introduce a handy software-interface design rule that ensures constructed software can scale with the number of processors present in datacenter servers. Finally, we learn how to protect the integrity of sensitive data, even from access by the cloud operator. We conclude with an outlook on how these paradigms enable an ecosystem of execution environments for datacenter applications.

Dealing with the Data Deluge

Peter, S., et al. 2014. Arrakis: The operating system is the control plane.
Usenix Symposium on Operating Systems Design and Implementation.
<https://www.usenix.org/conference/osdi14/technical-sessions/presentation/peter>

Belay, A., et al. 2014. IX: A protected dataplane operating system for high throughput and low latency.
Usenix Symposium on Operating Systems Design and Implementation.
<https://www.usenix.org/conference/osdi14/technical-sessions/presentation/belay>

These papers discuss the design of operating systems that provide high I/O performance to request-intensive server applications. The authors find the complexity of monolithic OS kernels is the biggest barrier to server I/O performance and remedy the situation by introducing an I/O model that bypasses

the kernel in the common case without losing any of its protection guarantees. Both papers split the OS into a control and a data plane: A kernel-level control plane carries out access control and resource management, while a user-level data plane is responsible for fast I/O mechanisms.

The papers differ in how network I/O policy is enforced. Arrakis reaches for utmost performance by relying on hardware to enforce per-application maximum I/O rates and allowed communication peers. IX trades performance for software control over network I/O, thus allowing the precise enforcement of the I/O behavior of a particular network protocol, such as TCP congestion control.

Both OS models do extremely well supporting an emerging bit of cloud infrastructure: containers. Containers bundle all required components of an application into a manageable unit. Arrakis and IX empower containers to use all I/O capabilities of the underlying server hardware without the overhead of a monolithic OS kernel.

Keeping All Processors Busy

Clements, A. T., et al. 2013. The scalable commutativity rule: Designing scalable software for multicore processors.
ACM Symposium on Operating System Principles.
<http://queue.acm.org/rfp/vol14iss2.html>

Many OS researchers have worked on the problem of using an increasing number of processor cores to handle growing workload demands. Manually identifying and working around scalability bottlenecks caused by shared resource contention in implementations has often been the answer. This paper asks a different question: Can APIs have an impact on software scalability? The surprising answer is that the impact is not only profound, but also fundamental.

The paper distills its insight into a simple yet effective software-development rule: Whenever interface operations commute, they can be implemented in a way that scales. The authors provide a tool that helps developers apply the rule by generating test cases that find scalability bottlenecks in commutative API implementations. They use the tool to evaluate the POSIX

INTERACTIONS



ACM's *Interactions* magazine explores critical relationships between people and technology, showcasing emerging innovations and industry leaders from around the world across important applications of design thinking and the broadening field of interaction design.

Our readers represent a growing community of practice that is of increasing and vital global importance.



To learn more about us, visit our award-winning website <http://interactions.acm.org>

Follow us on Facebook and Twitter  

To subscribe: <http://www.acm.org/subscribe>

Association for
Computing Machinery



API and point out where the API has the ability to scale but its OS implementation hits a bottleneck. They employ the results to develop a new OS that is practically free of scalability bottlenecks.

The scalable commutativity rule applies not just to operating system design, but also to any multicore software system. It should be part of the toolkit of any multicore application developer.

Keeping Sensitive Data Confidential

Baumann, A., et al. 2014.
Shielding applications from an untrusted cloud with Haven.

Usenix Symposium on Operating Systems Design and Implementation.
<https://www.usenix.org/conference/osdi14/technical-sessions/presentation/baumann>

Customers trust their cloud providers not to expose any of their data—a tall order, given the staggering complexity of the cloud hardware/software platform. Bugs may easily compromise sensitive data. This paper introduces Haven, a software system that protects the integrity of a program and its data from the entire cloud-execution platform, except for a small trusted block of firmware.

To achieve this, Haven uses the recently introduced Intel SGX technology to develop a non-hierarchical OS security model that allows applications to run in a secure region of memory that is protected from outside access, including privileged software such as OS kernels and hypervisors. To support execution on top of an untrusted OS kernel, Haven introduces a mutually distrusting kernel interface that applications access via a user-level library that provides the Windows API.

Haven introduces a new way of protecting data confidentiality. While previous attempts use encryption techniques such as homomorphic encryption to compute on encrypted data in limited cases, Haven relies on hardware-protection technology to address the problem in a more general way.

An Ecosystem of Application Execution Environments

These papers establish a new baseline for datacenter OS design. Not the traditional Unix model where processes run on top of a shared kernel invoked via POSIX system calls, but protected

software containers using scalable library invocations that map directly to hardware mechanisms allow applications to break out of existing OS performance and protection limitations.

This new OS design has the potential to enable an ecosystem of library execution environments that support applications in various ways. For example, a fast library network stack may be linked to a Web server to improve its webpage delivery latency and throughput. A Haven-like system call library may be linked to protect the integrity of confidential data held by the application. Finally, a scalable storage stack may be linked to a database to allow it to keep pace with the throughput offered by parallel flash memory. In many cases, these libraries can improve application execution transparently. Together, these new execution environments have the potential to allow applications to match the performance and integrity demands of current and future datacenter workloads.

Simon Peter is an assistant professor at the University of Texas at Austin, where he leads research in operating systems and networks.



NFV and Middleboxes By Justine Sherry

We usually think of networks as performing only one task: Delivering packets from sender to receiver. Today's networks, however, do a lot more by deploying special-purpose *middleboxes* to inspect and transform packets, usually to improve performance or security. A middlebox may scan a connection for malicious behavior, compress data to provide better performance on low-resource mobile devices, or serve content from a cache inside the network to reduce bandwidth costs. Both industry and research sources have recently begun to refer to the features implemented by middleboxes as “network functions.” Popular open source network functions include the Snort Intrusion Detection System³ and the Squid Web Proxy.⁴

To deploy a new network function, a network administrator traditionally purchases a specialized, fixed-function hardware device (the middlebox) implementing, for example, intrusion

detection or caching, and physically installs the device at a chokepoint in the network such that all traffic entering or exiting the network must pass through it. Alternatively, an administrator might use an off-the-shelf server as a middlebox, installing software such as Snort, Squid, or a proprietary software package, and then routing traffic through the server at a chokepoint in the network.

Network functions virtualization (NFV) is a new movement in networking that takes the software-based approach to an extreme. The NFV ISG (industry specification group) envisions a future in which all middlebox functionality is implemented in software.² Network administrators will deploy a server or cluster of servers dedicated to network functions, and network virtualization software will automatically route traffic through various network functions.

NFV promises many benefits for network administrators. It reduces costs by moving from special-purpose to general-purpose hardware, makes upgrades as easy as a software patch, offers the opportunity to scale on demand, and promises more efficient installations with multiple network functions potentially sharing a single server, leaving few resources wasted. NFV has tremendous momentum in the networking community—the NFV working group has more than 200 industrial members¹—but is in its infancy and was founded only in late 2012.

Here we present three highlights from the research community on middleboxes and NFV, and conclude by discussing some of the challenges and opportunities that NFV presents for application developers.

What Capabilities Do Network Functions Implement?

Carpenter, B., Brim, S. 2002. *Middleboxes: taxonomy and issues*. RFC 3234, IETF.
<https://tools.ietf.org/html/rfc3234>

Though it predates NFV by about a decade, this article remains a nice summary of the features for which middleboxes are commonly deployed. The document could have gone into more depth about application-layer behaviors such as ex-filtration detection or intrusion detection—increasingly common in today's

corporate networks—but these behaviors are more common today than they were in 2002 when the article was written. Nonetheless, it remains the most comprehensive survey of middlebox functionality to date, and most of the features it describes remain in common use.

What Does an NFV-Managed Network Look Like?

Palkar, S., Lan, C., et al. 2015. E2: A framework for NFV Applications.
ACM Symposium on Operating Systems Principles.
<http://dl.acm.org/citation.cfm?id=2815423>

This article provides the cleanest vision for an NFV-managed cluster to date. The authors describe a system called E2, which automatically schedules and configures network functions on a cluster of general-purpose servers. E2 allows a network administrator to specify a “configuration” (for example, all traffic on port 80 should be routed through this HTTP proxy, all traffic to this subnet should be processed by an IDS), and the framework will automatically instantiate software instances and a routing configuration to ensure that the policy is met. E2 is conceptually similar to cloud frameworks such as OpenStack or RightScale but in practice involves many different technical challenges, including scheduling to ensure bandwidth is not overutilized, ensuring low latency, and enabling efficient communication and “chaining” between network functions.

Can I Control How Network Functions Process My Traffic?

Naylor, D., et al. 2015. Multi-Context TLS (mcTLS): Enabling secure in-network functionality in TLS.
ACM SIGCOMM.
<http://queue.acm.org/rfp/vol14iss2.html>

Sherry, J., et al. 2015. BlindBox: Deep packet inspection over encrypted traffic. *ACM SIGCOMM.*
<http://queue.acm.org/rfp/vol14iss2.html>

Today, application developers have no way of controlling which network functions process their traffic, short of making a phone call to their network administrators. Nonetheless, developers may have concerns about inspection or modification of traffic sent by their applications—especially with regard to

privacy. Hence, many developers choose to encrypt their entire connection (for example, using SSL/TLS). While this preserves privacy, it also prevents *all* benefits of middlebox processing. These two articles propose new cryptographic protocols—mcTLS and BlindBox—that would let application developers allow certain middlebox operations but restrict others. The two articles propose very different approaches to the same problem and are worth reading side by side.

What Does NFV Mean for Application Developers?

As NFV makes the deployment and configuration of network functions/middleboxes easier, application developers can expect to see increasingly complex behavior from their networks. While this capability for complex behavior retains some of the old challenges of middleboxes (for example, privacy), it also introduces a huge new opportunity for application developers. NFV enables application developers to run and execute their code not only on end hosts they maintain, but also in the network itself.

For example, a developer who designs a custom load-balancing filter based on a unique service architecture might write the new code to run on the load balancer itself. A Web service may implement a custom cache to serve encrypted content to its users, deploying the in-network cache within its customers' ISPs within virtual machines hosted in the provider's infrastructure. With the ability to execute *arbitrary* code in the network—and smart routing and scheduling to ensure the right traffic receives such processing—NFV opens an entirely new programming platform for developers. The next big app store may be for features deployed within datacenter networks, ISPs, or even on home routers. 

References

1. NFV ISG. List of members; <https://portal.etsi.org/TBSiteMap/NFV/NFVMembership.aspx>.
2. NFV ISG. 2012. Network functions virtualization: An introduction, benefits, enablers, challenges, and call for action; https://portal.etsi.org/NFV/NFV_White_Paper.pdf.
3. Snort; <https://www.snort.org/>.
4. Squid; <http://www.squid-cache.org/>.

Justine Sherry is an assistant professor at Carnegie Mellon University, Pittsburgh, PA, starting in the Fall of 2017.

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

**When people don't work well together
they make bad decisions.**

BY KATE MATSUDAIRA

Bad Software Architecture Is a People Problem

IT ALL STARTED with a bug.

Customers were complaining their information was out of date on the website. They would make an update and for some reason their changes were not being reflected. Caching seemed like the obvious problem, but once we started diving into the details, we realized it was a much bigger issue.

What we discovered was the back-end team managing the APIs and data didn't see eye-to-eye with the front-end team consuming the data. The back-end team designed the APIs the way they thought the data should be queried—one that was optimized for the way they had designed the schema. The challenge was that when the front-end team wrote the interface, the API seemed clunky to them—there were too many parameters, and they had to make too many calls. This negatively

impacted the mobile experience where browsers can't handle as many concurrent requests, so the front-end team made the decision to cache part of the data locally.

The crux of the issue was the teams had not communicated well with each other. Neither team had taken the time to understand the needs of the other team. The result was a weird caching bug that affected the end user.

You might be thinking this could never happen on your team, but the reality is that when many different people are working on a problem, each could have a different idea about the best solution. And when you don't have a team that works well together, it can hurt your software design, along with its maintainability, scalability, and performance.

Most software systems consist of parts and pieces that come together to perform a larger function. Those parts and pieces can be thought out and planned, and work together in a beautiful orchestra. Or they can be designed by individuals, each one as unique as the person who created it. The challenge is if you want your software to last, uniformity and predictability are good things—unique snowflakes are not.

One of the challenges of managing a software team is balancing the knowledge levels across your staff. In an ideal world, every employee would know enough to do his or her job well, but the truth is in larger software teams there is always someone getting up to speed on something: a new technology, a way of building software, or even the way your systems work. When someone doesn't know something well enough to do a great job, there is a knowledge gap, and this is pretty common.

When building software and moving fast, people don't always have enough time to learn everything they need to bridge their gaps. So each person will make assumptions or concessions that can impact the effectiveness of any software that individual works on.

For example, an employee may choose a new technology that hasn't been road tested enough in the wild, and later that technology falls apart under heavy production load. Another example is someone writing code for a particular function, without knowing that code already exists in a shared library written by another team—reinventing the wheel and making maintenance and updates more challenging in the future.

On larger teams, one of the common places these knowledge gaps exist is between teams or across disciplines: for example, when someone in operations creates a Band-Aid in one area of the system (like repetitively restarting a service to fix a memory leak), because the underlying issue is just too complex to diagnose and fix (the person doesn't have enough understanding of the running code to fix the leaky resources).

Everyday, people are making decisions with imperfect knowledge. The real question is, how can you improve the knowledge gaps and leverage your team to make better decisions?

Here are a few strategies that can help your team work better, and in turn help you create better software. While none of these strategies is a new idea, they are all great reminders of ways to make your teams and processes that much better.

Define how you will work together. Whether you are creating an API or consuming someone else's data, having a clearly defined contract is the first step toward a good working relationship. When you work with another service it is important to understand the guardrails and best practices for consuming that service. For example, you should establish the payload maximums and discuss the frequency and usage guidelines. If for some reason the existing API doesn't meet your needs, then instead of just working around it, talk about why it isn't working and collaboratively figure out the best way to solve the problem (whether it would be updating the API or leveraging a caching strategy). The key here is communication.

Decide how you will test the whole system. One of the most important strategies is to think about how you will truly test the end-to-end functionality of a system. Having tests that investigate only *your* parts of the system (like the back-end APIs) but not the end-customer experience can result in uncaught errors or issues (such as my opening example of caching). The challenge then becomes, who will own these tests? And who will run these tests and be responsible for handling failures? You may not want tests for every scenario, but certainly the most important ones are worth having.



When bugs happen, work together to solve them. When problems arise, try to avoid solutions that only mask the underlying issue. Instead, work together to figure out what the real cause of the problem is, and then make a decision as a team on the best way of addressing it going forward. This way the entire team can learn more about how the systems work, and everyone involved will be informed of any potential Band-Aids.

Use versioning. When another team consumes something you created (an API, a library, a package), versioning is the smartest way of making updates and keeping everyone on the same page with those changes. There is nothing worse than relying on something and having it change underneath you. The author may think the changes are minor or innocuous, but sometimes

those changes can have unintended consequences upstream. By starting with versions, it is easy to keep everyone in check and predictably manage their dependencies.

Create coding standards. Following standards can be really helpful when it comes to code maintenance. When you depend on someone else and have access to that source code, being able to look at it—and know what you are looking at—can give you an edge in understanding, debugging, and integration. Similarly, in situations where styles are inherited and reused throughout the code, having tools like a style guide can help ensure that the user interfaces look consistent—even when different teams throughout the company develop them.

Do code reviews. One of the best ways of bridging knowledge gaps on a team is to encourage sharing among team members. When other members review and give feedback, they learn the code, too. This is a great way of spreading knowledge across the team.

Of course, the real key to great software architecture for a system developed by lots of different people is to have great communication. You want everyone to talk openly to everyone else, ask questions, and share ideas. This means creating a culture where people are open and have a sense of ownership—even for parts of the system they didn't write. 

Related articles on queue.acm.org

Hitchhiker's Guide to Biomorph Software

Kenneth N. Lodding

<http://queue.acm.org/detail.cfm?id=1016985>

Ground Control to Architect Tom...

Alex Bell

<http://queue.acm.org/detail.cfm?id=1317396>

Broken Builds

Kode Vicious

<http://queue.acm.org/detail.cfm?id=1740550>

Kate Matsudaira (katemats.com) is the founder of her own company, Popforms. Previously she worked in engineering leadership roles at companies like Decide (acquired by eBay), Moz, Microsoft, and Amazon.

© 2016 ACM 0001-0782/16/09 \$15.00

The operations side of the story.

BY THOMAS A. LIMONCELLI

10 Optimizations on Linear Search

A FRIEND WAS asked the following question during a job interview: What is the fastest algorithm to find the largest number in an unsorted array?

The catch, of course, is that the data is unsorted. Because of that, each item must be examined; thus, the best algorithm would require $O(N)$ comparisons, where N is the number of elements. Any computer scientist knows this. For that reason, the fastest algorithm will be a linear search through the list.

End of story.

All the computer scientists may leave the room now.

Are all the computer scientists gone?

Good!

Now let's talk about the operational answer to this question.

System administrators (DevOps engineers or SREs or whatever your title) must deal with the operational aspects of computation, not just the theoretical aspects. Operations is where the rubber hits the road. As a result, operations people see things from a different perspective and can realize opportunities outside of the basic $O()$ analysis.

Let's look at the operational aspects of the problem of trying to improve something that is theoretically optimal already.

1. Don't Optimize Code that is Fast Enough

The first optimization comes from deciding to optimize time and not the algorithm itself. First, ask whether the code is fast enough already. If it is, you can optimize your time by not optimizing this code at all. This requires a definition of *fast enough*.

Suppose 200ms and under is *fast enough*. Anything that takes less than 200ms is perceived to be instantaneous by the human brain. Therefore, any algorithm that can complete the task in less than 200ms is usually good enough for interactive software.

Donald Knuth famously wrote that premature optimization is the root of all evil. Optimized solutions are usually more complex than the solutions they replace; therefore, you risk introducing bugs into the system. A bird in hand is worth two in the bush. Why add complexity when you don't have to?

My biggest concern with premature optimization is it is a distraction from other, more important work. Your time is precious and finite. Time spent on a premature optimization is time that could be spent on more important work.

Prioritizing your work is not about deciding in what order you will do the items on your to-do list. Rather, it is deciding which items on your to-do list will be intentionally dropped on the floor. I have 100 things I would like to do this week. I am going to complete only about 10 of them. How I prioritize my work determines which 90 tasks won't get done. I repeat this process ev-



ery week. One of the best time-management skills you can develop is to learn to let go of that 90%.

In the case of the interview question, whether optimizing is worthwhile relates to the number of data items. It isn't worth optimizing if only a small amount of data is involved. I imagine that if, during the interview, my friend had asked, "How many elements in the list?" the interviewer would have told him that it doesn't matter. From a theoretical point of view, it doesn't; from

an operational point of view, however, it makes *all* the difference.

Deciding if an optimization is worth your time requires a quick back-of-the-envelope estimate to determine what kinds of improvements are possible, how long they might take to be achieved, and if the optimization will result in a return on investment. The ability to use rough estimates to decide whether or not an engineering task is worthwhile may be one of the most important tools in a system administrator's toolbox.

If *small* is defined to mean any amount of data that can be processed in under 200ms, then you would be surprised at how big *small* can be.

I conducted some simple benchmarks in Go to find how much data can be processed in 200ms. A linear search can scan 13 million elements in less than 200ms on a three-year-old MacBook laptop, and 13 million is no small feat.

This linear search might be buggy, however. It is five lines long and, not to

brag, but I can pack a lot of bugs into five lines. What if I were to leverage code that has been heavily tested instead? Most languages have a built-in sort function that has been tested far more than any code I've ever written. I could find the max by sorting the list and picking the last element. That would be lazy and execute more slowly than a linear search, but it would be very reliable. A few simple benchmarks found that on the same old laptop, this “lazy algorithm” could sort 700,000 elements and still be under the 200ms mark.

What about smaller values of N ?

If $N = 16,000$, then the entire dataset fits in the L1 cache of the CPU, assuming the CPU was made in this decade. This means the CPU can scan the data so fast it will make your hair flip. If $N = 64,000$, then the data will fit in a modern L2 cache, and your hair may still do interesting things. If the computer wasn't made in this decade, I would recommend that my friend reconsider working for this company.

If N is less than 100, then the lazy algorithm runs imperceptibly fast. In fact, you could repeat the search on demand rather than storing the value, and unless you were running the algorithm thousands of times, the perceived time would be negligible.

The algorithms mentioned so far are satisfactory until $N = 700,000$ if we are lazy and $N = 13,000,000$ if we aren't; 13 million 32-bit integers (about 52MB) is hardly small by some standards. Yet, in terms of human perception, it can be searched instantly.

If my friend had known these benchmark numbers, he could have had some fun during the interview, asking the interviewer to suggest a large value of N , and replying, “What? I don't get out of bed for less than 13 million integers!” (Of course, this would probably have cost him the job.)

2. Use SIMD Instructions

Most modern CPUs have SIMD (single instruction, multiple data) instructions that let you repeat the same operation over a large swath of memory. They are able to do this very quickly because they benefit from more efficient memory access and parallel operations.

According to one simple benchmark (<http://stackoverflow.com/a/2743040/71978>), a 2.67GHz Core i7 saw a 7–8x improvement by using SIMD instructions where $N = 100,000$. If the amount of data exceeded the CPU's cache size, the benefit dropped to 3.5x.

With SIMD, *small* becomes about 45 million elements, or about 180MB.

3. Work in Parallel

Even if N is larger than the *small* quantity, you can keep within your 200ms time budget by using multiple CPUs. Each CPU core can search a shard of the data. With four CPU cores, *small* becomes $4N$, or nearly 200 million items.

When I was in college, the study of parallel programming was hypothetical because we didn't have access to computers with more than one CPU. In fact, I didn't think I would ever be lucky enough to access a machine with such a fancy architecture. Boy, was I wrong! Now I have a phone with eight CPU cores, one of which, I believe, is dedicated exclusively to crushing candy.

Parallel processing is now the norm, not the exception. Code should be written to take advantage of this.

4. Hide Calculation in Another Function

The search for the max value can be hidden in other work. For example, earlier in the process the data is loaded into memory. Why not have that code also track the max value as it iterates through the data? If the data is being loaded from disk, the time spent waiting for I/O will dominate, and the additional comparison will be, essentially, free.

If the data is being read from a text file, the work to convert ASCII digits to 32-bit integers is considerably more than tracking the largest value seen so far. Adding max-value tracking would be “error in the noise” of any benchmarks. Therefore, it is essentially free.

You might point out that this violates the SoC (separation of concerns) principle. The method that loads data from the file should just load data from a file. Nothing else. Having it also track the maximum value along the way adds complexity. True, but we've already decided the added complexity is worth the benefit.

Where will this end? If the `LoadDataFromFile()` method also calculates the max value, what's to stop

us from adding other calculations? Should it also calculate the min, count, total, and average? Obviously not. If you have the count and total, then you can calculate the average yourself.

5. Maintain the Max Along the Way

What if the max value cannot be tracked as part of loading the dataset? Perhaps you don't control the method that loads the data. If you are using an off-the-shelf JSON (JavaScript Object Notation) parser, adding the ability to track the max value would be very difficult. Perhaps the data is modified after being loaded, or it is generated in place.

In such situations I would ask why the data structure holding the data isn't doing the tracking itself. If data is only added, never removed or changed, the data structure can easily track the largest value seen so far. The need for a linear search has been avoided altogether.

If items are being removed and changed, more sophisticated data structures are required. A heap makes the highest value accessible in $O(1)$ time. The data can be kept in the original order but in a heap or other index on the side. You will then always have fast access to the highest value, though you will suffer from additional overhead maintaining the indexes.

6. Hide Long Calculations from Users

Maybe the process can't be made any faster, but the delay can be hidden from the user.

One good place to hide the calculation is when waiting for user input. You don't need the entire processing power of the computer to ask “Are you sure?” and then wait for a response. Instead, you can use that time to perform calculations, and no one will be the wiser.

One video-game console manufacturer requires games to have some kind of user interaction within a few seconds of starting. Sadly, most games need more time than that to load and initialize. To meet the vendor's requirement, most games first load and display a title screen, then ask users to click a button to start the game. What users don't realize is that while they are sitting in awe of the amazing title screen, the game is finishing its preparations.

Get Out of Your Silo

Before discussing the remaining optimizations, let's discuss the value of thinking more globally about the problem. Many optimizations come from end-to-end thinking. Rather than optimizing the code itself, we should look at the entire system for inspiration.

To do this requires something scary: talking to people. Now, I understand that a lot of us go into this business because we like machines more than people, but the reality is that operations is a team sport.

Sadly, often the operations team is put in a silo, expected to work issues out on their own without the benefit of talking to the people who created the system. This stems from the days when one company created software and sold it on floppy disks. The operations people were in a different silo from the developers because they were literally in a different company. System administrators' only access to developers at the other company was through customer support, whose job it was to insulate developers from talking to customers directly. If that ever did happen, it was called an escalation, an industry term that means that a customer accidentally got the support he or she paid for. It is something that the software industry tries to prevent at all costs.

Most (or at least a growing proportion of) IT operations, however, deal with software that is developed in-house. In that situation there is very little excuse to have developers and operations in separate silos. In fact, they should talk to each other and collaborate. There should be a name for this kind of collaboration between developers and operations ... and there is: DevOps.

If your developers and operations teams are still siloed away from each other, then your business model hasn't changed since software was sold on floppy disks. This is ironic since your company probably didn't exist when floppy disks were in use. What's wrong with this picture?

Get out of your silo and talk to people. Take a walk down the hallway and introduce yourself to the developers in your company. Have lunch with them. Indulge in your favorite after-work beverage together. If you are a manager who requires operations and developers to communicate only through



Many optimizations come from end-to-end thinking. Rather than optimizing the code itself, we should look at the entire system for inspiration.



“proper channels” involving committees and product management chains, get out of their way.

Once operations has forged a relationship with developers, it is easier to ask important questions, such as: How is the data used? What is it needed for and why?

This kind of social collaboration is required to develop the end-to-end thinking that makes it possible to optimize code, processes, and organizations. Every system has a bottleneck. If you optimize upstream of the bottleneck, you are simply increasing the size of the backlog waiting at the bottleneck. If you optimize downstream of the bottleneck, you are adding capacity to part of a system that is starved for work. If you stay within your silo, you'll never know enough to identify the actual bottleneck.

Getting out of your silo opens the door to optimizations such as our last four examples.

7. Use a “Good Enough” Value Instead

Is the maximum value specifically needed, or is an estimate good enough?

Perhaps the calculation can be avoided entirely.

Often an estimate is sufficient, and there are many creative ways to calculate one. Perhaps the max value from the previous dataset is good enough.

Perhaps the max value is being used to preallocate memory or other resources. Does this process really need to be fine-tuned every time the program runs? Might it be sufficient to adjust the allocations only occasionally—perhaps in response to resource monitoring or performance statistics?

If you are dealing with a *small* amount of data (using the earlier definition of small), perhaps preallocating resources is overkill. If you are dealing with large amounts of data, perhaps preallocating resources is unsustainable and needs to be reengineered before it becomes dangerous.

8. Seek Inspiration from the Upstream Processes

Sometimes we can get a different perspective by examining the inputs.

Where is the data coming from?

I once observed a situation where a developer was complaining that an

operation was very slow. His solution was to demand a faster machine. The sysadmin who investigated the issue found that the code was downloading millions of data points from a database on another continent. The network between the two hosts was very slow. A faster computer would not improve performance.

The solution, however, was not to build a faster network, either. Instead, we moved the calculation to be closer to the data. Rather than download the data and do the calculation, the sysadmin recommended changing the SQL query to perform the calculation at the database server. Instead of downloading millions of data points, now we were downloading the single answer.

This solution seems obvious but eluded the otherwise smart developer. How did that happen? Originally, the data was downloaded because it was processed and manipulated many different ways for many different purposes. Over time, however, these other purposes were eliminated until only one purpose remained. In this case the issue was not calculating the max value, but simply counting the number of data points, which SQL is very good at doing for you.

9. Seek Inspiration from The Downstream Processes

Another solution is to look at what is done with the data later in the process. Does some other processing step sort the data? If so, the max value doesn't need to be calculated. You can simply sort the data earlier in the process and take the last value.

You wouldn't know this was possible unless you took the time to talk with people and understand the end-to-end flow of the system.

Once I was on a project where data flowed through five different stages, controlled by five different teams. Each stage took the original data and sorted it. The data didn't change between stages, but each team made a private copy of the entire dataset so they could sort it. Because they had not looked outside their silos, they didn't realize how much wasted effort this entailed.

By sorting the data earlier in the flow, the entire process became much faster. One sort is faster than five.

10. Question the Question

When preparing this column I walked around the New York office of Stack Overflow and asked my coworkers if they had ever been in a situation where calculating the max value was a bottleneck worth optimizing.

The answer I got was a resounding no.

One developer pointed out that calculating the max is usually something done infrequently, often once per program run. Optimization effort should be spent on tasks done many times.

A developer with a statistics background stated that the max is useless. For most datasets it is an outlier and should be ignored. What *are* useful to him are the top N items, which presents an entirely different algorithmic challenge.

Another developer pointed out that anyone dealing with large amounts of data usually stores it in a database, and databases can find the max value very efficiently. In fact, he asserted, maintaining such data in a homegrown system is a waste of effort at best and negligent at worst. Thinking you can maintain a large dataset safely with homegrown databases is hubris.

Most database systems can determine the max value very quickly because of the indexes they maintain. If the system cannot, it isn't the system administrator's responsibility to rewrite the database software, but to understand the situation well enough to facilitate a discussion among the developers, vendors, and whoever else is required to find a better solution.

Conclusion: Find Another Question

This brings me to my final point. Maybe the interview question posed at the beginning of this column should be retired. It might be a good logic problem for a beginning programmer, but it is not a good question to use when interviewing system administrators because it is not a realistic situation.

A better question would be to ask job candidates to describe a situation where they optimized an algorithm. You can then listen to their story for signs of operational brilliance.

I would like to know that the candidates determined ahead of time what would be considered good enough. Did they talk with stakeholders to

determine whether the improvement was needed, how much improvement was needed, and how they would know if the optimization was achieved? Did they determine how much time and money were worth expending on the optimization? Optimizations that require an infinite budget are not nearly as useful as one would think.

I would look to see if they benchmarked the system before and after, not just one or the other or not at all. I would like to see that they identified a specific problem, rather than just randomly tuning parts until they got better results. I would like to see that they determined the theoretical optimum as a yardstick against which all results were measured.

I would pay careful attention to the size of the improvement. Was the improvement measured, or did it simply "feel faster?" Did the candidates enhance performance greatly or just squeeze a few additional percentage points out of the existing system? I would be impressed if they researched academic papers to find better algorithms.

I would be most impressed, however, if they looked at the bigger picture and found a way to avoid doing the calculation entirely. In operations, often the best improvements come not from adding complexity, but by eliminating processes altogether. 

Related articles on queue.acm.org

You're Doing It Wrong

Poul-Henning Kamp

<http://queue.acm.org/detail.cfm?id=1814327>

You Don't Know Jack about Network Performance

Kevin Fall and Steve McCanne

<http://queue.acm.org/detail.cfm?id=1066069>

Why Writing Your Own Search Engine is Hard

Anna Patterson

<http://queue.acm.org/detail.cfm?id=988407>

Thomas A. Limoncelli is a site reliability engineer at Stack Overflow Inc. in New York City. His books include *The Practice of Cloud Administration* (<http://the-cloud-book.com>), *The Practice of System and Network Administration* (<http://the-sysadmin-book.com>), and *Time Management for System Administrators*. He blogs at EverythingSysadmin.com and tweets at @YesThatTom.

Copyright held by author.
Publication rights licensed to ACM. \$15.00

Computing Reviews

Connect with our Community of Reviewers

“There aren’t too many walk-in technical bookshops where I live; checking out reviewer comments on new books provides me with an equivalent experience.”

- Graham Jenkins



Association for
Computing Machinery

ThinkLoud

www.computingreviews.com

DOI:10.1145/2893181

Using database views to define citable units is the key to specifying and generating citations to data.

BY PETER BUNEMAN, SUSAN DAVIDSON, AND JAMES FREW

Why Data Citation Is a Computational Problem

CITATION IS ESSENTIAL to traditional scholarship. Citations identify the cited material, help retrieve it, give credit to its creator, date it, and so on. In the context of printed materials (such as books and journals), citation is well understood. However, the world is now digital. Most scholarly and scientific resources are held online, and many are in some kind of database, or a structured, evolving collection of data. For example, most biological reference works have been replaced by curated databases, and vast amounts of basic scientific data—geospatial, astronomical, molecular, and more—are now available online. There is strong demand^{13,23} that these databases should be given the same scholarly status and appropriately cited, but how can this be done effectively?

Database citation is a challenge due to the structure and evolution of databases. Attributes such as ownership

and authorship may change for different parts of the database. Even for a simple collection of files, good methods may be needed for citing subsets of these files; that is, we want to do better than cite the whole collection or generate a huge number of citations to individual files.

A citation is a collection of “snippets” of information (such as authorship, title, ownership, and date) that are specified by the database administrators and that may be prescribed by some standard. However, if we expect people to cite digital data, simply providing principles and standards for citation is not enough; we must also generate the citations. Even when making conventional citations to the literature, authors try to avoid typing in citations. Instead, they look for the citation in some database of citations, such as the ACM Digital Library^a and DBLP,^b and insert it into their document using a reference manager, such as BibTeX, Mendeley, and Zotero, or by copy-paste. In the context of citing databases, if the citation is not available or if the standard appears complicated, an author may well omit the citation or provide an inaccurate one. In short, unless citations are generated along with answers to data, the advocacy of data citation will have limited effect.

How can citations be automatically generated for data extracted from a

a <http://dl.acm.org/>

b <http://dblp.uni-trier.de/>

» key insights

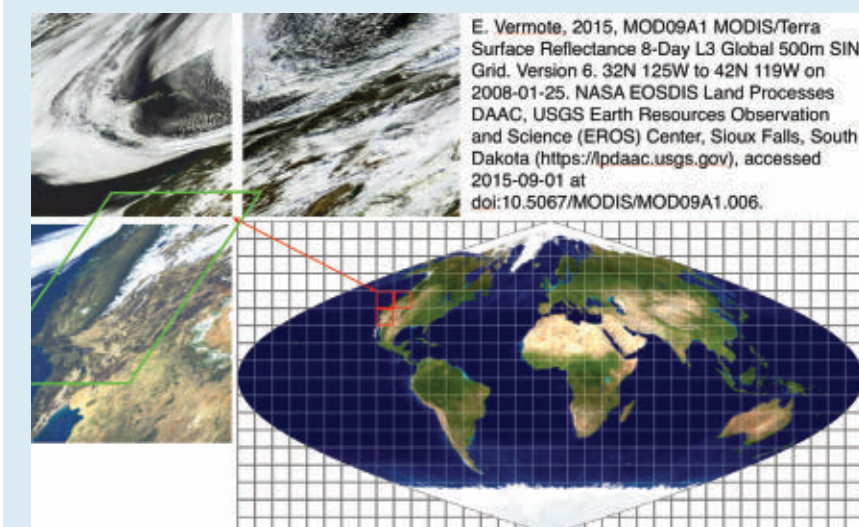
- **Most information is now published in evolving databases or datasets; even traditional reference works are published as curated databases.**
- **Data citation depends on both the query and the data; since there is an unlimited number of queries, views can be used to specify citable units.**
- **A query may be supported by several different views; the choice of what citation to use for the query can be simplified when the views form a hierarchy.**



Figure 1. GtoPdb family and introductory pages with independent citations.



Figure 2. The MODIS grid, with highlighted tiles (red) of spatial extent for California (green), with citation.



database? Here, we use the term “database” in a broad sense and “query” to mean any mechanism used to extract the data, such as a set of file names, an SQL query, a URL, or a special-purpose GUI. The computational problem this poses can be broadly and simply formulated as:

Given a database D and a query Q , generate an appropriate citation.

It is often the case that the curators,

authors, or publishers of a database have good ideas about how their data should be cited. However, it is unlikely that they will know how to associate a citation with some complex SQL query, and even less likely that the user of the data, whose query was generated by some user interface, will understand what is wanted. In order to extract the citation *automatically* from the query Q and the database D , two questions

need to be answered:

- Does the citation depend on both Q and D or just on the data $Q(D)$ extracted by Q from D ?

- If we have appropriate citations for some queries, can we use them to construct citations for other queries?

If the retrieved data is simply a number or an image, one cannot expect to find the citation in the retrieved data. Moreover, even if the query returns nothing, it may be worthy of citation, but what citation is associated with the empty set? We need at least context information; so we need both Q and D .

The answer to the second question is important because authors and publishers frequently have ideas as to how to cite certain parts of the database; that is, they can provide citations for certain queries but do not know what to do about other queries.

Numerous organizations^{2,6,12,16} have advocated data citation and developed principles^{2-4,7,8,12,13,15} that refine and standardize the notion.^{1,3,4,8,9,18} The purpose of these standards is mostly to prescribe the information in a citation—the snippets—and also to define its structure.

A major, but not the only, purpose of a citation is to identify the cited material, and citation is often linked to

persistent identifiers (such as DOIs,^c ARKs,^d and URIs^e). These identifiers, while they may have certain fixed properties, do not guarantee that the cited material remains unchanged, a property known as “fixity.” Beyond observing that citations should reference the appropriate version, we do not address fixity in this article, nor do we address the closely related topic of provenance, which involves a record of the whole process of data extraction. For a discussion of these issues and a prototype system that combines citation and provenance, see Pröll and Rauber.^{21,22}

In this article, we propose a general approach to citation generation, and illustrate it using two scientific databases that are radically different in both their structure and how they should be cited.

Sample scientific datasets. We now describe these two databases. One is a curated relational database that is widely used in pharmacology; the other is a collection of files in a scientific data format that supports research in Earth sciences.

GtoPdb. The IUPHAR/BPS Guide to PHARMACOLOGY (GtoPdb)^{20,f} is a relational database that contains expertly curated information about drugs in clinical use and some experimental drugs, together with information on the cellular targets of the drugs and their mechanisms of action in the body. This resource is particularly useful to researchers who hypothesize that a particular cellular mechanism is involved in a physiological process of interest and want to find tools (drugs) to impose a specific activation level on the pathway to test their hypotheses.

Users view information through a hierarchy of webpages. The top level divides information by “families” of drug targets that reflect typical pharmacological thinking; lower levels divide the families hierarchically into subfamilies and so on down to individual drug targets and drugs. At the lowest level are expert-created overviews and, for some entries, pages containing details of chemical and genetic structures and properties. Despite its underlying relational imple-



Unless citations are generated along with answers to data, the advocacy of data citation will have limited effect.



mentation, GtoPdb can therefore be thought of as a structured hierarchy.

Information in GtoPdb is generated by hundreds of expert contributors, and different database entries are associated with different lists of contributors. While the suggested citation for GtoPdb as a whole (the root) is a traditional journal article written by its curators, a citation to a subtree of GtoPdb includes the contributors who generated the content (see Figure 1). The citation may also depend on the path to the subtree (the query), as a few targets are members of more than one family, and the classification of the target is part of the citation. Queries against GtoPdb may return a Boolean value or the empty set, and to cite this fact, to, say, determine the relevant contributors, one clearly needs the query. A useful property of GtoPdb is that nearly all the information needed to construct a citation (such as names of contributors) is in the database itself.

MODIS. MODERate-resolution Imaging Spectrometer (MODIS)²⁴ is an optical imaging system currently flying aboard NASA's Terra and Aqua satellites. Each MODIS sensor images the entire surface of the Earth every one to two days as a strip approximately 2,000km wide beneath the satellite's orbit. The MODIS sensor records the top-of-atmosphere radiance in several spectral bands, but MODIS data products typically process these values into Earth surface properties (such as reflectance, snow cover, ocean, and color).

MODIS data products are distributed as granules—fixed-size subsets representing either an interval (typically five minutes) of the satellite's orbit or a tile within a standard map projection of all or part of the Earth (see Figure 2). Each MODIS granule is created, stored, and distributed as a Hierarchical Data Format file. MODIS data product search and access systems typically identify and return entire granules, not subsets of granules.

Each MODIS data product defines a granule naming convention, typically incorporating the product identifier, a version number, date-times of acquisition and generation, and, if applicable, a tile identifier. A granule name is thus a unique identifier for the granule but is not itself a complete citation for two reasons. First, applications of MODIS data products frequently use multiple gran-

c <http://dx.doi.org/10.1000/182>

d <http://confluence.ucop.edu/display/Curation/ARK>

e <http://www.ietf.org/rfc/rfc3986>

f <http://www.guidetopharmacology.org/>

ules, and there is no standard way to refer to a set of granules other than by complete enumeration. Second, applications of MODIS data products frequently focus on spatiotemporal regions of interest that are not precisely aligned with granule boundaries; an application's query against a MODIS data product may thus not be precisely reflected in the corresponding set of product granules. For example, compare the latitude-longitude bounding box for California in Figure 2 with the non-rectangular set of MODIS tiles that intersect the box. While enumerating this set is important for provenance, a spatiotemporal bounding box is a compact description of the coverage that—if expressed in a common coordinate system—allows easy searching for studies relevant to a particular region. Such bounding boxes are a common feature of geospatial citations; a spatial bounding box is indeed one of the optional fields in the DataCite schema.⁹

Toward a Solution

We now address the problem of generating a citation for a query Q on database D . As with both GtoPdb and MODIS, the citation will depend on both Q and D . This dependence would appear to be a major problem, since anything that involves the analysis of a query or program is likely to be computationally expensive, if not undecidable. However, as we will show, the problem may be alleviated if we have a base of citations for certain “views” of D that may then be used to generate citations for other queries. From a practical perspective, it is unlikely that data publishers will be able to associate a citation with an arbitrarily complex query; however, it should be possible for them to say, “For this part of the database, the citation should look like this.” If several “parts of the database” can be formalized as views, then there is a basis for generating citations.

Views and citable units. The standard notion of a database view is: Given a database schema S , a view is some function V , which, when applied to any instance of S (or any database that conforms to S), produces a database in some other schema S' . Note the input and output database schemas do not have to be in the same data model; it is possible to, for example, have an XML view of a relational database. Views have been used in traditional database



A simple rule-based language using XPath-like syntax can be used to produce an appropriate citation when the views form a hierarchy.



architectures to describe “areas of responsibility” for parts of a database. What we propose here is to use views to create “citable units.”^g

Figure 3 is a simplified^h representation of GtoPdb as a hierarchy, which is how it is published as web pages and understood by many contributors and users. The hierarchy contains four different node classes: root, families, introductions (to families), and targets. Each of these nodes defines a view that is the subtree beneath it, and the GtoPdb curators have specified a different citation for each class. The higher levels of the hierarchy have citations with collaborators (editors or curators) and the lower levels with contributors. The curators of GtoPdb would like to carry citations down to the level of tables and tuples, but currently a citation for any other node in the hierarchy is the citation for the nearest ancestor of that node.

This is a promising start for defining citations for the hierarchical (Web) presentation of the database, but recall that the underlying database is relational. How can these ideas be used to provide a citation for some SQL query against the database? We can turn this question into one about views. Suppose we are given a database schema S , a view V over S , and a query Q . If Q can be expressed as a query over V , then the citation associated with V is a candidate citation for Q . More formally, if there is a query Q' such that, for all instances D of S , $Q(D) = Q'(V(D))$, then the citation for V is a candidate citation for Q .

The view (the subtree) for each node in the hierarchy is given by a simple query on the underlying database. For example, there is a TARGET table in which the primary key is a target identifier TID. For any value x of TID, and for any table that has TID as a foreign key, we select the rows that contain x . We now get a set of tables, each of which is a subset of the rows of the table in the original database. This is a view defined by x , and each such value of TID defines a distinct “target” view. A similar construction works for families; there is a

^g The online appendix includes a discussion on citable units and recommended reading on database views.

^h To simplify our presentation, we assume families are all directly under the root in GtoPdb; however, in reality, some families may be grouped together as subfamilies of another family.

Figure 4. A citation rule and sample result for GtoPdb.

```
{ Title: "IUPHAR/BPS Guide to Pharmacology", Version: $v,
  Family: $$f, Contributors: $a, URI: "www.iuphar.org" }
←
/Root[VersionNumber: $v]/Family[FamilyName: $$f]/Introduction[Contributor-list: $a]

{ Title: "IUPHAR/BPS Guide to Pharmacology", Version: 26, Family: "Calcitonin",
  Contributors: ["Debbie Hay", "David R. Poyner"], URI: "www.iuphar.org" }
```

Figure 5. A citation rule and sample result for MODIS.

```
{ author: m_auth($p,$$v), m_year:($p,$$v), title: m_title($p), version: $v,
  bounding-box : [$ $minlong, $ $minlat, $ $maxlong, $ $maxlat],
  interval: [$ $mint, $ $maxt], organization: m_org($p), url: m_url($p), accessed: DATE(),
  doi = m_doi($p,$$v) }
←
/root/product[ProdName=$p]/version[vnum=$$v]
  /file[Lat ≥ $ $minlat and Lat ≤ $ $maxlat and
    Lon ≥ $ $minlon and Lon ≤ $ $maxlon and
    Time ≥ $ $mint and Time ≤ $ $maxt]

{ author: "E. Vermote", title: "MOD09A1 ... SIN Grid", version: 6,
  bounding-box: [-125, 32, -119, 42], interval: [2008-01-25, 2008-01-25],
  organization: "NASA EOSDIS ... South Dakota", URL: "https://lpdaac.usgs.gov",
  accessed: "2015-09-01", doi: "10.5067/MODIS/MOD09A1.006" }
```

XPath-like expression that contains two kinds of variables: $$$x$ variables are the view parameters; and $$x$ variables are bound once the $$$x$ variables have been matched. Here, the names of contributors are extracted. They depend on the family and on the version number, which is unique to the database.

The left-hand side of the rule contains the citation in whatever syntax is preferred. Here, we have assumed a simple JSON-style syntax, but the syntax could be in one of the numerous citation “styles” or some more generic syntax (such as BibTeX^j and DataCite⁹). In this example we have assumed the database name and the URI are constants in the citation.

The sample result in Figure 4 is the citation for the simple path

```
/Root/
Family[FamilyName="Calcitonin"]
```

It is also the citation for a simple SQL selection on the Family table with Name = “Calcitonin”. In these cases, it is again easy to determine that the query can be answered using the appropriate relational version of the Family view.

Citations and MODIS. From a database perspective, MODIS is much

simpler than GtoPdb. It is a hierarchically organized collection of products (such as surface reflectance products) consisting of a set of granules we assume for now are tiles, as in Figure 2. A typical retrieval will ask for a set of tiles that cover a certain region of the Earth’s surface and in which the time stamp is within a given interval—a spatiotemporal bounding box of granules. For example, if a researcher was interested in the surface reflectance for California on January 25, 2008, the granules could be specified by a bounding box in which latitude and longitude are the ranges [32,42] degrees and [-125,-119] degrees^k and whose time is 2008-01-25.

The query to retrieve these granules can be expressed as a range query. If we group MODIS products into a hierarchy, our spatiotemporal query may be expressed in a path language as follows:

```
/root/product[ProdName=
"surface reflectance"]/
file[ Lat ≥ 32 and Lat ≤ 42 and
    Lon ≥ -125 and Lon ≤ -119 and
    Time = 2008-01-25 ]
```

This example closely reflects the retrieval capabilities of many MODIS

product-distribution systems. To describe this common bounding box retrieval pattern, an appropriate parameterized view would be:

```
/root/product[ProdName=$$p]/
file[ Lat ≥ $ $minlat and
    Lat ≤ $ $maxlat and
    Lon ≥ $ $minlon and
    Lon ≤ $ $maxlon and
    Time ≥ $ $mint and
    Time ≤ $ $maxt ]
```

GtoPdb and MODIS differ in where they store information needed to construct the citation. In GtoPdb it is in the database, while in MODIS it is mostly kept elsewhere. This is easily solved by having functions in the citation rule that query an appropriate metadata repository with parameters extracted from the matching rule. For example, in Figure 5, `m_auth()` is a function that, given a product and version, queries the metadata for authorship. To our knowledge, there is currently no such organized metadata repository for MODIS, but having one would clearly be beneficial.

The version and access time (`DATE` function) are also not part of the view definition but can be calculated when the query is executed. Note that

^j <http://www.bibtex.org/>

^k This is approximately the green box in Figure 2.

in MODIS, when newer analysis software becomes available, the entire database of products is reanalyzed, yielding a complete new version; old versions are not kept. While this is undesirable from the standpoint of provenance and reproducibility, the citation still carries useful information, even though its referent may not exist.

Conclusion

We have addressed a critical issue in the adoption of data citation—automatically generating a citation from the query and database that was used to obtain the data. A preliminary implementation of the rule-based citation language for hierarchical data is given in Buneman and Silvello.⁵ What we have described here is quite general and applies to any database with a well-defined query language. Rewriting queries through views was originally developed for query optimization and subsequently exploited in data integration. The idea of using views for data citation bears some relationship to that of using them to define security levels in a database.¹¹

Using database views to specify citable units is the key to both specifying and generating citations. It is important for data publishers who want their data to be properly cited to define these views and ensure the data necessary to generate the citation from them is available. We have shown how this can be done for two quite different scientific databases and believe the idea can work on forms of data (such as RDF²⁵) and databases in other fields, including the humanities. We have looked at some examples, and the main barrier is that the data needed to generate the citation may not be available, either in the database or in some metadata repository.¹

In this article, we have focused on the problem of automatically generating citations, but it is almost impossible to do it in isolation from other topics (such as citation standards). For example, the citation snippets required by the curators of our two examples do not quite conform to the DataCite metadata schema;⁹ although DataCite has an entry for a spatial bounding box, it does not have one for a temporal interval as required by MODIS. A good

problem for database research is to determine whether citations generated by a rule are consistent with a given citation schema.

We also mentioned archiving (ensuring fixity) and provenance as related computational challenges, but there are many others. We have tacitly assumed a rather conventional view of citations and how they are used, but there are many ways in which the form and use of citations may change radically (such as papers with 10,000 authors or papers with 10,000 references). Maybe, by analogy with PageRank,¹⁹ there should be some notion of transitivity of credit in citation. These new approaches to the content, structure, and use of citations are all likely to require new ideas from computer science.

Acknowledgments

Tony Harmar, who led the development of GtoPdb, introduced us to the problem of data citation. We are also indebted to Sarah Cohen Boulakia, Jamie Davies, Wenfei Fan, Andreas Rauber, Joanna Sharman, Gianmaria Silvello, and the reviewers for much useful input. This work is supported by National Science Foundation Information and Intelligent Systems grant 1302212 and Engineering and Physical Sciences Research Council grant EP/J017728/1 The Theory and Practice of Social Machines. 

References

- Altman, M. and King, G. A proposed standard for the scholarly citation of quantitative data. *D-Lib Magazine* 13, 3/4 (Mar./Apr. 2007).
- American Geophysical Union. *AGU Publications Data Policy*. Washington, D.C., Dec. 2013; <http://publications.agu.org/author-resource-center/publication-policies/data-policy/> (accessed Nov. 2015)
- American Meteorological Society. *Data Archiving and Citation*. Boston, MA; <https://www.ametsoc.org/ams/index.cfm/publications/authors/journal-and-bams-authors/journal-and-bams-authors-guide/data-archiving-and-citation/> (accessed Nov. 2015)
- Ball, A. and Duke, M. *How to Cite Datasets and Link to Publications*. The Digital Curation Centre, Edinburgh, U.K., 2012; <http://www.dcc.ac.uk/resources/how-guides/cite-datasets> (accessed Nov. 2015)
- Buneman, P. and Silvello, G. A rule-based citation system for structured and evolving datasets. *IEEE Data Engineering Bulletin* 33, 3 (2010), 33–41.
- Coalition on Publishing Data in the Earth and Space Sciences (COPDESS). Statement of commitment from earth and space science publishers and data facilities, 2015; <http://www.copdess.org/statement-of-commitment/> (accessed Nov. 2015)
- CODATA-ICSTI Task Group on Data Citation Standards and Practices. Out of cite, out of mind: The current state of practice, policy, and technology for the citation of data. *Data Science Journal* 12, 0 (2013), CIDCR1–CIDCR75.
- Data Observation Network for Earth (DataONE). *Data Citation and Attribution*; <https://www.dataone.org/citing-dataone> (accessed Nov. 2015)
- DataCite. DataCite metadata schema for the publication and citation of research data; http://schema.datacite.org/meta/kernel-3/doc/DataCite-MetadataKernel_v3.1.pdf

- Deutsch, A., Popa, L., and Tannen, V. Query reformulation with constraints. *SIGMOD Record* 35, 1 (2006), 65–73.
- Fan, W., Chan, C.-Y., and Garofalakis, M. Secure XML querying with security views. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*. ACM Press, New York, 2004, 587–598.
- Federation of Earth Science Information Partners (ESIP). *Data Citation Guidelines for Data Providers and Archives*; <http://doi.org/10.7269/P34F1NNJ>
- FORCE11. Data Citation Synthesis Group: Joint Declaration of Data Citation Principles; <https://www.force11.org/datacitation> (accessed Nov. 2015)
- Halevy, A.Y. Answering queries using views: A survey. *The VLDB Journal* 10, 4 (2001), 270–294.
- International Council for Science Committee on Data for Science and Technology. *Data Citation Standards and Practices*, 2010; <http://www.codata.org/task-groups/data-citation-standards-and-practices> (accessed Nov. 2015)
- Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S. Citation and peer review of data: Moving towards formal data publication. *International Journal of Digital Curation* 6, 2 (2011), 4–37.
- Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Madison, WI, June 3–6). ACM Press, New York, 2002, 233–246.
- McCallum, I., Plag, H.-P., and Fritz, S. GEOSS data citation guidelines: Version 2.0, 2012; http://www.gstss.org/library/GEOSS_Data_Citation_Guidelines_V2.0.pdf (accessed Nov. 2015)
- Page, L., Brin, S., Motwani, R., and Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Stanford InfoLab, Stanford University, Nov. 1999.
- Pawson, A.J., Sharman, J.L. et al. The IUPHAR/BPS Guide to PHARMACOLOGY: An expert-driven knowledgebase of drug targets and their ligands. *Nucleic acids research* 42, D1 (2014), D1098–D1106.
- Pröhl, S. and Rauber, A. Scalable data citation in dynamic, large databases: Model and reference implementation. In *Proceedings of the 2013 IEEE International Conference on Big Data* (Santa Clara, CA, Oct. 6–9). IEEE Press, 2013, 307–312.
- Pröhl, S. and Rauber, A. A scalable framework for dynamic data citation of arbitrary structured data. In *Proceedings of the Third International Conference on Data Management Technologies and Applications* (Vienna, Austria, Aug. 29–31, 2014), 223–230.
- Research Data Alliance Working Group on Data Citation. Making data citable: Case statement; <https://rd-alliance.org/group/data-citation-wg/case-statement/wg-data-citation-making-data-citable-case-statement.html> (accessed Nov. 2015)
- Salomonson, V.V., Barnes, W., and Masuoka, E.J. Introduction to MODIS and an overview of associated activities. In *Earth Science Satellite Remote Sensing: Vol. 1: Science and Instruments*. Springer, Berlin, Heidelberg, Germany, 2006, 12–32.
- Silvello, G. A methodology for citing linked open data subsets. *D-Lib Magazine* 21, 1/2 (2015).

Peter Buneman (opb@inf.ed.ac.uk) is a professor in the School of Informatics at the University of Edinburgh, Edinburgh, U.K.

Susan Davidson (susan@cis.upenn.edu) is the Weiss Professor of Computer and Information Science at the University of Pennsylvania, Philadelphia, PA.

James Frew (frew@ucsb.edu) is an associate professor of environmental informatics in the Bren School of Environmental Science & Management at the University of California, Santa Barbara, CA.

© 2016 ACM 0001-0782/16/09 \$15.00



Watch the author discuss her work in this exclusive *Communications* video. <http://cacm.acm.org/videos/why-data-citation-is-a-computational-problem>

¹ See use cases and linked data sections in the annotated bibliography in the online appendix.

DOI:10.1145/2971322

Smartphone mapping apps routinely fail to follow centuries-old mapmaking dynamic consistency principles and practices.

BY HANAN SAMET, SARANA NUTANONG, AND BRENDAN C. FRUIN

Dynamic Presentation Consistency Issues in Smartphone Mapping Apps

THE SEPTEMBER 2012 introduction of the Apple iPhone 5 smartphone and accompanying iOS 6 software replaced a mapping app on Apple's mobile devices based on Google's map data with an app that uses Apple's map data. It also changed Apple's decisions as to what data is displayed (served to the user) in response to queries, especially implicit ones through manipulation of the viewing window. This change led to significant related changes in the user experience with applications that use and serve map data and resulted in closer scrutiny

of mapping apps on mobile devices, as we do here.

Applications on mobile devices (smartphones and tablets) are not the traditional ones where the map is used in a passive manner, as in atlases, including maps that are browsed leisurely. Maps on mobile devices are instead used in an active manner as a tool to enable such tasks as navigation and location finding, using pan and zoom. In this case, accuracy is paramount, and data quality and lack of quality-assurance policies and protocols by Apple in releasing the iOS 6 mapping app became apparent. This resulted in such errors as misplaced towns; see, for example, Dobson,³ Tumblr,³⁵ and Whitney,³⁶ and misclassified areas, as in Tumblr,³⁵ which persists (see Samet et al.²⁷) The public uproar was so great it eventually led to the dismissal of Apple's leader of its new mapping app project.² Most such errors were fixed in subsequent releases of iOS 6 and its successors iOS 7 and iOS 8.

Notwithstanding this resolution, we have occasionally found the iOS 6, iOS 7, and iOS 8 mapping apps to be lacking from the perspective of presentation consistency when deployed on mobile devices like smartphones due to limited screen "real estate." Surprisingly, criticism from such a perspective was rarely leveled before (see Paolino et al.²¹ and Samet et al.²⁶), as we do here through examples of how they also plague other mapping apps.

Our definition of presentation consistency is motivated by centuries-old classical principles and practices used by cartographers derived from the static way maps have been browsed, as well as by the evolving dynamic ways maps are browsed involving manipulation, and have much to do with the platform used to view them (such as an atlas instead of a smartphone).

Dynamic presentation consistency properties, including pan, zoom, full zoom out, and wraparound, are a result of the manipulation actions users take to browse the map and are often achieved by gesturing (but see also Es-



perança et al.⁴ and Samet et al.²⁵) Pan and zoom consistency properties correspond to the integrity of the available gesturing actions in terms of retaining label information as an action takes place. The premise is that if a spatial entity has been labeled, then the label should persist as long as the spatial entity remains visible in its entirety. The full zoom-out property reflects the user's desire to view the Earth in its entirety rather than being compelled to apply pan operations to do so. Similarly, the wraparound property is an acknowledgment that the Earth is round and again reflects the user's desire to not have to

apply pan operations in the opposite direction to be able to view adjacent locations on the map. These properties are the subject of this article.

The classical static presentation consistency properties involve the undesirability of label overlap and having a reasonable label distribution that is aesthetic in nature in a way the user finds aesthetically appealing. Other static consistency properties include hierarchical consistency, which seeks a consistent way of presenting labels of containing entities by requiring they must be included whenever visible in their entirety, while sibling consistency

» key insights

- For pan and zoom-in operations, if a spatial entity is labeled, then the label should persist as long as the spatial entity remains visible in its entirety.
- Smartphone mapping apps must take into account the small form factor that limits the size of screen real estate while still enabling the whole world to be seen in one view without having to pan.
- Updates to mapping apps do not necessarily result in improvements, especially on smartphones where the small form factor limits what can be seen and manipulated when using gesturing user interfaces, unlike command-line and search-box user interfaces.

Figure 1. The result (b) of panning the map (a) to the left (achieved by a swipe to the right) in the iOS 8 mapping app on an iPhone 5. The result (d) of panning the map (c) to the left (achieved by a swipe to the right) in the iOS 6 mapping app on an iPhone 5. The result (f) of panning the map (e) to the left (achieved by a swipe to the right) in the Android mapping app on Lollipop. Note the absence of The Netherlands in (b) and (d) and Athens in (f), demonstrating object panning inconsistency.



corresponds to labeling all visible spatial entities that are at the same level of the mapping object hierarchy. The static presentation consistency properties pertain to the identity of the labels that should be present and how they are presented; these properties are discussed in greater detail in Samet et al.²⁷

The motivation for our study is to understand how to take advantage of the fact that a map provides an efficient way of accessing spatially referenced data when we cannot look at all of it at once. Our observations are based on our experience building the STEWARD,¹⁴ NewsStand,^{13,28,34} TwitterStand,^{9,11,32} PhotoStand,²³ and TweetPhoto⁷ systems and adapting them (especially NewsStand and TwitterStand) to run on smartphones.^{24,29} These systems access such documents as news and photos with a map-query interface, usually by location, and, to a lesser extent, by topic. In these applications, as well as in many related ones, the map on the smartphone helps users anchor and orient answers to queries in which they want to take advantage of spatial synonyms. In addition, we continue to be motivated by the desire to be able to place spatially referenced information on the map (such as icons for topics, image thumbnails,^{17,18,22} names of particular locations, names of people and diseases,¹² mentions of brands,¹ and any other data that lends itself to being classified using an ontology). Note the Gazetteer used to translate textual specifications to geometric specifications can also be considered an ontology. The result is analogous to a mashup, except that, in our case, the mashup is hierarchical in the sense that, as we zoom in on the map, additional spatially referenced information is displayed that was not sufficiently important to be displayed when we zoom out completely. This zoom-out capability is not available in comparable systems like HealthMap^{6,10} for disease monitoring.

Comparison

In terms of devices, we compare the iOS 6 mapping app (initially iOS version 6.1 on iPhone 5 and most recently 6.1.4), the iOS 7 mapping app (iOS version 7.0 on iPhone 4), the iOS 8 mapping app (iOS version 8.1.2 on iPhone 5), the iOS 5 mapping app (iOS version 5.1.1 on

iPod Touch), the Android mapping app (Maps version 8.0.0 on Android 4.3), Google's iOS mapping app (version 1.0) for Google Maps data (referred to here as the iOS mapping app for Google), and the HERE Maps app on Microsoft's Windows Phone 8 (HERE Maps version 3.5.481.8 with map data 8.0.50.116). Although each vendor's mapping apps are similar, they do not always yield the same result. At times, we also use the qualifiers "old" and "new" to distinguish between the versions of iOS 6 that we used in our initial tests (version 6.1 in October 2012) and in our most recent tests (version 6.1.4 in April 2014 and later), respectively. This distinction is necessary because we observed that the algorithms used to implement the various mapping apps are frequently changed for a particular version of the operating system, even if the operating system is not updated. It is especially true for the label-distribution-and-placement algorithms that we often found yield different results for the same queries. This difference in results is due to all queries being transmitted to the map tile server over the Internet, and the server makes the final decision as to what labels will be placed, and where, on the resulting map tiles that are also transmitted by the server. Do not be surprised if you are unable to always repeat our observations. The important takeaway from them is that the undesirable behaviors of some mapping apps should not be taken as absolutes but as indications of what could possibly go wrong. Our comparison also included the iOS apps of Bing Maps, Nokia Maps, ESRI, MapQuest, and OpenStreetMap.^a

The table here outlines the comparison in terms of presentation consistency properties that any user would want satisfied. Such satisfaction is primarily an issue on smartphones where the screen size is small, thereby requiring

panning and zooming for more information, while not needed on tablets where screens are larger, as described later in greater detail with examples from the various mapping apps. The apps are identified using I5 for iOS 5, I6 for iOS 6, I7 for iOS 7, I8 for iOS 8, A for Android, WP for HERE Maps on Windows Phone 8, IG for iOS of Google, IB for iOS of Bing Maps, IN for iOS of Nokia Maps, IQ for iOS of MapQuest, IO for iOS of OpenStreetMap, and IE for iOS of ESRI. The table denotes whether the property does not (×), partially (P), or holds (✓) for the apps.

Note we do not compare the mapping APIs, as they correspond to a set of features in the programming environments that are designed to make it easy for users to build mapping apps. However, just because a feature is not available in a mapping API does not mean a user cannot deploy a more complex workaround to obtain this functionality in the mapping app being built. Horizontal wraparound is an example operation available in the iOS 6 mapping app but not in the corresponding API, as discussed later. Another example is the proportion of the Earth's surface that can be viewed at the maximum zoom-out level, which is much greater for the Android mapping app than for its corresponding API.

Note, too, the variation in the relative sizes of the screenshots in some of the figures due to the different devices we used. In particular, the screenshots for the Android and Windows Phone mapping Apps are larger due to a 2.5-inch × 4-inch screen (sometimes a 2.5-inch × 4.3-inch screen for the Android) instead

of a 2-inch × 3-inch screen on the iPhone 4 and iPod Touch devices we used. We used the iPod Touch to perform comparisons with the iOS 5 mapping app that is not available on the iPhone 5, as well as with the iOS mapping apps. However, although the iPhone 5 has a 2-inch × 3.5-inch screen, the scope of the maximum zoom-out level for both landscape and portrait modes vis-à-vis the iPod Touch is unchanged. On the other hand, the iPhone 6 Plus, as of September 2014, with a 2.7-inch × 4.8-inch screen retains the same maximum zoom-out level for the landscape mode while slightly enlarging it for the portrait mode.

Panning Consistency

When panning on the map, the objects, as well as the type of the objects (such as cities, states, countries, and continents), being displayed should be consistent and not disappear or change as long as the underlying space is visible, or "panning consistency."³³ This property is usually satisfied when the display screen is relatively large, even though panning consistency conflicts arise at times for the smaller form factors, as in smartphones. We found panning consistency to hold for many of the iOS mapping apps, while we were able to find examples of panning inconsistency for each of the current most commonly used mapping apps.

We also encountered this problem on an iPhone 5 running iOS 6 (see Figure 1c and the result of panning it to the left in Figure 1d) but not on an iPhone 4 running iOS 7. We attribute the latter, not seeing a misapplied label on

Consistency property comparison of mobile mapping apps.

Properties	Apps											
	I6	I7	I8	I5	A	IG	IB	IN	WP	IQ	IO	IE
Hierarchical C	×	×	×	P	P	P	✓	P	×	×	✓	×
Sibling C	×	×	×	✓	×	×	×	×	×	×	✓	×
No Overlaps	✓	✓	✓	✓	✓	✓	P	✓	✓	✓	✓	×
Label Distribution	×	×	P	✓	✓	✓	×	×	×	✓	✓	✓
Panning C	×	×	×	✓	✓	✓	✓	✓	×	✓	✓	✓
Zoom C	×	×	×	✓	×	×	✓	✓	×	✓	✓	P
H Wraparound	✓	✓	✓	×	✓	✓	×	✓	✓	✓	✓	✓
V Wraparound	×	×	×	×	×	×	×	✓	×	✓	×	×
Full Zoom Out	P	P	P	✓	×	P	✓	×	✓	×	✓	✓

a We include the iOS apps of Bing Maps (version 3.03), Nokia Maps (HERE Maps version 1.8) that increasingly serve as the source for Bing Maps,⁸ ESRI (ArcGIS version 2.3.2), MapQuest (version 3.3.1), and OpenStreetMap, or OSM, where open source map data is the basis of OpenSeaMap version 1.1, which is used here. OpenStreetMap could have also been used as the source map data for the MapQuest app.^{16,19,20} Note that the iOS mapping apps developed by Google, MapQuest, Nokia Maps, OSM, Bing Maps, and ESRI were all tested on iOS version 6.1.

an iPhone 4, to the different position of the United Kingdom label on an iPhone 4 running iOS 7 from the one on an iPhone 5 running both iOS 6 and iOS 8, as the iPhone 4 screen is smaller than the iPhone 5 screen.

This particular problem did not arise in the Android mapping app, as the positions of the labels are not rigid in the sense they can change as zoom levels are changed; see, for example, the varying positions of the United Kingdom label in Figure 2. Nevertheless, we did find examples using other objects (such as cities) where panning consistency does not hold for the Android mapping app on Lollipop; for example, Athens, GA, vanishes as the map in Figure 1e is panned to the left, even though there is room for it, resulting in Figure 1f. A similar example using other cities can be constructed for the Windows Phone App (such as Rotterdam in The Netherlands, although not shown here). Note in all these examples, where the objects were represented on the map with a label corresponding to its name and a symbol, as with a hollow circle at its geographic position (such as a city at a zoom level where it can be represented as a point), map app designers would disregard edge cases where the symbol is outside the display window. Most of the time the label is not present in such a case, but we do not deem this absence after a panning operation an instance of panning inconsistency.

As an example of the panning inconsistency of the types of the objects map app designers would seek to avoid, consider the portion of the world map including Africa and Europe using the iOS 7 mapping app on an iPhone 4 in Figure 3a. Note the implicit border in the middle of the display screen so the left half consists of names of countries and the right half consists of names of cities. This implicit border is already an ominous sign, as the map should display the same type of information at all locations. Panning the map to the right (achieved by a swipe to the left) yields Figure 3b where we see the resulting map now consists primarily of names of cities, with a few names of countries on the extreme left and right. Panning the map further to the right (again achieved by a swipe to the left) yields Figure 3c, where the resulting map now consists primarily of names of countries with a few names of cities in which case the name of the containing country is usually given; that is, the map is hierarchically consistent. This behavior is not what is expected, in that the types of the objects that are displayed should be consistent as users pan. It can be said the original data was not consistent so users should not expect it to be consistent once they pan. We do not agree and were not able to repeat this example on the iOS 6 and iOS 8 mapping apps on an iPhone 5.

As an aside, note the presence of Dji-

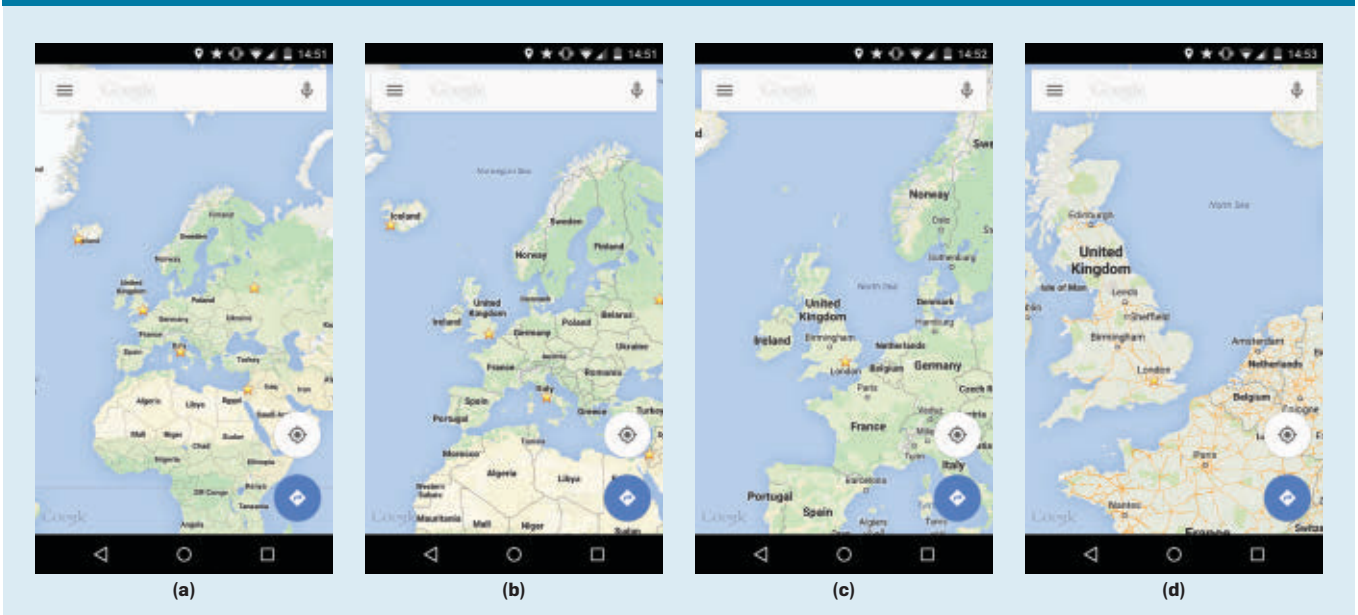
bouti in Figure 3c using a style that capitalizes the first letter, as is used for cities. However, Djibouti is the name of both a city and a country and thus should probably be presented here in upper-case letters. If we zoom in sufficiently in the iOS 7 mapping app (also in iOS 6 and iOS 8), then Djibouti is displayed twice—once in upper case corresponding to its interpretation as a country and once where only the first letter is capitalized, corresponding to its interpretation as a city.

Zoom Consistency

As the user zooms in or pans, names of places that are displayed should continue to be displayed as long as the area they span is visible in its entirety, or “zoom consistency” and “pan consistency.”³³ Names of large containers (such as “United States”) may vanish altogether as the zoom gets very deep. Note the distinction from hierarchical consistency, which deals with just one map view, while zoom consistency is concerned with multiple successive map views.

Consider Figure 4, which shows the failure of zooming consistency to hold for the Android mapping app as we zoom into Europe. The zoom starts in Figure 4a, where Croatia is labeled, while Slovenia is not, even though there is room for its label. Subsequent zooming leads to labeling Zagreb in Croatia in Figure 4b. Further zooming in Figure 4c finds that Ljubljana, the capital

Figure 2. Varying positions of the United Kingdom label in the Android mapping app on Lollipop depending on the zoom-in level, progressing from (a) corresponding to a low zoom-in level to (d) corresponding to the highest zoom-in level.



of Slovenia, is labeled, though Slovenia is not; at this point the app is hierarchically inconsistent. Further zooming in Figure 4d finds both Ljubljana and Zagreb disappear only to be replaced by Venice, while Slovenia is labeled for the first time. Another zoom in Figure 4e causes Zagreb to reappear. Another zoom in Figure 4f fills the display screen with Slovenia with more (and less prominent) cities but still no mention of Ljubljana. A final zoom in Figure 4g finds Ljubljana reappearing with additional cities, but now there is no mention of the containing country Slovenia.

It appears the Slovenia label is missing from Figure 4g in order to avoid overlap on account of the Slovenia label being in approximately the same position on the map as the Ljubljana label. The same example (not shown here) can be used to show that both hierarchical and zooming consistency also fail to hold for the iOS 7 and iOS 8 mapping apps, as well as for the Windows Phone mapping app. The difference, in, say, the iOS 7 and iOS 8 mapping apps is that Slovenia and Croatia are both initially visible, but subsequent zooming in finds Slovenia disappearing only to reappear on further zooming in. However, in the case of the iOS 8 mapping app, when a user zooms in further so Slovenia occupies most of the display screen, then at times both its label and the Ljubljana label are present, in which case the iOS 8 mapping app labels Slovenia with a watermark style font, as in Figure 4h. The figure shows the watermark style font is also used for Croatia, which is the adjacent neighboring country to Slovenia. Interestingly, Apple designers recognized the utility of a watermark style font label yet did not permit it to overlap a label for another object of a different type.

We also observed that, as a user zooms in further on Slovenia in the iOS 8 mapping app, the Slovenia label disappears and the font and style of the Ljubljana label change from the standard first-letter capitalized used for cities (see Figure 4h) to one that uses all caps with wide spacing similar to that used for countries (see Figure 4i). This is in contrast to the Android and Windows Phone mapping apps where the font and style do not vary with the level of zoom. We thus have another variant of zoom inconsistency that involves the

Figure 3. The result (b) of panning the map in (a) containing city and country names to the right (achieved by a swipe to the left) in the iOS 7 mapping app on an iPhone 4 yielding mostly names of cities followed by the result (c) of a further pan to the right yielding mostly names of countries, demonstrating type panning inconsistency.



font/style used for labeling objects of a given type as the user zooms in and out. Observe this is related in spirit to the object type consistency discussed earlier in the context of panning consistency. Also worth noting is the sparseness of the data associated with Slovenia that is common in the mapping apps provided by Apple—iOS 6, iOS 7, and iOS 8—compared with those available for Android (see Figure 4g), iOS 5 (not shown here), and Windows Phone (see Figure 4j).

Zoom inconsistency is also the case for the Windows Phone app for Croatia and Cincinnati (not shown here), as well as for Philadelphia (not shown here) in the iOS mapping app for Google. Observe the Windows Phone mapping app does not use watermark-style font labels, even though the Windows Phone mapping app is rooted in the iOS mapping app for Bing, which used it well. Note, too, that zoom consistency does hold for older mapping apps (such as the iOS 5 mapping app and the iOS mapping apps for Bing, Nokia, MapQuest, and OSM). It also holds partially for the iOS mapping app for ESRI, as long as users do not mind that labels change their location at the different zoom-in and zoom-out levels. This change in locations is fine, as such changes are usually prompted by a need to avoid label overlap, which is usually the highest priority. Oddly, zoom consistency is currently satisfied by the iOS 6 mapping app, though we found it failed consistently in earlier experiments. Finally, this change in locations was also less of a problem on an older version of the

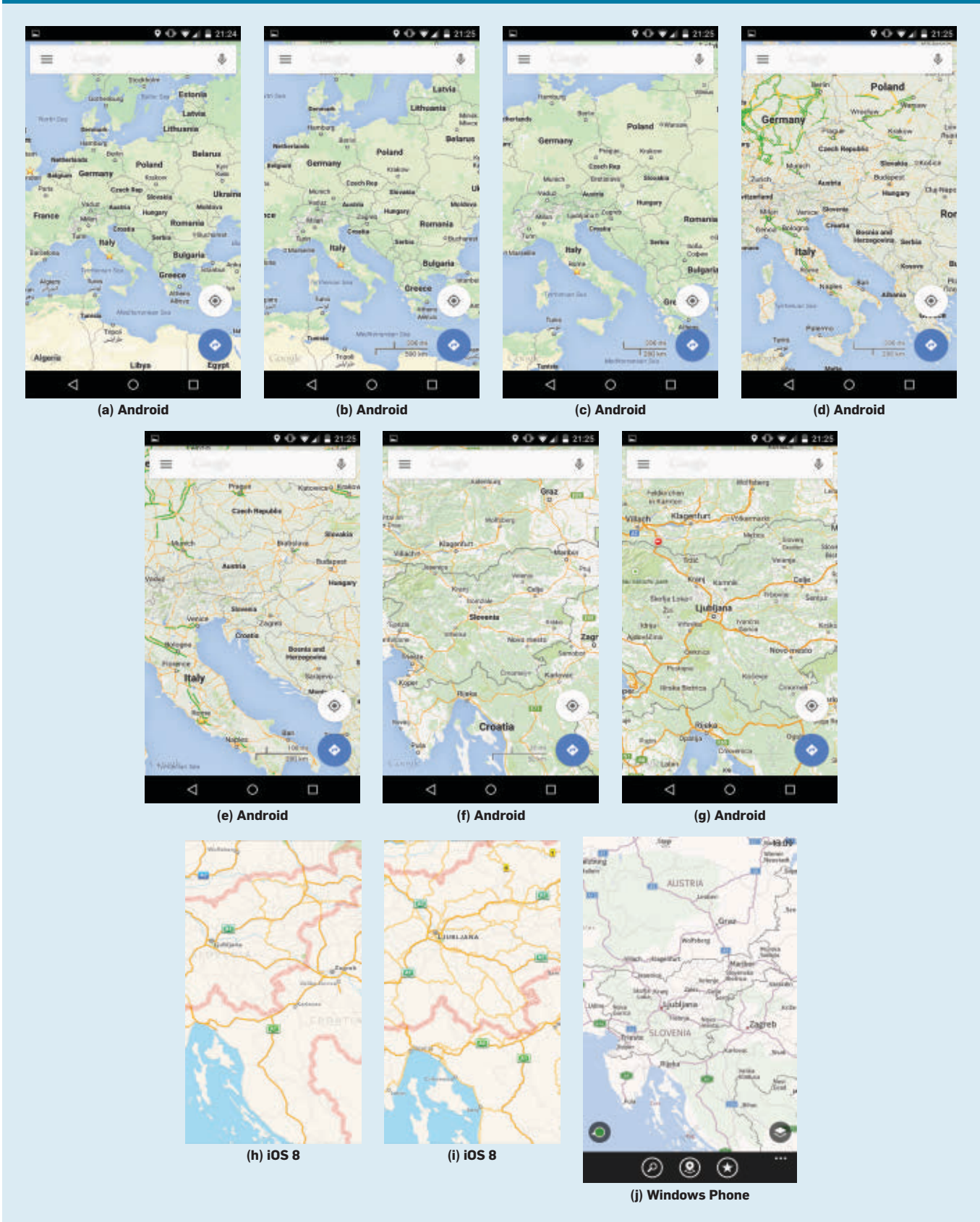
Android mapping app—Google Maps 7.4.0 on Android 4.2.1—where, once the Slovenia label appeared, it stayed put as the user continued to zoom in; the Ljubljana label never appeared until a zoom in at a very high level, at which time the Slovenia label did not appear and, likewise, neither did the Croatia label. During this process, the Zagreb label vanished at times, so this Android variant also failed to satisfy zoom consistency.

Wraparound

A user should be able to view every location both left and right or above and below one another; that is, continuous panning (wraparound) should be possible, as the Earth is a sphere. It is useful in the vicinity of, say, the Bering Strait where, without it, a user would need to do a considerable amount of panning to transition between North America and Asia.

Horizontal wraparound is possible in the iOS 7, iOS 8, Windows Phone, and Android mapping apps and APIs, as well as in the iOS mapping app and API for Google. In addition, horizontal wraparound is possible in the iOS 6 mapping app and the iOS mapping apps for MapQuest, Nokia, ESRI, and OSM. Although the iOS 5 mapping app and the iOS mapping app for Bing do not support horizontal wraparound, they do enable zooming out in landscape mode, so the entire world can be viewed in a single frame, as discussed later. This means there is at least the possibility of viewing both sides of the Bering Strait in one

Figure 4. Example showing zoom inconsistency for the Android mapping app on Lollipop while successively zooming in on Ljubljana in Slovenia and on Croatia. (a) Croatia is labeled initially while Slovenia is not. (b) Zagreb is labeled in Croatia. (c) Ljubljana is labeled while Slovenia is not. (d) Ljubljana and Zagreb vanish while Venice and Slovenia appear. (e) Zagreb reappears. (f) Slovenia fills the map. (g) Ljubljana replaces Slovenia. (h) Zoomed in on Slovenia on the iOS 8 mapping app on an iPhone 5. (i) Zoomed-in on Ljubljana on the iOS 8 mapping app on an iPhone 5. (j) Zoomed-in on Slovenia on the Windows Phone mapping app.



view, though they are not adjacent (see Figure 5a and Figure 6a).

Vertical wraparound is possible in only the iOS mapping apps for MapQuest and Nokia and impossible in the rest. Vertical wraparound is impossible on all of the APIs. Vertical wraparound would be useful in panning around Antarctica or the Arctic. It could also be useful in creating maps where the Southern Hemisphere is on top, in contrast to the prevalent use of the Northern Hemisphere on top. In addition, it would ease making maps like the Wizard of New Zealand's upside-down world map centered in New Zealand.⁵

Full or Maximum Zoom Out

The ability to zoom out completely makes it possible for a user to see the entire world on the device display with one view instead of having to pan the map. Figure 5 and Figure 6 show the extent of its availability in both portrait and landscape modes, respectively, for iOS 5, iOS 8, and Android mapping apps and only for the portrait mode in the Windows Phone mapping app, as the landscape mode option is not presently available for it. At the maximum zoom-out level, the entire world can be seen in both portrait and landscape modes in the iOS mapping apps for Bing and OSM and in portrait mode for the Windows Phone mapping app.

At this level in the iOS 5 mapping app and the iOS mapping app for ESRI, the entire world can be seen in landscape mode, while 95% can be seen in portrait mode. Continuing at this level, the portrait (landscape) iOS mapping apps for Nokia and MapQuest all present a much narrower view of the world, enabling only about 25% (35%) of the world to be seen, while the portrait (landscape) iOS 6, iOS 7, iOS 8, and Android mapping apps, and the iOS mapping app for Google, being remarkably similar in coverage, are somewhere in between, enabling approximately 35% (60%) of the world to be seen. The same behavior as in the mapping app is available in the mapping APIs for iOS 5, iOS 6, iOS 7, iOS 8, Windows Phone, and the iOS mapping API for Google. However, for Android, the mapping API allows the user to see a greater part of the world than can be seen through the mapping app.

At maximum zoom-out level, most mapping apps obey hierarchical consistency, with the exception of the iOS 6, iOS 7, and iOS 8 mapping apps, which present only the names of cities (see Figure 5b and Figure 6b) and not their containing countries.

The iOS 5 and Windows Phone mapping apps display only names of continents and oceans at the maximum zoom-out level. On the other

hand, the iOS 6, iOS 7, and iOS 8 mapping apps also display the names of a few cities (but no countries), while the Android mapping app displays only the names of a few countries (but no continents or cities) at this maximum zoom-out level. The Windows Phone mapping app (see Figure 5d) is the only app able to label and display all seven continents and all five oceans at this maximum zoom-out level. This is done without requiring any panning or additional zooming as needed by the remaining mapping apps and thus is the only one that obeys sibling consistency at this maximum zoom-out level. The iOS 5 mapping app has similar properties that differ only by requiring some panning to see Antarctica and some additional zooming in to see the Arctic and Southern oceans (see Figure 5a and Figure 6a). The Android mapping app shows all oceans upon panning save for the Arctic Ocean, which requires more zooming in.

The iOS 6, iOS 7, and iOS 8 mapping apps label all continents after panning and all oceans save for the Southern Ocean, which they never label regardless of how far they zoom in. The iOS 6, iOS 7, and iOS 8 mapping apps differ primarily in their use of a boldface font for the continent names in the iOS 7 and iOS 8 mapping apps (see Figure 5b and

Figure 5. (a) World map in portrait mode demonstrating the maximum zoom-out level for (a) iOS 5 on an iPod Touch, (b) iOS 8 on an iPhone 5, (c) Android, and (d) Windows Phone mapping apps.

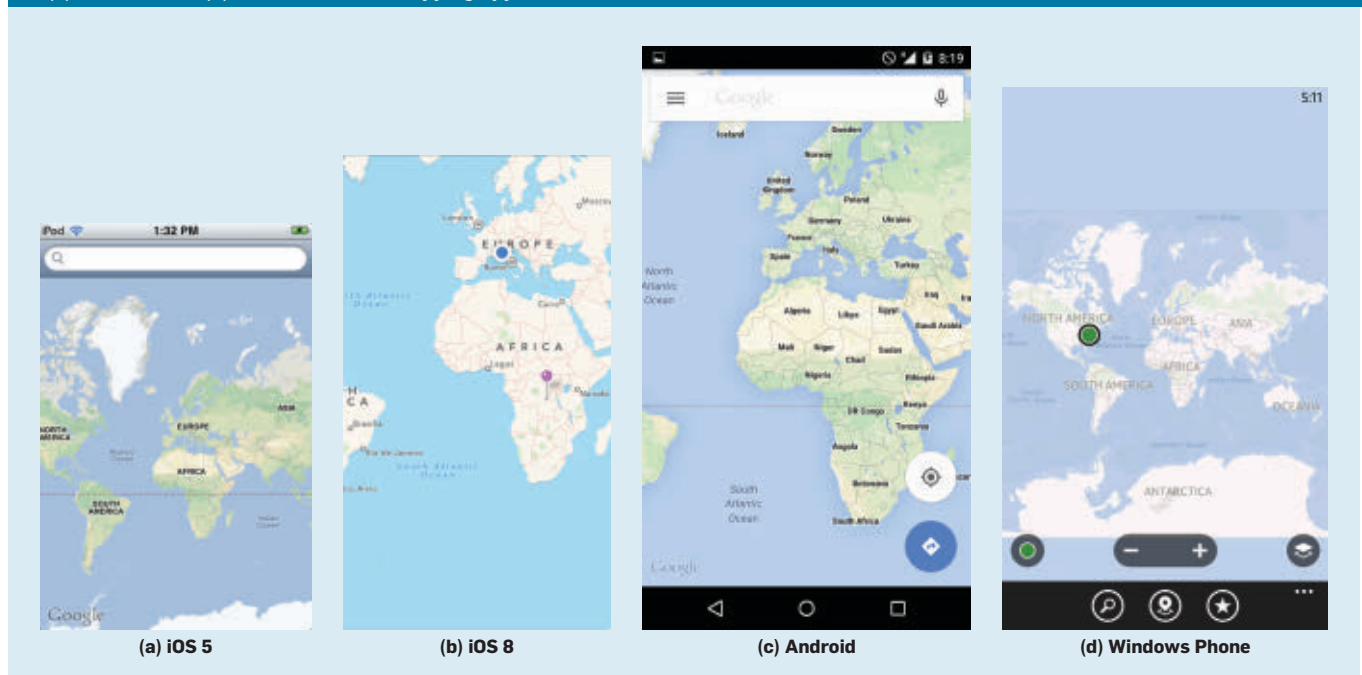


Figure 6. World map in landscape mode demonstrating maximum zoom-out level for (a) iOS 5 on an iPod Touch, (b) iOS 8 on an iPhone 5, and (c) Android mapping apps.



Figure 7. Examples of spatially referenced news clusters with (a) iOS6 mapping app and (b) iOS 5 mapping app and examples of spatially referenced mentions of diseases using (c) iOS 6 mapping app and (d) iOS 5 mapping app.



Figure 6b) and watermark style in the iOS 6 mapping apps (not shown here). As we see, this difference in the labeling of continent names occurs in both landscape and portrait modes for the iOS 6, iOS 7, and iOS 8 mapping apps.

Seeing the whole world is important if a user wants to observe a feature's behavior over the whole world with one view rather than having to pan the map to see the full extent of the feature's behavior. Figure 7a and Figure 7b are examples of spatially referenced news clusters from NewsStand^{13,28,34} using the iOS 6 and iOS 5 mapping apps, respectively, while Figure 7c and Figure 7d are spatially referenced mentions of diseases from the same system using the iOS 6 and iOS 5 mapping apps, respectively. A slider serves to vary the displayed clusters or disease mentions.

Conclusion

Many of the manipulation-presentation-consistency issues we have discussed here could be resolved by caching map data, as discussed by, for example, Liu et al.,¹⁵ and dynamic map labeling by, for example, Peng et al.,²² which are directions for future study. We now review some manipulation-presentation-consistency issues for the various mapping apps we found noteworthy. However, we first re-emphasize that our aim here is not to criticize Apple, Google, or Microsoft. Instead, it is to use examples motivated by Apple's foray into the maps space where Google and Microsoft have a longer history due in part to their work on Microsoft Virtual Earth and Google Earth and Maps, to point out the difficulty of such a task, and the need to consider centuries-old principles and practices in mapmaking.

Despite the obvious similarities in Android and iOS 5 mapping apps, we identified important differences, including the way they deal with the whole world—both portrait and landscape—where only the iOS 5 mapping app provides a full view at the maximum zoom-out level.

From our limited comparison, as summarized in the table, we conclude that newer is not always better in that the iOS 5 mapping app is probably still best with respect to our four properties of presentation consistency—hierarchical, sibling, panning, and

zoom—making for better map-based applications on smartphone form-factor devices. The Android mapping app often performs like the iOS 5 mapping app, though it was plagued by zoom and sibling inconsistency that are also common in the iOS 6, iOS 7 iOS 8, and Windows Phone mapping apps, as well as others. The iOS mapping apps for Nokia and Bing maps were a distant third, with the main advantage for the iOS mapping app for Nokia being the ability to wrap around fully, which was sorely missing in the iOS 5 mapping app.

Note the main emphasis of this article has been to point out manipulation-presentation consistency issues in popular mapping apps. Ideally, any mapping app should satisfy all presentation consistency properties outlined earlier. In practice, however, an app developer may choose to partially satisfy or even completely ignore some of the properties due to such factors as physical space and execution time. In some cases, trade-offs must be made, and a possible future research direction is to study how to make them without compromising the user experience. Such a study would require access to different labeling algorithms and involve some form of usability testing to assess which ones deliver the best performance.

An important by-product of our study is the empirical observation that both the Google Maps API^b and the Bing Maps API^c use crow-flying distance rather than road-network distance, as discussed by Sankaranarayanan and Samet,^{30,31} in their distance-ordering queries, though both use road-network distance when providing directions. This is another example of the failure of modern mapping systems to adhere to centuries-old lessons in dealing with spatial data.

Acknowledgments

This article is based in part on an earlier article by Samet et al.²⁶ This work was supported in part by the National Science Foundation under Grants IIS-10-18475, IIS-12-19023, and IIS-13-20791. We are also grateful to Anis Abboud for his help in generating some of the screenshots. 

b <http://tinyurl.com/GoogKNN>

c <http://tinyurl.com/BingMapsKNN>

References

1. Abdelrazek, A., Hand, E., and Samet, H. Brands in NewsStand: Spatiotemporal browsing of business news. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, WA, Nov. 4–6). ACM Press, New York, 2015, article 97.
2. Biddle, S. Bloomberg: Apple fires guy responsible for crappy Apple Maps. *Gizmodo* (Nov. 27, 2012); <http://gizmodo.com/5963651/apple-firesguy-responsible-for-apple-maps>
3. Dobson, M. Google Maps announces a 400 year advantage over Apple Maps. *TeleMapics* (Sept. 20, 2012); <http://blog.telemapics.com/?p=399>
4. Esperança, C. and Samet, H. Experience with SAND-Tcl: A scripting tool for spatial databases. *Journal of Visual Languages and Computing* 13, 2 (Apr. 2002), 229–255.
5. Flourish. *The Upsidedown Map Page*; <http://flourish.org/upsidedownmap/>
6. Freifeld, C.C., Mandl, K.D., Reis, B.Y., and Brownstein, J.S. Model formulation: HealthMap: Global infectious disease monitoring through automated classification and visualization of Internet media reports. *Journal of the American Medical Informatics Association* 15, 2 (Mar.-Apr. 2008), 150–157.
7. Fruin, B.C., Samet, H., and Sankaranarayanan, J. Tweetphoto: Photos from news tweets. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9). ACM Press, New York, 2012, 582–585.
8. Gabriel, C. Nokia's Bing Maps deal is a sign of the future. *Rethink TV* (May 28, 2012); <http://www.rethink-tv.com/2012/05/28/nokia-bing-maps-deal-sign-future.htm>
9. Gramsky, N. and Samet, H. Seeder finder: Identifying additional needles in the Twitter haystack. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (Orlando, FL, Nov. 5). ACM Press, New York, 2013, 44–53.
10. HealthMap. *HealthMap: Global Health, Local Information* (Sept. 2012); <http://www.healthmap.org/>
11. Jackoway, A., Samet, H., and Sankaranarayanan, J. Identification of live news events using Twitter. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (Chicago, IL, Nov. 1). ACM Press, New York, 2011, 25–32.
12. Lan, R., Lieberman, M.D., and Samet, H. The picture of health: Map-based, collaborative spatiotemporal disease tracking. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Use of GIS in Public Health* (Redondo Beach, CA, Nov. 6). ACM Press, New York, 2012, 27–35.
13. Lieberman, M.D. and Samet, H. Supporting rapid processing and interactive map-based exploration of streaming news. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9). ACM Press, New York, 2012, 179–188.
14. Lieberman, M.D., Samet, H., Sankaranarayanan, J., and Sperling, J. STEWARD: Architecture of a spatio-textual search engine. In *Proceedings of the 15th annual ACM International Symposium on Advances in Geographic Information Systems* (Seattle, WA, Nov. 7–9). ACM Press, New York, 2007, 186–193.
15. Liu, C., Fruin, B.C., and Samet, H. SAC: Semantic adaptive caching for spatial mobile applications. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Orlando, FL, Nov. 6–8). ACM Press, New York, 2013, 184–193.
16. MapQuest; <http://platform.mapquest.com/>
17. Nutanong, S., Adelfio, M.D., and Samet, H. Multi-resolution select distinct queries on large geographic point sets. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9). ACM Press, New York, 2012, 159–168.
18. Nutanong, S., Adelfio, M.D., and Samet, H. An efficient layout method for a large collection of geographic data entries. In *Proceedings of the 16th International Conference on Extending Database Technology* (Genoa, Italy, Mar. 18–22). ACM Press, New York, 2013, 717–720.
19. OpenSeaMap; <http://www.openseamap.org/>
20. OpenStreetMap; <http://www.openstreetmap.org/>
21. Paolino, L., Romano, M., Tortora, G., and Vitello, G. Spatial data visualization on mobile interface: A usability study. In *Proceedings of the Ninth International Wireless Communications & Mobile Computing Conference* (Cagliari, Sardinia, Italy, July 1–5). IEEE Press, New York, 2013, 959–963.
22. Peng, S.F., Adelfio, M.D., and Samet, H. Viewing streaming spatially referenced data at interactive rates. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Dallas, TX, Nov. 5–7). ACM Press, New York, 2014, 409–412.
23. Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., and Sankaranarayanan, J. PhotoStand: A map query interface for a database of news photos. In *Proceedings of the 39th Conference on Very Large Data Bases* (Riva del Garda, Trento, Italy, Aug. 26–30, 2013), 1350–1353.
24. Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., and Teitler, B.E. Porting a Web-based mapping application to a smartphone app. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Chicago, IL, Nov. 2–4). ACM Press, New York, 2011, 525–528.
25. Samet, H., Alborzi, H., Brabec, F., Esperança, C., Hjaltason, G.R., Morgan, F., and Tanin, E. Use of the SAND spatial browser for digital government applications. *Commun. ACM* 46, 1 (Jan. 2003), 61–64.
26. Samet, H., Fruin, B.C., and Nutanong, S. DUKing it out at the smartphone mobile app mapping API corral: Apple, Google, and the competition. In *Proceedings of the Fourth ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems* (Seattle, WA, Nov. 3). ACM Press, New York, 2012, 41–48.
27. Samet, H., Nutanong, S., and Fruin, B.C. Static presentation consistency issues in smartphone mapping apps. *Commun. ACM* 59, 5 (May 2016), 88–98.
28. Samet, H., Sankaranarayanan, J., Lieberman, M.D., Adelfio, M.D., Fruin, B.C., Lotkowski, J.M., Panozzo, D., Sperling, J., and Teitler, B.E. Reading news with maps: The power of searching with spatial synonyms. *Commun. ACM* 57, 10 (Oct. 2014), 64–77.
29. Samet, H., Teitler, B.E., Adelfio, M.D., and Lieberman, M.D. Adapting a map query interface for a gesturing touchscreen interface. In *Proceedings of the 20th International World Wide Web Conference* (Companion Volume) (Hyderabad, India, Mar. 28–Apr. 1). ACM Press, New York, 2011, 257–260.
30. Sankaranarayanan, J. and Samet, H. Query processing using distance oracles for spatial networks. *IEEE Transactions on Knowledge and Data Engineering* 22, 8 (Aug. 2010), 1158–1175.
31. Sankaranarayanan, J. and Samet, H. Distance oracles for spatial networks. In *Proceedings of the 25th IEEE International Conference on Data Engineering* (Shanghai, China, Mar. 29–Apr. 2). IEEE Press, New York, 2009, 652–663.
32. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., and Sperling, J. Twitterstand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, WA, Nov. 4–6). ACM Press, New York, 2009, 42–51.
33. Sarma, A.D., Lee, H., Gonzalez, H., Madhavan, J., and Halevy, A.Y. Efficient spatial sampling of large geographical tables. In *Proceedings of the ACM International Conference on Management of Data* (Scottsdale, AZ, May 20–24). ACM Press, New York, 2012, 193–204.
34. Teitler, B., Lieberman, M.D., Panozzo, D., Sankaranarayanan, J., Samet, H., and Sperling, J. NewsStand: A new view on news. In *Proceedings of the 16th annual ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Irvine, CA, Nov. 5–7). ACM Press, New York, 2008, 144–153.
35. Tumblr. The amazing iOS 6 maps; <http://theamazingios6maps.tumblr.com/>
36. Whitney, L. Apple's Maps app slammed over missing cities and other mistakes. *CNET* (Sept. 20, 2012); <http://www.cnet.com/news/apples-maps-app-slammed-over-missing-cities-and-other-mistakes/>

Hanan Samet (hjs@cs.umd.edu) is a distinguished university professor in the Computer Science Department, Center for Automation Research, and Institute for Advanced Computer Studies at the University of Maryland, College Park, MD.

Sarana Nutanong (nutanong@umiacs.umd.edu) is an assistant professor in the Computer Science Department of the City University of Hong Kong, Hong Kong SAR, China; his research for this article was conducted while he was a research associate at the University of Maryland, College Park, MD.

Brendan C. Fruin (bcfruin@gmail.com) is a software engineer at Zillow, Seattle, WA; his research for this article was conducted while he was a student at the University of Maryland, College Park, MD.

Semantic parsing is a rich fusion of the logical and the statistical worlds.

BY PERCY LIANG

Learning Executable Semantic Parsers for Natural Language Understanding

A LONG-STANDING GOAL of artificial intelligence (AI) is to build systems capable of understanding natural language. To focus the notion of “understanding” a bit, let us say the system must produce an appropriate action upon receiving an input utterance from a human. For example:

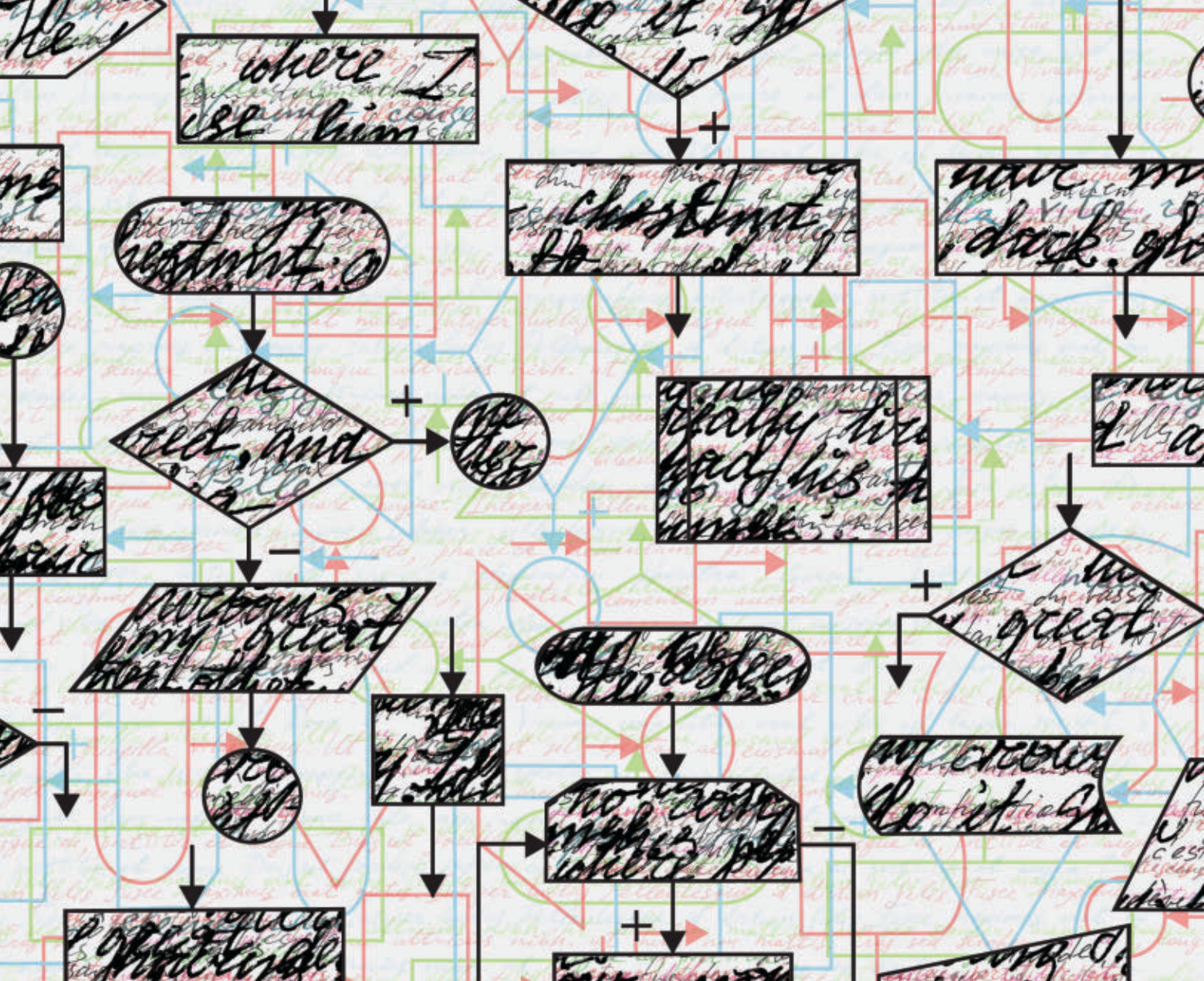
- Context: knowledge of mathematics
Utterance: *What is the largest prime less than 10?*
Action: 7
- Context: knowledge of geography
Utterance: *What is the tallest mountain in Europe?*
Action: Mt. Elbrus
- Context: user’s calendar
Utterance: *Cancel all my meetings after 4pm tomorrow.*
Action: (removes meetings from calendar)



We are interested in utterances such as the ones listed here, which require deep understanding and reasoning. This article focuses on *semantic parsing*, an area within the field of natural language processing (NLP), which has been growing over the last decade.

» key insights

- Natural language understanding can be factored into mapping utterances to logical forms (semantic parsing) and executing logical forms to produce an action.
- It is possible to learn a semantic parser from utterance-action pairs alone by automatically inferring the hidden logical forms.
- The semantic parsing framework is highly modular, consisting of an executor, grammar, model, parser, and learner, each of which can be developed and improved separately, but the system is trained to maximize end-to-end accuracy.



Semantic parsers map input utterances into semantic representations called *logical forms* that support this form of reasoning. For example, the first utterance listed previously would map onto the logical form $\max(\text{primes} \cap (-\infty, 10))$. We can think of the logical form as a program that is *executed* to yield the desired behavior (for example, answering 7). The second utterance would map onto a database query; the third, onto an invocation of a calendar API.

Semantic parsing is rooted in formal semantics, pioneered by logician Montague,²⁵ who famously argued that there is “no important theoretical difference between natural languages and the artificial languages of logicians.” Semantic parsing, by residing in the practical realm, is more exposed to the differences between natural language and logic, but it inherits two general

insights from formal semantics: The first idea is *model theory*, which states that expressions (for example, primes) are mere symbols that only obtain their meaning or denotation (for example, $\{2, 3, 5, \dots\}$) by executing the expression with respect to a model, or in our terminology, a context. This property allows us to factor out the understanding of language (semantic parsing) from world knowledge (execution). Indeed, one can understand the utterance “What is the largest prime less than 10?” without actually computing the answer. The second idea is *compositionality*, a principle often attributed to Gottlob Frege, which states that the denotation of an expression is defined recursively in terms of the denotations of its sub-expressions. For example, *primes* denotes the set of primes, $(-\infty, 10)$ denotes the set of numbers smaller

than 10, and so $\text{primes} \cap (-\infty, 10)$ denotes the intersection of those two sets. This compositionality is what allows us to have a succinct characterization of meaning for a combinatorial range of possible utterances.

Early systems. Logical forms have played a foundational role in natural language understanding systems since their genesis in the 1960s. Early examples included LUNAR, a natural language interface into a database about moon rocks,³⁴ and SHRDLU, a system that could both answer questions and perform actions in a toy blocks world environment.³² For their time, these systems were significant achievements. They were able to handle fairly complex linguistic phenomena and integrate syntax, semantics, and reasoning in an end-to-end application. For example, SHRDLU was able to process “Find a

block which is taller than the one you are holding and put it into the box." However, as the systems were based on hand-crafted rules, it became increasingly difficult to generalize beyond the narrow domains and handle the intricacies of general language.

Rise of machine learning. In the early 1990s, influenced by the successes of statistical techniques in the neighboring speech recognition community, the field of NLP underwent a statistical revolution. Machine learning offered a new paradigm: Collect *examples* of the desired input-output behavior and then fit a statistical model to these examples. The simplicity of this paradigm coupled with the increase in data and computation allowed machine learning to prevail.

What fell out of favor was not only rule-based methods, but also the natural language understanding problems. In the statistical NLP era, much of the community's attention turned to tasks—documentation classification, part-of-speech tagging, and syntactic parsing—which fell short of full end-to-end understanding. Even question answering systems relied less on understanding and more on a shallower analysis coupled with a large collection of unstructured text documents,¹⁰ typified by the TREC competitions.

Statistical semantic parsing. The spirit of deep understanding was kept alive by researchers in statistical semantic parsing.^{19,24,33,36,37} A variety of different semantic representations and learning algorithms were employed, but all of these approaches relied on having a labeled dataset of natural language utterances paired with annotated logical forms, for example:

Utterance: *What is the largest prime less than 10?*
 Logical form: $\max(\text{primes} \cap (-\infty, 10))$

Weak supervision. Over the last few years, two exciting developments have really spurred interest in semantic parsing. The first is reducing the amount of supervision from annotated logical forms to answers:^{13,21}

Utterance: *What is the largest prime less than 10?*
 Action: 7

This form of supervision is much easier to obtain via crowdsourcing. Although the logical forms are not observed, they are still modeled as latent variables, which must be inferred from the answer. This results in a more difficult learning problem, but Liang et al.²¹ showed it is possible to solve it without degrading accuracy.

Scaling up. The second development is the scaling up of semantic parsers to more complex domains. Previous semantic parsers had only been trained on limited domains such as U.S. geography, but the creation of broad-coverage knowledgebases such as Freebase⁸ set the stage for a new generation of semantic parsers for question answering. Initial systems required annotated logical forms,¹¹ but soon, systems became trainable from answers.^{4,5,18} Semantic parsers have even been extended beyond fixed knowledgebases to semi-structured tables.²⁶ With the ability to learn semantic parsers from question-answer pairs, it is easy to collect datasets via crowdsourcing. As a result, semantic parsing datasets have grown by an order of magnitude.

In addition, semantic parsers have been applied to a number of applications outside question answering: robot navigation,^{2,29} identifying objects in a scene,^{15,23} converting natural language to regular expressions,¹⁷ and many others.

Outlook. Today, the semantic parsing community is a vibrant field, but it is still young and grappling with the complexities of the natural language understanding problem. Semantic parsing poses three types of challenges:

- *Linguistic:* How should we represent the semantics of natural language and construct it compositionally from the natural language?
- *Statistical:* How can we learn semantic parsers from weak supervision and *generalize* well to new examples?
- *Computational:* How do we efficiently search over the combinatorially large space of possible logical forms?

In the remainder of this article, we present a general framework for semantic parsing, introducing the key components. The framework is

pleasantly modular, with different choices of the components corresponding to existing semantic parsers in the literature.

Framework

Natural language understanding problem. In this article, we focus on the following natural language understanding problem: Given an *utterance* x in a *context* c , output the desired *action* y . Figure 1 shows the setup for a question answering application, in which case x is a question, c is a knowledgebase, and y is the answer. In a robotics application, x is a command, c represents the robot's environment, and y is the desired sequence of actions to be carried by the robot.²⁹ To build such a system, assume that we are given a set of n examples $\{(x_i, c_i, y_i)\}_{i=1}^n$. We would like to use these examples to train a model that can generalize to new unseen utterances and contexts.

Semantic parsing components. This article focuses on a statistical semantic parsing approach to the above-mentioned problem, where the key is to posit an intermediate *logical form* z that connects x and y . Specifically, z captures the semantics of the utterance x , and it also executes to the action y (in the context of c). In our running example, z would be $\max(\text{primes} \cap (-\infty, 10))$. Our semantic parsing framework consists of the following five components (see Figure 2):

1. **Executor:** computes the denotation (action) $y = [z]_c$ given a logical form z and context c . This defines the semantic representation (logical forms along with their denotations).
2. **Grammar:** a set of rules G that produces $D(x, c)$, a set of candidate derivations of logical forms.
3. **Model:** specifies a distribution $p_\theta(d|x, c)$ over derivations d parameterized by θ .
4. **Parser:** searches for high probability derivations d under the model p_θ .
5. **Learner:** estimates the parameters θ (and possibly rules in G) given training examples $\{(x_i, c_i, y_i)\}_{i=1}^n$.

We now instantiate each of these components for our running example: "*What is the largest prime less than 10?*"

Executor. Let the semantic representation be the language of mathematics, and the executor is the standard interpretation, where the interpretations of predicates (for example, `primes`) are given by c . With $c(\text{primes}) = \{2, 3, 5, 7, 11, \dots\}$, the denotation is $\llbracket \text{primes} \cap (-\infty, 10) \rrbracket_c = \{2, 3, 5, 7\}$.

Grammar. The grammar G connects utterances to possible *derivations* of logical forms. Formally, the grammar is a set of rules of the form $\alpha \Rightarrow \beta$.^a Here is a simple grammar for our running example:

(R1) <i>prime</i>	$\Rightarrow$ NP[primes]
(R2) <i>10</i>	$\Rightarrow$ NP[10]
(R3) <i>less than</i> NP[z]	$\Rightarrow$ QP[$(-\infty, z)$]
(R4) NP[z ₁] QP[z ₂]	$\Rightarrow$ NP[z ₁ $\cap$ z ₂]
(R5) <i>largest</i> NP[z]	$\Rightarrow$ NP[max(z)]
(R6) <i>largest</i> NP[z]	$\Rightarrow$ NP[min(z)]
(R7) <i>What is the</i> NP[z]?	$\Rightarrow$ ROOT[z]

We start with the input utterance and repeatedly apply rules in G . A rule $\alpha \Rightarrow \beta$ can be applied if some span of the utterance matches α , in which case a derivation over the same span with a new syntactic category and logical form according to β is produced. Equation (1) illustrates one possible derivation (call it d_1) for our running example.

What is the	largest	prime	less than	10	?
		(R1) —			(R2) —
		NP[primes]			NP[10]
		(R3) —			
		QP[$(-\infty, 10)$]			
		(R4) —			
		NP[primes $\cap$ $(-\infty, 10)$]			
		(R5) —			
		NP[max(primes $\cap$ $(-\infty, 10)$)]			
		(R7) —			
		ROOT[max(primes $\cap$ $(-\infty, 10)$)]			

(1)

For example, applying (R3) produces category QP and logical form $(-\infty, 10)$ over span [5:7] corresponding to “less than 10.” We stop when we produce the designated ROOT category over the entire utterance. Note that we could have also applied (R6) instead of (R5) to generate the (incorrect) logical form $\min(\text{primes} \cap (-\infty, 10))$; let this derivation be d_2 . We have $D(x, c) = \{d_1, d_2\}$ here, but in general, there could be exponentially many derivations, and

^a The standard way context-free grammar rules are written is $b \rightarrow a$. Because our rules build logical forms, reversing the arrow is more natural.

multiple derivations can generate the same logical form. In general, the grammar might contain nonsense rules (R6) that do not reflect ambiguity in language but are rather due to model uncertainty prior to learning.

Model. The model scores the set of candidate derivations generated by the grammar. A common choice used by virtually all existing semantic parsers are log-linear models (generalizations of logistic regressions). In a log-linear model, define a *feature vector* $\phi(x, c, d) \in \mathbb{R}^F$ for each possible derivation d . We can think of each feature as casting a vote for various derivations d based on some coarse property of the derivation. For example, define $F = 7$ features, each counting the number of times a given grammar rule is invoked in d , so that $\phi(x, c, d_1) = [1, 1, 1, 1, 1, 0, 1]$ and $\phi(x, c, d_2) = [1, 1, 1, 1, 0, 1, 1]$.

Next, let $\theta \in \mathbb{R}^F$ denote the *parameter vector*, which defines a weight for each feature representing how reliable that feature is. Their weighted combination $\text{score}(x, c, d) = \phi(x, c, d) \cdot \theta$ represents how good the derivation is. We can exponentiate and normalize these scores to obtain a distribution over derivations as shown in Equation (2).

$$p_\theta(d | x, c) = \frac{\exp(\text{score}(x, c, d))}{\sum_{d' \in D(x, c)} \exp(\text{score}(x, c, d'))}. \quad (2)$$

If $\theta = [0, 0, 0, 0, +1, -1, 0]$, then p_θ would assign probability $\frac{\exp(1)}{\exp(1) + \exp(-1)} \approx 0.88$ to d_1 and ≈ 0.12 to d_2 .

Parser. Given a trained model p_θ , the parser (approximately) computes the highest probability derivation(s) for an utterance x under p_θ . Assume the utterance x is represented as a sequence of tokens (words). A standard approach is to use a *chart parser*, which recursively builds derivations for each span of the utterance. Specifically, for each category A and span $[i : j]$ (where $0 \leq i < j \leq \text{length}(x)$), we loop over the applicable rules in the grammar G and apply each one to build new derivations of category A over $[i : j]$. For binary rules—those of the form $B C \Rightarrow A$ such as (R4), we loop over split points k (where $i < k \leq j$), recursively compute derivations $B[z_1]$ over $[i : k]$ and $C[z_2]$ over $[k : j]$, and combine them into a new derivation $A[z]$ over $[i : j]$, where z is determined by the rule; for example, $z = z_1 \cap z_2$ for (R4). The final derivations for the utterance are collected in the ROOT category over span $[0 : \text{length}(x)]$.

This procedure would generate all derivations, which could be

Figure 1. A natural language understanding problem where the goal is to map an utterance x in a context c to an action y .

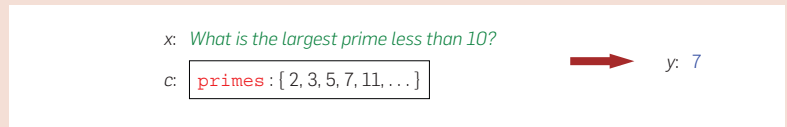
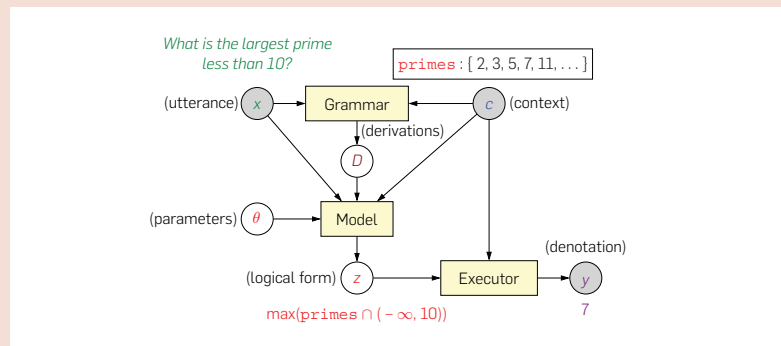


Figure 2. Semantic parsing framework depicting the executor, grammar, and model. The parser and learner are algorithmic components that are responsible for generating the logical form z and parameters q , respectively.



exponentially large. Generally, we only wish to compute the derivations with high probability under our model p_θ . If the features of p_θ were to *decompose* as a sum over the rule applications in d —that is, $\phi(x, c, d) = \sum_{(r, i, j) \in d} \phi_{\text{rule}}(x, c, r, i, j)$, then we could use dynamic programming: For each category A over $[i:j]$, compute the highest probability derivation. However, in executable semantic parsing, feature decomposition isn't sufficient, since during learning, we also need to incorporate the constraint that the logical form executes to the true denotation ($\mathbb{I}[\llbracket d.z \rrbracket_c = y] \rrbracket$; see Equation (6). To maintain exact computation in this setting, the dynamic programming state would need to include the entire logical form $d.z$, which is infeasible, since there are exponentially many logical forms. Therefore, *beam search* is generally employed, where we keep only the K sub-derivations with the highest model score based on only features of the sub-derivations. Beam search is not guaranteed to return the K highest scoring derivations, but it is often an effective heuristic.

Learner. While the parser turns parameters into derivations, the learner solves the inverse problem. The dominant paradigm in machine learning is to set up an objective function and optimize it. A standard principle is to maximize the likelihood of the training data $\{(x_i, c_i, y_i)\}_{i=1}^n$. An important point is that we do not observe the correct derivation for each example, but only the action y_i , so we must consider all derivations d whose logical form $d.z$ satisfy $\llbracket d.z \rrbracket_{c_i} = y_i$. This results in the log-likelihood of the observed action y_i :

$$\mathcal{O}_i(\theta) \stackrel{\text{def}}{=} \log \sum_{\substack{d \in D(x_i, c_i) \\ \llbracket d.z \rrbracket_{c_i} = y_i}} p_\theta(d | x_i, c_i). \quad (3)$$

The final objective is then simply the sum across all n training examples:

$$\mathcal{O}(\theta) \stackrel{\text{def}}{=} \sum_{i=1}^n \mathcal{O}_i(\theta), \quad (4)$$

The simplest approach to maximize $\mathcal{O}(\theta)$ is to use *stochastic gradient descent* (SGD), an iterative algorithm that takes multiple passes (for example, say 5) over the training data and makes the following update on example i :

$$\theta \leftarrow \theta + \eta \nabla \mathcal{O}_i(\theta), \quad (5)$$

where η is a step size that governs how aggressively we want to update parameters (for example, $\eta = 0.1$). In the case of log-linear models, the gradient has a nice interpretable form:

$$\nabla \mathcal{O}_i(\theta) = \sum_{d \in D(x_i, c_i)} (q(d) - p_\theta(d | x_i, c_i)) \phi(x_i, c_i, d), \quad (6)$$

where $q(d) \propto p_\theta(d | x_i, c_i) \mathbb{I}[\llbracket d.z \rrbracket_{c_i} = y_i]$ is the model distribution p_θ over derivations d , but restricted to ones consistent with y_i . The gradient pushes θ to put more probability mass on q and less on p_θ . For example, if p_θ assigns probabilities $[0.2, 0.4, 0.1, 0.3]$ to four derivations and the middle two derivations are consistent, then q assigns probabilities $[0, 0.8, 0.2, 0]$.

The objective function $\mathcal{O}(\theta)$ is not concave, so SGD is at best guaranteed to converge to a local optimum, not a global one. Another problem is we cannot enumerate all derivations $D(x_i, c_i)$ generated by the grammar, so we approximate this set with the result of beam search, which yields K candidates (typically $K = 200$); p_θ is normalized over this set. Note this candidate set depends on the current parameters θ , resulting a heuristic approximation of the gradient $\nabla \mathcal{O}_i$.

Summary. We have covered the components of a semantic parsing system. Observe that the components are relatively loosely coupled: The executor is concerned purely with what we want to express independent of how it would be expressed in natural language. The grammar describes how candidate logical forms are constructed from the utterance but does not provide algorithmic guidance nor specify a way to score the candidates. The model focuses on a particular derivation and defines features that could be helpful for predicting accurately. The parser and the learner provide algorithms largely independent of semantic representations. This modularity allows us to improve each component in isolation.

Refining the Components

Having toured the components of a semantic parsing system, we now return to each component and discuss the key design decisions and possibilities for improvement.

Executor. By describing an executor, we are really describing the

language of the logical form. A basic textbook representation of language is *first-order logic*, which can be used to make quantified statements about relations between objects. For example, “Every prime greater than two is odd.” would be expressed in first-order logic as $\forall x. \text{prime}(x) \wedge \text{more}(x, 2) \rightarrow \text{odd}(x)$. Here, the context c is a model (in the model theory sense), which maps predicates to sets of objects or object pairs. The execution of this logical form with respect to the standard mathematical context would be `true`. Blackburn and Bos⁷ gives a detailed account on how first-order logic is used for natural language semantics.

First-order logic is reasonably powerful, but it fails to capture some common phenomena in language. For example, “How many primes are less than 10?” requires constructing a set and manipulating it and thus goes beyond the power of first-order logic. We can instead augment first-order logic with constructs from *lambda calculus*. The logical form corresponding to the preceding question would be `count( $\lambda x. \text{prime}(x) \wedge \text{less}(x, 10)$ )`, where the λ operator can be thought of as constructing a set of all x that satisfy the condition; in symbols, $\llbracket \lambda x. f(x) \rrbracket_c = \{x : \llbracket f(x) \rrbracket_c = \text{true}\}$. Note that `count` is a higher-order function that takes a function as an argument.

Another logical language, which can be viewed as syntactic sugar for lambda calculus, is *lambda dependency-based semantics* (DCS).²⁰ In lambda DCS, the logical form would be `count(prime  $\sqcap$  (less . 10))`, where the constant 10 represent $\lambda x. (x = 10)$, the intersection operator $z_1 \sqcap z_2$ represents $\lambda x. z_1(x) \wedge z_2(x)$, and the join operator $r.z$ represents $\lambda x. \exists y. r(x, y) \wedge z(y)$.

Lambda DCS is “lifted” in the sense that operations combine functions from objects to truth values (think sets) rather than truth values. As a result, lambda DCS logical forms partially eliminate the need for variables. Noun phrases in natural language (for example, “prime less than 10”) also denote sets. Thus, lambda DCS arguably provides a transparent interface with natural language.

From a linguistic point of view, the logical language seeks primarily to model natural language phenomena. From an application point of view, the

logical language dictates what actions we support. It is thus common to use application-specific logical forms, for example, regular expressions.¹⁷ Note that the other components of the framework are largely independent of the exact logical language used.

Grammar. Recall that the goal of the grammar in this article is just to define a set of candidate derivations for each utterance and context. Note that this is in contrast to a conventional notion of a grammar in linguistics, where the goal is to precisely characterize the set of valid sentences and interpretations. This divergence is due to two factors: First, we will learn a statistical model over the derivations generated by the grammar anyway, so the grammar can be simple and coarse. Second, we might be interested in application-specific logical forms. In a flight reservation domain, the logical form we wish to extract from “*I’m in Boston and would like to go to Portland*” is $\text{flight } \square \text{ from.Boston } \square \text{ to.Portland}$, which is certainly not the full linguistic meaning of the utterance, but suffices for the task at hand. Note the connection here between language and logic is less direct compared to “*prime less than 10*” $\Rightarrow \text{prime } \square (\text{less.10})$.

CCG. One common approach to the grammar in semantic parsing is *combinatory categorial grammar* (CCG),²⁸ which had been developed extensively in linguistics before it was first used for semantic parsing.³⁷ CCG is typically coupled with logical forms in lambda calculus. Equation (7) shows an example of a CCG derivation:

$$\begin{array}{c}
 \text{prime} \qquad \text{less than} \qquad 10 \\
 \hline
 \text{N}[\lambda x.\text{prime}(x)] \quad \text{N}[\lambda y.\text{NP}[\lambda y.\lambda x.f(x) \wedge \text{less}(x,y)] \quad \text{NP}[10] \\
 \hline
 \text{N}[\lambda x.\lambda y.f(x) \wedge \text{less}(x,10)] \\
 \hline
 \text{N}[\lambda x.\text{prime}(x) \wedge \text{less}(x,10)]
 \end{array}
 \quad (7)$$

Syntactic categories in CCG include nouns (N) denoting sets and NP denoting objects. Composite categories ((NN)/NP) represent functions of multiple arguments, but where the directionality of the slashes indicate the location of the arguments. Most rules in CCG are lexical (for example, $[\text{prime} \Rightarrow \text{N}[\lambda x.\text{prime}(x)]]$), which mention particular words. The rest of the rules are glue; for example, we have a forward

From a linguistic point of view, the logical language seeks primarily to model natural language phenomena. From an application point of view, the logical language dictates what actions we support.

application ($>$) and backward application ($<$) rule:

$$\begin{array}{lcl}
 (>) & A/B[f] \quad B[x] & \Rightarrow A[f(x)] \\
 (<) & B[x] \quad A \backslash B[f] & \Rightarrow A[f(x)]
 \end{array}$$

It is common to use other combinators that both handle more complex linguistic phenomena such as NP coordination “*integers that divide 2 or 3*” as well as ungrammatical language,³⁸ although these issues can also be handled by having a more expansive lexicon.¹⁹

Crude rules. In CCG, the lexical rules can become quite complicated. An alternative approach that appears mostly in work on lambda DCS is to adopt a much cruder grammar whose rules correspond directly to the lambda DCS constructions, join and intersect. This results in the derivation shown in Equation (8):

$$\begin{array}{c}
 \text{prime} \quad \text{less than} \quad 10 \\
 \hline
 \text{N}[\text{prime}] \quad \text{N}[\text{less}] \quad \text{N}[10] \\
 \hline
 \text{N}[\text{less.10}] \quad (\text{join}) \\
 \hline
 \text{N}[\text{prime } \square \text{ less.10}] \quad (\text{intersect})
 \end{array}
 \quad (8)$$

$$\begin{array}{lcl}
 (\text{join}) & \text{N}[\text{N}[r]] \quad \text{N}[z] & \Rightarrow \text{N}[r;z] \\
 (\text{intersect}) & \text{N}[z_1] \quad \text{N}[z_2] & \Rightarrow \text{N}[z_1 \square z_2]
 \end{array}$$

The lack of constraints permits the derivation of other logical forms such as $10 \square \text{less.prime}$, but these incorrect logical forms can be ruled out statistically via the model. The principle is to keep the lexicon simple and lean more heavily on features (whose weights are learned from data) to drive the selection of logical forms.

Floating rules. While this new lexicon is simpler, the bulk of the complexity comes from having it in the first place. Where do the rules $[\text{prime} \Rightarrow \text{N}[\text{N} : \text{prime}]]$ and $[\text{less than} \Rightarrow \text{N}[\text{N} : \text{less}]]$ come from? We can reduce the lexicon even further by introducing *floating rules* $[\emptyset \Rightarrow \text{N}[\text{N} : \text{prime}]]$ and $[\emptyset \Rightarrow \text{N}[\text{N} : \text{less}]]$, which can be applied in the absence of any lexical trigger. This way, the utterance “*primes smaller than 10*” would also be able to generate $\text{prime } \square \text{less.10}$, where the predicates *prime* and *less* are not anchored to any specific words.

While this relaxation may seem linguistically blasphemous, recall the purpose of the grammar is to merely deliver a set of logical forms, so floating rules

are quite sensible provided we can keep the set of logical forms under control. Of course, since the grammar is so flexible, even more nonsensical logical forms are generated, so we must lean heavily on the features to score the derivations properly. When the logical forms are simple and we have strong type constraints, this strategy can be quite effective.^{5,26,30}

The choice of grammar is arguably the most important component of a semantic parser, as it most directly governs the trade-off between expressivity and statistical/computational complexity. We have explored three grammars, from a tightly constrained CCG to a very *laissez faire* floating grammar. CCG is natural for capturing complex linguistic phenomena in well-structured sentences, whereas for applications where the utterances are noisy but the logical forms are simple, more flexible grammars are appropriate. In practice, one would probably want to mix and match depending on the domain.

Model. At the core of a statistical model is a function $\text{score}(x, c, d)$ that judges how good a derivation d is with respect to the utterance x and context c . Previously, we described a simple log-linear model (2) in which $\text{score}(x, c, d) = \phi(x, c, d) \cdot \theta$, and with simple rule features. There are two ways to improve the model: use more expressive features and use a nonlinear scoring function.

Most existing semantic parsers stick with a linear model and leverage more targeted features. One simple class of features $\{\phi_{a,b}\}$ is equal to the number of times word a appears in the utterance and predicate b occurs in the logical form; if the predicate b is generated by a floating rule, then a is allowed to range over the entire utterance; otherwise, it must appear in the span of b .

These features are quite numerous, which provides flexibility but require a lot of data. Note that logical predicates (for example, *birthplace*) and the natural language (for example, *born in*) often share a common vocabulary (English words). One can leverage this structure by defining a few features based on statistics from large corpora, external lexical resources, or simply string overlap. For example, ϕ_{match} might be the number of word-predicate



Going forward, the two big questions are: How to represent the semantics of language and what supervision to use to learn the semantics from.



pairs that match almost exactly. This way, the lexical mapping need not be learned from scratch using only the training data.^{4,18}

A second class of useful features is based on the denotation $\llbracket d.z \rrbracket_c$ of the predicted logical form. For example, in a robot control setting, a feature would encode whether the predicted actions are valid in the environment c (picking up a cup requires the cup to be present). These features make it amply clear that semantic parsing is not simply a natural language problem but one that involves jointly reasoning about the nonlinguistic world. In addition, we typically include type constraints (features with weight $-\infty$) to prune out ill-formed logical forms such as *birthplace*⁴ that are licensed the grammar.

One could also adopt a nonlinear scoring function, which does not require any domain knowledge about semantic parsing. For example, one could use a simple one-layer neural network, which takes a weighted combination of m nonlinear basis functions: $\text{score}(x, c, d) = \sum_{i=1}^m \alpha_i \tanh(\phi(x, c, d) \cdot w_i)$. The parameters of the model we would learn are then the m top-level weights $\{\alpha_i\}$ and the mF bottom-level weights $\{w_{ij}\}$; recall that F is the number of features ($\phi(x, c, d) \in \mathbb{R}^F$). With more parameters, the model becomes more powerful, but also requires more data to learn.

Parsing. Most semantic parsing algorithms are based on chart parsing, where we recursively generate a set of candidate derivations for each syntactic category (for example, NP) and span (for example, $[3 : 5]$). There are two disadvantages of chart parsing: First, it does not easily support incremental contextual interpretation: The features of a span $[i : j]$ can only depend on the sub-derivations in that span, not on the derivations constructed before i . This makes anaphora (resolution of “it”) difficult to model. A solution is to use shift-reduce parsing rather than chart parsing.⁴⁰ Here, one parses the utterance from left to right, and new sub-derivations can depend arbitrarily on the sub-derivations constructed thus far.

A second problem is the chart parsing generally builds derivations in some fixed order—for example, of increasing span size. This causes the parser to waste equal resources

on non-promising spans which are unlikely to participate in the final derivation that is chosen. This motivates agenda-based parsing, in which derivations that are deemed more promising by the model are built first.⁶

Learner. In learning, we are given examples of utterance-context-response triples (x, c, y) . There are two aspects of learning: inducing the grammar rules and estimating the model parameters. It is important to remember that practical semantic parsers do not do everything from scratch, and often the hard-coded grammar rules are as important as the training examples. First, some lexical rules that map named entities (for example, $[paris \Rightarrow \text{ParisFrance}]$), dates, and numbers are generally assumed to be given,³⁷ though we need not assume these rules are perfect.²¹ These rules are also often represented implicitly.^{4, 21}

How the rest of the grammar is handled varies across approaches. In CCG-style approach, inducing lexical rules is an important part of learning. In Zettlemoyer and Collins,³⁷ a procedure called GENLEX is used to generate candidate lexical rules from a utterance-logical form pair (x, z) . A more generic induction algorithm based on higher-order unification does not require any initial grammar.¹⁹ Wong and Mooney³³ use machine translation ideas to induce a synchronous grammar (which can also be used to generate utterances from logical forms). However, all these lexicon induction methods require annotated logical forms z . In approaches that learn from denotations y ,^{4, 21} an initial crude grammar is used to generate candidate logical forms, and rest of the work is done by the features.

As we discussed earlier, parameter estimation can be performed by SGD on the log-likelihood; similar objectives based on max-margin are also possible.²² It can be helpful to also add an L_1 regularization term $\lambda \|\theta\|_1$, which encourages feature weights to be zero, which produces a more compact model that generalizes better.⁵ In addition, one can use AdaGrad,¹⁴ which maintains a separate step size for each feature. This can improve stability and convergence.

Datasets and Results

In a strong sense, datasets are the main driver of progress for statistical

approaches. We will now survey some of the existing datasets, describe their properties, and discuss the state of the art.

The Geo880 dataset³⁶ drove nearly a decade of semantic parsing research. This dataset consists of 880 questions and database queries about U.S. geography (for example, “*what is the highest point in the largest state?*”). The utterances are compositional, but the language is clean and the domain is limited. On this dataset, learning from logical forms¹⁸ and answers²¹ both achieve around 90% accuracy.

The ATIS-3 dataset³⁸ consists of 5418 utterances paired with logical forms (for example, “*show me information on american airlines from fort worth texas to philadelphia*”). The utterances contain more disfluencies and flexible word order compared with Geo880, but they are logically simpler. As a result, slot filling methods have been a successful paradigm in the spoken language understanding community for this domain since the 1990s. The best reported result is based on semantic parsing and obtains 84.6% accuracy.³⁸

The Regexp824 dataset¹⁷ consists of 824 natural language and regular expression pairs (for example, “*three letter word starting with X*”). The main challenge here is that there are many logically equivalent regular expressions, some aligning better to the natural language than others. Kushman and Barzilay¹⁷ uses semantic unification to test for logical form equivalence and obtains 65.6% accuracy.

The Free917 dataset¹¹ consists of 917 examples of question-logical form pairs that can be answered via Freebase, for example, “*how many works did mozart dedicate to joseph haydn?*” The questions are logically less complex than those in the semantic parsing datasets earlier, but introduces the new challenge of scaling up to many more predicates (but is in practice manageable by assuming perfect named entity resolution and leveraging the strong type constraints in Freebase). The state-of-the-art accuracy is 68% accuracy.¹⁸

WebQuestions⁴ is another dataset on Freebase consisting of 5810 question-answer pairs (no logical forms) such as “*what do australians call their money?*” Like Free917, the questions are not very compositional, but unlike Free917, they are real questions asked

by people on the Web independent from Freebase, so they are more realistic and more varied. Because the answers are required to come from a single Freebase page, a noticeable fraction of the answers are imperfect. The current state of the art is 52.5%.³⁵

The goal of WikiTableQuestions²⁶ is to extend question answering beyond Freebase to HTML tables on Wikipedia, which are semi-structured. The dataset consists of 22033 question-table-answer triples (for example, “*how many runners took 2 minutes at the most to run 1500 meters?*”), where each question can be answered by aggregating information across the table. At test time, we are given new tables, so methods must learn how to generalize to new predicates. The result on this new dataset is 37.1%.

Wang et al.³⁰ proposed a new recipe for quickly using crowdsourcing to generate new compositional semantic parsing datasets consisting of question-logical form pairs. Using this recipe, they created eight new datasets in small domains consisting of 12,602 total question-answer pairs, and achieved an average accuracy across datasets of 58.8%.

Chen and Mooney¹² introduced a dataset of 706 navigation instructions (for example, “*facing the lamp go until you reach a chair*”) in a simple grid world. Each instruction sequence contains multiple sentences with various imperative and context-dependent constructions not found in previous datasets. Artzi and Zettlemoyer² obtained 65.3% sentence-level accuracy.

Discussion

We have presented a semantic parsing framework for the problem of natural language understanding. Going forward, the two big questions are: How to represent the semantics of language and what supervision to use to learn the semantics from.

Alternative semantic representations.

One of the main difficulties with semantic parsing is the divergence between the structure of the natural language and the logical forms—purely compositional semantics will not work. This has led to some efforts to introduce an intermediate layer between utterances and logical forms. One idea is to use general paraphrasing models to map input utterances to the “canonical utterances” of logical forms.^{5, 30} This reduces semantic parsing

to a text-only problem for which there is much more data and resources.

One could also use domain-general logical forms that capture the basic predicate-argument structures of sentences.¹⁸ Abstract meaning representation (AMR)³ is one popular representation backed by an extension linguistic annotation effort. Multiple AMR parsers have been developed, including one based on CCG.¹ While these representations offer broad coverage, solving downstream understanding tasks still requires additional work (for example, inferring that “*population of X*” is synonymous with “*number of people living in X*”). In contrast, *executable* semantic parsing operates on the full pipeline from utterance to task output, but compromises on domain-generalizability. This is partially inevitable, as any understanding task requires some domain-specific knowledge or grounding. Designing the best general representation that supports many downstream tasks remains an open challenge.

Alternative supervision. Much of the progress in semantic parsing has been due to being able to learn from weaker supervision. In the framework we presented, this supervision is the desired actions y (for example, answers to questions). One can use a large corpus of text to exploit even weaker supervision.^{16,27} More generally, one can think about language interpretation in a reinforcement learning setting,⁹ where an agent presented with an utterance in some context performs some action, and receives a corresponding reward signal. This framework highlights the importance of context-dependence in language interpretation.^{2,39}

Due to their empirical success, there has been a recent surge of interest in using recurrent neural networks and their extensions for solving NLP tasks such as machine translation and question answering.^{31,35} These methods share the same spirit of end-to-end utterance-to-behavior learning as executable semantic parsing, but they do not explicitly separate parsing from execution. This makes them architecturally simpler than semantic parsers, but they are more data hungry and it is unclear whether they can learn to perform complex logical reasoning in a generalizable way. Nonetheless, it is quite likely that these

methods will play an important role in the story of language understanding.

Outlook. This is an exciting time for semantic parsing and natural language understanding. As natural language interfaces (for example, Siri) become more ubiquitous, the demand for deep understanding will continue to grow. At the same time, there is a fresh wave of ambition pushing the limits of what can be machine learnable. The confluence of these two factors will likely generate both end-user impact as well as new insights into the nature of language and learning. **C**

References

- Artzi, Y., Zettlemoyer, K.L.L. Broad-coverage CCG semantic parsing with AMR. In *Empirical Methods in Natural Language Processing* (2015).
- Artzi, Y., Zettlemoyer, L. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Trans. Assoc. Computat. Linguist.* 1 (2013): 49–62.
- Banarescu, L., Cai, C.B.S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N. Abstract meaning representation for sembanking. In *7th Linguistic Annotation Workshop and Interoperability with Discourse* (2013).
- Berant, J., Chou, A., Frostig, R., Liang, P. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing* (2013).
- Berant, J., Liang, P. Semantic parsing via paraphrasing. In *Association for Computational Linguistics* (2014).
- Berant, J., Liang, P. Imitation learning of agenda-based semantic parsers. *Trans. Assoc. Comput. Linguist.* (2015).
- Blackburn, P., Bos, J. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. CSLI Publishers, 2005.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In *International Conference on Management of Data (SIGMOD)* (2008), 1247–1250.
- Branavan, S., Chen, H., Zettlemoyer, L.S., Barzilay, R. Reinforcement learning for mapping instructions to actions. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing* (2009), 82–90.
- Brill, E., Dumais, S., Banko, M. An analysis of the AskMSR question-answering system. In *Association for Computational Linguistics* (2002), 257–264.
- Cai, Q., Yates, A. Large-scale semantic parsing via schema matching and lexicon extension. In *Association for Computational Linguistics* (2013).
- Chen, D.L., Mooney, R.J. Learning to interpret natural language navigation instructions from observations. In *Association for the Advancement of Artificial Intelligence* (2011), 859–865.
- Clarke, J., Goldwasser, D., Chang, M., Roth, D. Driving semantic parsing from the world's response. In *Computational Natural Language Learning* (2010), 18–27.
- Duchi, J., Hazan, E., Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Conference on Learning Theory* (2010).
- Krishnamurthy, J., Kollar, T. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Trans. Assoc. Computat. Linguist.* 1 (2013), 193–206.
- Krishnamurthy, J., Mitchell, T. Weakly supervised training of semantic parsers. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2012), 754–765.
- Kushman, N., Barzilay, R. Using semantic unification to generate regular expressions from natural language. In *Human Language Technology and North American Association for Computational Linguistics* (2013), 826–836.
- Kwiatkowski, T., Choi, E., Artzi, Y., Zettlemoyer, L. Scaling semantic parsers with on-the-fly ontology matching. In *Empirical Methods in Natural Language Processing* (2013).
- Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., Steedman, M. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Empirical Methods in Natural Language Processing* (2010), 1223–1233.
- Liang, P. Lambda dependency-based compositional semantics. *arXiv*, 2013.
- Liang, P., Jordan, M.I., Klein, D. Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)* (2011), 590–599.
- Liang, P., Potts, C. Bringing machine learning and compositional semantics together. *Annu. Rev. Linguist.* 1, 1 (2015), 355–376.
- Matuszek, C., FitzGerald, N., Zettlemoyer, L., Bo, L., Fox, D. A joint model of language and perception for grounded attribute learning. In *International Conference on Machine Learning* (2012), 1671–1678.
- Miller, S., Stallard, D., Bobrow, R., Schwartz, R. A fully statistical approach to natural language interfaces. In *Association for Computational Linguistics* (1996), 55–61.
- Montague, R. The proper treatment of quantification in ordinary English. In *Approaches to Natural Language* (1973), 221–242.
- Pasupat, P., Liang, P. Compositional semantic parsing on semi-structured tables. In *Association for Computational Linguistics* (2015).
- Reddy, S., Lapata, M., Steedman, M. Large-scale semantic parsing without question-answer pairs. *Trans. Assoc. Comput. Linguist.* 2, 10 (2014), 377–392.
- Steedman, M. *The Syntactic Process*. MIT Press, 2000.
- Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S.J., Roy, N. Understanding natural language commands for robotic navigation and mobile manipulation. In *Association for the Advancement of Artificial Intelligence* (2011).
- Wang, Y., Berant, J., Liang, P. Building a semantic parser overnight. In *Association for Computational Linguistics* (2015).
- Weston, J., Chopra, S., Bordes, A. Memory networks. In *International Conference on Learning Representations* (2015).
- Winograd, T. *Understanding Natural Language*. Academic Press, 1972.
- Wong, Y.W., Mooney, R.J. Learning synchronous grammars for semantic parsing with lambda calculus. In *Association for Computational Linguistics* (2007), 960–967.
- Woods, W.A., Kaplan, R.M., Webber, B.N. The lunar sciences natural language information system: Final report. Technical report, BBN Report 2378, Bolt Beranek and Newman Inc., 1972.
- Yih, W., Chang, M., He, X., Gao, J. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Association for Computational Linguistics* (2015).
- Zelle, M., Mooney, R.J. Learning to parse database queries using inductive logic programming. In *Association for the Advancement of Artificial Intelligence* (1996), 1050–1055.
- Zettlemoyer, L.S., Collins, M. Learning to map sentences to logical form: Structured classification with probabilistic categorical grammars. In *Uncertainty in Artificial Intelligence* (2005), 658–666.
- Zettlemoyer, L.S., Collins, M. Online learning of relaxed CCG grammars for parsing to logical form. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2007), 678–687.
- Zettlemoyer, L.S., Collins, M. Learning context-dependent mappings from sentences to logical form. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing* (2009).
- Zhao, K., Huang, L. Type-driven incremental semantic parsing with polymorphism. In *North American Association for Computational Linguistics* (2015).

Percy Liang (pliang@cs.stanford.edu) is an assistant professor of computer science at Stanford University, Stanford, CA.

Copyright held by author.
Publication rights licensed to ACM. \$15.00.

research highlights

P. 78

Technical Perspective **The Dawn of Computational Light Transport**

By Kyros Kutulakos

P. 79

Imaging the Propagation of Light through Scenes at Picosecond Resolution

By Andreas Velten, Di Wu, Belen Masia, Adrian Jarabo,
Christopher Barsi, Chinmaya Joshi, Everett Lawson,
Moungi Bawendi, Diego Gutierrez, and Ramesh Raskar

P. 87

Technical Perspective **Jupiter Rising**

By Andrew W. Moore

P. 88

Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network

By Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson,
Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai,
Bob Felderman, Paulie Germano, Anand Kanagala, Hong Liu,
Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer,
Urs Hölzle, Stephen Stuart, and Amin Vahdat

Technical Perspective

The Dawn of Computational Light Transport

By Kyros Kutulakos

IT IS EASY to forget, when casually observing our surroundings, that the speed of light is finite. Light travels so quickly that even though it may scatter, refract, or bounce many times on nearby surfaces before reaching our eyes, these events are spaced just trillionths of a second apart—too fast for any conventional camera to resolve, and certainly too fast for our own visual system to perceive as anything but instantaneous.

But are these individual light transport events really beyond the realm of direct visual observation? What would the world look like if we had a chance to observe it with a trillion-frame-per-second video camera? And what insights might one gain from such observations?

The following paper by Velten et al. represents an audacious attempt to answer these questions for the very first time for general objects. Much like a microscope can zoom to a tiny area on a specimen, the authors describe an imaging system that can zoom to a tiny interval of time—just a nanosecond across—and record a 512-frame video that spans it. Their online videos show wavefronts of light traveling across a variety of natural and man-made objects and are simply stunning (<http://web.media.mit.edu/~raskar/trillionfps/>; <http://giga.cps.unizar.es/~ajarabo/pubs/femtoSIG2013/>).

The authors' approach was to take a page out of Doc Edgerton's playbook. In the 1920s Edgerton, inventor of the electronic flash and famous for his photos of speeding bullets frozen in time, used a stroboscope to inspect rapidly rotating motors. By flashing a bright light very briefly in sync with the motor's rotation, he could make the motor look like it was standing still (and therefore easy to photograph). Here the authors use the same technique at much, much shorter time-scales to make light itself appear still. How short? The flash lasts one twenty-

thousandth of a nanosecond; a frame of video is exposed for about 40 times as long in sync with the flash; and this is repeated half a billion times at intervals of about 13ns in order to collect enough light. Simple!

The devil, of course, is in the details. Such brief flashes can only come from a femtosecond laser. Also, cameras with picosecond-scale frame rates do not actually exist. The closest sensor the authors could find is a streak camera—a 1D photo detector used primarily in chemistry and physics to take *x*-versus-time light measurements at picosecond resolutions. To capture 2D video, the sensor is panned across the field of view and makes repeated acquisitions, building the video row by row. Additional processing then ensures the final video is consistent with the arrow of time.

As with any “first” attempt to solve a problem, it is relatively easy to criticize. The equipment costs hundreds of thousands of dollars; it takes hours to record one video; the scene must remain perfectly still; and its surface geometry must be captured laboriously with a 3D digitizer.

These issues, however, are beside the point. The paper's principal contribution is the highest-order bit: that light transport through an everyday scene can be observed directly by ultra-fast imaging. With that settled, cheaper and faster solutions will surely follow (and they have—see <http://www.cs.ubc.ca/labs/imager/tr/2013/TransientPMD/>).

On a broader level, the paper marks the maturation of computational light transport—an emerging family of techniques that use lights, sensors, and computation to reveal and interpret the flow of light in our everyday world. The topic has been brewing for well over two decades in the research communities of computer vision and computer graphics but required the

confluence of several factors to finally come into its own. This includes new sensors; programmable light sources; powerful computing engines; new algorithmic tools for data-driven analysis of light transport; and growing interdisciplinary ties to optics.

Will these techniques ultimately prove useful? There are several reasons to be optimistic. As a person living in Canada, I can say there is a great need for imaging techniques that work well in the presence of complex light transport, and scattering in particular (fog, snow, and so on). Scattering is also the barrier to imaging deep under the skin and through other biological tissues and causes major accuracy problems in modern depth cameras and 3D scanners. While I don't expect vision-based driving on snowy roads and through raging snowstorms anytime soon, transport-robust 3D imaging techniques are already here (see DOIs 10.1145/2735702 and 10.1145/2766897), never-seen before abilities such as looking around corners are a reality (see DOI 10.1038/ncomms1747), and there is great potential for synergies with scientific imaging (see DOIs 10.1038/NPHYS3373 and 10.1145/2766928).

But perhaps the biggest reason for optimism is this: ultimately, computer vision algorithms can only be as powerful as the sensor data they take as input. Given the tremendous strides the field has made recently in tasks such as object recognition and navigation from conventional images, one can only speculate how much more could be accomplished from light transport data. So, read the paper and consider the possibilities. 

Kyros Kutulakos (kyros@cs.utoronto.ca) is a professor of computer science at the University of Toronto, Canada.

Copyright held by author.

Imaging the Propagation of Light through Scenes at Picosecond Resolution

By Andreas Velten, Di Wu, Belen Masia, Adrian Jarabo, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Mounig Bawendi, Diego Gutierrez, and Ramesh Raskar

Abstract

We present a novel imaging technique, which we call *femto-photography*, to capture and visualize the propagation of light through table-top scenes with an effective exposure time of 1.85 ps per frame. This is equivalent to a resolution of about one half trillion frames per second; between frames, light travels approximately just 0.5 mm. Since cameras with such extreme shutter speed obviously do not exist, we first re-purpose modern imaging hardware to record an ensemble average of repeatable events that are synchronized to a streak sensor, in which the time of arrival of light from the scene is coded in one of the sensor's spatial dimensions. We then introduce reconstruction methods that allow us to visualize the propagation of femtosecond light pulses through the scenes. Given this fast resolution and the finite speed of light, we observe that the camera does not necessarily capture the events in the same order as they occur in reality: we thus introduce the notion of *time-unwarping* between the camera's and the world's space-time coordinate systems, to take this into account. We apply our femto-photography technique to visualizations of very different scenes, which allow us to observe the rich dynamics of time-resolved light transport effects, including scattering, specular reflections, diffuse interreflections, diffraction, caustics, and subsurface scattering. Our work has potential applications in artistic, educational, and scientific visualizations; industrial imaging to analyze material properties; and medical imaging to reconstruct subsurface elements. In addition, our time-resolved technique has already motivated new forms of computational photography, as well as novel algorithms for the analysis and synthesis of light transport.

1. INTRODUCTION

The way in which light travels through a scene—the paths it follows, how its intensity evolves over time—is an incommensurate source of information on the nature of such scene. From it, we can for instance obtain the geometry of the scene (even of those parts which are not visible to the camera),^{6, 11, 17} the reflectance of the objects present,¹² or even derive other material properties.²⁰ As such, it can play a very important role in a variety of fields, including computer graphics, computer vision, and scientific imaging in general, and have applications in medicine, defense, or industrial processes, to name a few. However, traditionally,

light has been assumed to travel instantaneously through a scene (its speed assumed to be infinite), because conventional imaging hardware is very slow compared to the speed of light. Consequently, any information encoded in the time delays of light propagation is lost, and disambiguating light transport becomes an arduous, often impossible, task. In the past years, the joint design of novel optical hardware and smart computation (i.e., computational photography), has expanded the way we capture, analyze, and understand visual information; however, speed-of-light propagation has been largely unexplored at the macroscopic scale, due to the impossibility of capturing such information.

In this paper, we present a novel ultrafast imaging technique, which we term *femto-photography*, which allows us to capture movies of a scene at an effective frame rate of one half trillion frames per second. With this extremely high temporal resolution, we can obtain movies in which light travels less than a millimeter per frame (the duration of one frame is ca. 2 ps, i.e., 10^{-12} s), enabling us to capture light in motion as it travels through a scene. This allows us to see, for instance, a light pulse scattering inside a plastic bottle, or image formation in a mirror, or beam diffraction by a grating, as a function of time. Further, the captured, time-resolved data has the potential to allow us to determine light transport in complex scenes.

Our system makes use of femtosecond laser illumination, picosecond-accurate detectors, and mathematical reconstruction techniques to obtain and correctly visualize the final movie. In it, for each pixel of each frame, we have an intensity profile as a function of time, with a resolution of 1.85 ps.

Developing such time-resolved system is a challenging problem for several reasons. Our contribution in this work is to address these challenges and create the first prototype capable of capturing such temporal resolutions in macroscopic scenes. These challenges include: *First*, brute-force time exposures under 2 ps yield an impractical signal-to-noise (SNR) ratio, since during each exposure only a few photons will reach the sensor. To improve the SNR of the acquisition process, we exploit the statistical similarity of periodic light transport events; this allows recording multiple, ultrashort exposure times of one-dimensional

The original version of this paper is entitled “Femto-Photography: Capturing and Visualizing the Propagation of Light” and was published in *ACM Trans. Graph.*, 32(4), SIGGRAPH 2013.

A. Velten: The work was performed while at the MIT Media Lab.

views. These views are one-dimensional due to the *second* challenge we face: Suitable cameras to record 2D image sequences at this time resolution do not exist due to sensor bandwidth limitations. To solve this, we introduce a novel hardware implementation to sweep the exposures across a vertical field of view, to build 3D space-time data volumes. *Third*, comprehensible visualization of the captured time-resolved data is non-trivial: It is a novel type of data that we are not accustomed to seeing, and some observed effects can be counter-intuitive. We therefore create techniques for comprehensible visualization of this time-resolved data, including movies showing the dynamics of real-world light transport phenomena and the notion of peak-time, which partially overcomes the low-frequency appearance of integrated global light transport. *Finally*, direct measurements of events at this speed appear warped in space-time, because the finite speed of light implies that the recorded light propagation delay depends on camera position relative to the scene. To correct for this, and visualize events in their correct sequence, we introduce a *time-unwarping* technique, which accounts for the distortions in captured time-resolved information due to the finite speed of light.

In the following, we describe these contributions in detail. We explain our complete hardware, calibration, and data processing and visualization pipeline, and demonstrate its potential by acquiring time-resolved movies of significant light transport effects, including scattering, diffraction, or multiple diffuse interreflections. We further discuss possible applications of this new imaging modality, and the relevance of this work, not only in imaging, but also in areas such as biomedical research or astronomy.

2. RELATED WORK

2.1. Ultrafast devices

Repetitive illumination techniques used in incoherent LiDAR use cameras with typical exposure times on the order of hundreds of picoseconds, two orders of magnitude slower than our system.² The fastest 2D continuous, real-time monochromatic camera operates at hundreds of nanoseconds per frame, with a spatial resolution of 200×200 pixels; this is less than one-third of what we achieve in this paper.³ Avalanche photodetectors (APD) arrays can reach temporal resolutions of several tens of picoseconds if they are used in a photon starved regime where only a single photon hits a detector within a time window of tens of nanoseconds.¹ Liquid nonlinear shutters actuated with powerful laser pulses have been used to capture single analog frames imaging light pulses at picosecond time resolution. Other sensors that use a coherent phase relation between the illumination and the detected light, such as optical coherence tomography (OCT), coherent LiDAR, light in flight holography, or white light interferometry, achieve femtosecond resolutions; however, they require light to maintain coherence (i.e., wave interference effects) during light transport, and are therefore unsuitable for indirect illumination, in which diffuse reflections remove coherence from the light. Last, simple streak sensors capture incoherent light at picosecond to nanosecond speeds, but are limited to a line or low resolution (20×20) square field of view.¹⁵

In contrast, our system is capable of recording and reconstructing space-time world information of incoherent light propagation in free-space, table-top scenes, at a resolution of up to 672×1000 pixels and under 2 ps per frame. The varied range and complexity of the scenes we can capture allows us to visualize the *dynamics* of global illumination effects, such as scattering, specular reflections, inter-reflections, subsurface scattering, caustics, and diffraction.

2.2. Time-resolved imaging

Recent advances in time-resolved imaging have been exploited to recover geometry and motion around corners,^{4, 10, 14, 16, 17} as well as albedo from a single view point.¹² However, they all share some fundamental limitations (such as capturing only third-bounce light) that make them unsuitable for capturing videos of light in motion. The principles we develop in this paper were first demonstrated by the authors in two previous publications^{18, 19}; this has given rise to alternative, inexpensive PMD-based approaches (e.g., Ref.⁵), although the achieved temporal resolution is in the order of nanoseconds (instead of picoseconds). Wu et al.²¹ present a rigorous analysis of transient light transport in the frequency domain, and show how it can be applied to build a bare-sensor ultrafast imaging system. Last, two recent publications provide valuable tools for time-resolved imaging: Wu and colleagues²⁰ separate direct and global illumination components from time-resolved data captured with the system we describe in this paper, by analyzing the time profile of each pixel, and demonstrate a number of applications; whereas Jarabo et al.⁷ present a framework for the efficient *simulation* of light-in-flight movies, which enables analysis-by-synthesis approaches for the analysis of transient light transport.

3. CAPTURING SPACE-TIME PLANES

We capture time scales orders of magnitude faster than the exposure times of conventional cameras, in which photons reaching the sensor at different times are integrated into a single value, making it impossible to observe ultrafast optical phenomena. The system described in this paper has an effective exposure time down to 1.85 ps; since light travels at 0.3 mm/ps, light travels approximately 0.5 mm between frames in our reconstructed movies.

3.1. System

An ultrafast setup must overcome several difficulties in order to accurately measure a high-resolution (both in space and time) image. First, for an unamplified laser pulse, a single exposure time of less than 2 ps would not collect enough light, so the SNR would be unworkably low. As an example, for a table-top scene illuminated by a 100 W bulb, only about 1 photon on average would reach the sensor during a 2 ps open-shutter period. Second, because of the time scales involved, synchronization of the sensor and the illumination must be executed within picosecond precision. Third, standalone streak sensors sacrifice the vertical spatial dimension in order to code the time dimension, thus

www.hamamatsu.com/resources/pdf/sys/SHSS0006E_STREAK.pdf.

producing x-t images. As a consequence, their field of view is reduced to a single horizontal line of view of the scene.

We solve these problems with our ultrafast imaging system, outlined in Figure 2. The light source is a femtosecond Kerr lens mode-locked Ti:Sapphire laser, which emits 50-fs with a center wavelength of 795 nm, at a repetition rate of 75 MHz and average power of 500 mW. In order to see ultrafast events in a scene with macro-scaled objects, we focus the light with a lens onto a Lambertian diffuser, which then acts as a point light source and illuminates the entire scene with a spherically shaped pulse. Alternatively, if we want to observe pulse propagation itself, rather than the interactions with large objects, we direct the laser beam across the field of view of the camera through a scattering medium (see the *bottle* scene in Figure 1).

Because all the pulses are statistically identical, we can record the scattered light from many of them and integrate

the measurements to average out noise. The result is a signal with a high SNR. To synchronize the illumination with the streak sensor (Hamamatsu C5680), we split off a portion of the beam with a glass slide and direct it onto a fast photo-detector connected to the sensor, so that both detector and illumination operate synchronously (Figure 2a and b).

3.2. Capturing space-time planes

The streak sensor then captures an x-t image of a certain scanline (i.e., a line of pixels in the horizontal dimension) of the scene with a space-time resolution of 672×512 . The exact time resolution depends on the amplification of an internal sweep voltage signal applied to the streak sensor. With our hardware, it can be adjusted from 0.30 to 5.07 ps. Practically, we choose the fastest resolution that still allows for capture of the entire duration of the event. In the streak sensor, a photocathode converts incoming photons, arriving

Figure 1. What does the world look like at the speed of light? Our new computational photography technique allows us to visualize light in ultra-slow motion, as it travels and interacts with objects in table-top scenes. We capture photons with an effective temporal resolution of less than 2 ps per frame. Top row, left: a false color, single *streak image* from our sensor. Middle: time lapse visualization of the *bottle* scene, as directly reconstructed from sensor data. Right: time-unwarped visualization, taking into account the fact that the speed of light can no longer be considered infinite (see the main text for details). Bottom row: original scene through which a laser pulse propagates, followed by different frames of the complete reconstructed video. For this and other results in the paper, we refer the reader to the videos included in the project pages: femtocamera.info and <http://giga.cps.unizar.es/~ajarabo/pubs/femtoSIG2013/>.

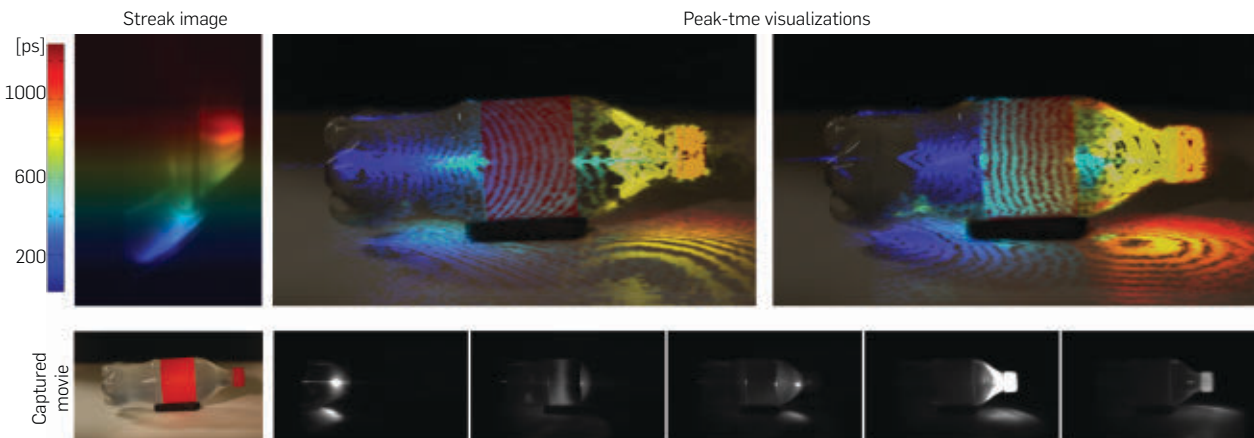
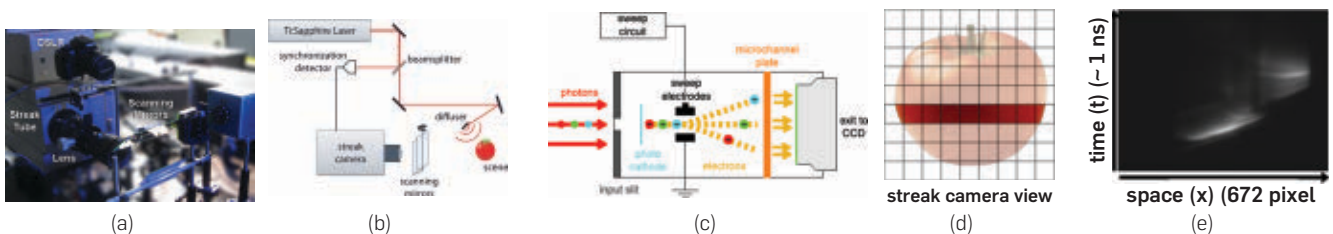


Figure 2. (a) Photograph of our ultrafast imaging system setup. The DSLR camera takes a conventional photo for comparison. (b) In order to capture a single 1D space-time photo, a laser beam strikes a diffuser, which converts the beam into a spherical energy front that illuminates the scene; a beamsplitter and a synchronization detector enable synchronization between the laser and the streak sensor. (c) After interacting with the scene, photons enter a horizontal slit in the camera and strike a photocathode, which generates electrons. These are deflected at different angles as they pass through a microchannel plate, by means of rapidly changing the voltage between the electrodes. The CCD records the horizontal position of each pulse and maps its arrival time to the vertical axis, depending on how much the electrons have been deflected. (d) We focus the streak sensor on a single narrow scanline of the scene. (e) Sample image taken by the streak sensor. The horizontal axis (672 pixels) records the photons' spatial locations in the acquired scanline, while the vertical axis (1 ns window in our implementation) codes their arrival time. Rotating the adjustable mirrors shown in (a) allows for scanning of the scene in the y-axis and generation of ultrafast 2D movies such as the one visualized in Figure 1 (b-d, credit: Greg Gbur).



from each spatial location in the scanline, into electrons. The streak sensor generates the x-t image by deflecting these electrons, according to the time of their arrival, to different positions along the t-dimension of the sensor (see Figure 2b and c). This is achieved by means of rapidly changing the sweep voltage between the electrodes in the sensor. For each horizontal scanline, the camera records a scene illuminated by the pulse and averages the light scattered by 4.5×10^8 pulses (see Figure 2d and e).

3.3. Performance validation

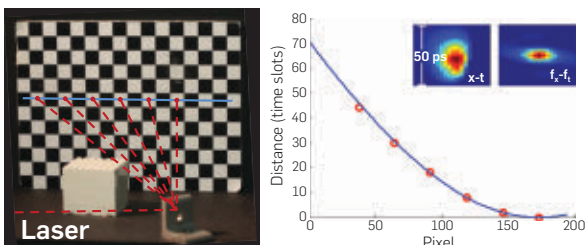
To characterize the streak sensor, we compare sensor measurements with known geometry and verify the linearity, reproducibility, and calibration of the time measurements. To do this, we first capture a streak image of a scanline of a simple scene: a plane being illuminated by the laser after hitting the diffuser (see Figure 3, left). Then, by using a Faro digitizer arm, we obtain the ground truth geometry of the points along that plane and of the point of the diffuser hit by the laser; this allows us to compute the total travel time per path (diffuser-plane-streak sensor) for each pixel in the scanline. We then compare the travel time captured by our streak sensor with the real travel time computed from the known geometry. The graph in Figure 3 (right) shows agreement between the measurement and calculation.

4. CAPTURING SPACE-TIME VOLUMES

Although the synchronized, pulsed measurements overcome SNR issues, the streak sensor still provides only a one-dimensional movie. Extension to two dimensions requires unfeasible bandwidths: a typical dimension is roughly 10^3 pixels, so a three-dimensional data cube has 10^9 elements. Recording such a large quantity in a 10^{-9} s (1 ns) time window requires a bandwidth of 10^{18} byte/s, far beyond typical available bandwidths.

We solve this acquisition problem by again utilizing the synchronized repeatability of the hardware: A mirror-scanning system (two 9 cm $\times$ 13 cm mirrors, see Figure 2a) rotates the camera's center of projection, so that it records horizontal slices of a scene sequentially. We use a computer-controlled, 1-rpm servo motor to rotate one of the mirrors and consequently scan the field of view vertically. The scenes are about 25 cm wide and

Figure 3. Performance validation of our system. Left: Measurement setup used to validate the data. We use a single streak image representing a line of the scene and consider the centers of the white patches because they are easily identified in the data. Right: Graph showing pixel position versus total path travel time captured by the streak sensor (red) and calculated from measurements of the checkerboard plane position with a Faro digitizer arm (blue). Inset: PSF, and its Fourier transform, of our system.



placed about 1 m from the camera. With high gear ratios (up to 1:1000), the continuous rotation of the mirror is slow enough to allow the camera to record each line for about 6 s, requiring about one hour for 600 lines (our video resolution). We generally capture extra lines, above and below the scene (up to 1000 lines), and then crop them to match the aspect ratio of the physical scenes before the movie is reconstructed.

These resulting images are combined into one matrix, M_{ijk} , where $i = 1, \dots, 672$ and $k = 1, \dots, 512$ are the dimensions of the individual x-t streak images, and $j = 1, \dots, 1000$ addresses the second spatial dimension y. For a given time instant k , the submatrix N_{ij} contains a two-dimensional image of the scene with a resolution of 672×1000 pixels, exposed for as short as 1.85 ps. Combining the x-t slices of the scene for each scanline yields a 3D x-y-t data volume, as shown in Figure 4 (left). An x-y slice represents one frame of the final movie, as shown in Figure 4 (right).

5. DEPICTING ULTRAFAST VIDEOS IN 2D

We have explored several ways to visualize the information contained in the captured x-y-t data cube in an intuitive way. First, contiguous N_{ij} slices can be played as the frames of a movie. Figure 1 (bottom row) shows a captured scene (*bottle*) along with several representative N_{ij} frames (effects are described for various scenes in Section 7). However, understanding all the phenomena shown in a video is not a trivial task, and movies composed of x-y frames such as the ones shown in Figure 8 may be hard to interpret. Merging a static photograph of the scene from approximately the same point of view with the N_{ij} slices aids in the understanding of light transport in the scenes (see movies at the project pages). Although straightforward to implement, the high dynamic range of the streak data requires a nonlinear intensity transformation to extract subtle optical effects in the presence of high intensity reflections. We employ a logarithmic transformation to this end.

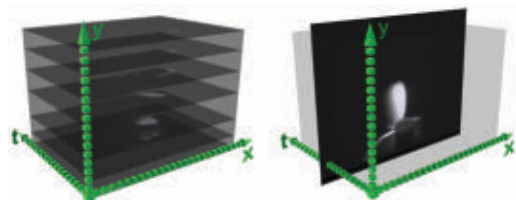
We have also explored single-image methods for intuitive visualization of full space-time propagation, such as the color-coding in Figure 1 (right), which we describe next.

5.1. Integral photo fusion

By integrating all the frames in novel ways, we can visualize and highlight different aspects of the light flow in one photo. Our photo fusion results are calculated as $N_{ij} = \sum w_k M_{ijk}$, $\{k = 1, \dots, 512\}$, where w_k is a weighting factor determined by the particular fusion method. We have tested several different

<http://www.faro.com>.

Figure 4. Left: Reconstructed x-y-t data volume by stacking individual x-t images (captured with the scanning mirrors). Right: An x-y slice of the data cube represents one frame of the final movie.



methods, of which two were found to yield the most intuitive results: the first one is *full fusion*, where $w_k = 1$ for all k . Summing all frames of the movie provides something resembling a black and white photograph of the scene illuminated by the laser, while showing time-resolved light transport effects. An example is shown in Figure 5 (left) for the *alien* scene (more information about the scene is given in Section 7). A second technique, *rainbow fusion*, takes the fusion result and assigns a different RGB color to each frame, effectively color-coding the temporal dimension. An example is shown in Figure 5 (middle).

5.2. Peak time images

The inherent integration in fusion methods, though often useful, can fail to reveal the most complex or subtle behavior of light. As an alternative, we propose peak time images, which illustrate the time evolution of the *maximum* intensity in each frame. For each spatial position (i, j) in the x-y-t volume, we find the peak intensity along the time dimension, and keep information within two time units to each side of the peak. All other values in the streak image are set to zero, yielding a more sparse space-time volume. We then color-code time and sum up the x-y frames in this new sparse volume, in the same manner as in the *rainbow fusion* case but use only every 20th frame in the sum to create black lines between the equi-time paths, or isochrones. This results in a map of the propagation of maximum intensity contours, which we term *peak time image*. These color-coded isochronous lines can be thought of intuitively as propagating energy fronts. Figure 5 (right) shows the peak time image for the *alien* scene and Figure 1 (top, middle) shows the captured data for the *bottle* scene depicted using this visualization method. As explained in the next section, this visualization of the bottle scene reveals significant light transport phenomena that could not be seen with the rainbow fusion visualization.

6. TIME UNWARPING

Visualization of the captured movies (Sections 5 and 7) reveals results that are counter-intuitive to theoretical and established knowledge of light transport. Figure 1 (top, middle) shows a peak time visualization of the *bottle* scene, where several abnormal light transport effects can be observed: (1) the caustics on the floor, which propagate towards the bottle, instead of away from it; (2) the curved spherical energy fronts in the label area, which should be rectilinear as seen from the camera; and (3) the pulse itself being located behind these energy fronts, when it would

need to precede them. These are due to the fact that usually light propagation is assumed to be infinitely fast, so that events in world space are assumed to be detected simultaneously in camera space. In our ultrafast photography setup; however, this assumption no longer holds, and the finite speed of light becomes a factor: we must now take into account the time delay between the occurrence of an event and its detection by the camera sensor.

We therefore need to consider two different time frames, namely *world time* (when events happen) and *camera time* (when events are detected). This duality of time frames is explained in Figure 6: light from a source hits a surface first at point $P_1 = (i_1, j_1)$ (with (i, j) being the x-y pixel coordinates of a scene point in the x-y-t data cube), then at the farther point $P_2 = (i_2, j_2)$, but the reflected light is captured in the reverse order by the sensor, due to different total path lengths ($z_1 + d_1 > z_2 + d_2$). Generally, this is due to the fact that, for light to arrive at a given time instant t_0 , all the rays from the source, to the wall, to the camera, must satisfy $z_i + d_i = ct_0$, so that isochrones are elliptical. Therefore, although objects closer to the source receive light earlier, they can still lie on a higher-valued (later-time) isochrone than farther ones.

In order to visualize all light transport events as they have occurred (not as the camera captured them), we transform the captured data from camera time to world time, a transformation which we term *time unwarping*. Mathematically, for a scene point $P = (i, j)$, we apply the following transformation:

$$t'_{ij} = t_{ij} + \frac{z_{ij}}{c/\eta} \quad (1)$$

where t'_{ij} and t_{ij} represent camera and world times respectively, c is the speed of light in vacuum, η the index of refraction of the medium, and z_{ij} is the distance from point P to the camera. For our table-top scenes, we measure this distance with a Faro digitizer arm, although it could be obtained from the data and the known position of the diffuser, as the problem is analogous to that of bi-static LiDAR. We can thus define light travel time from each point (i, j) in the scene to the camera as $\Delta t_{ij} = t'_{ij} - t_{ij} = z_{ij} / (c/\eta)$. Then, time unwarping effectively corresponds to offsetting data in the x-y-t volume along the time dimension, according to the value of Δt_{ij} for each (i, j) point, as shown in Figure 7.

Figure 6. Understanding reversal of events in captured videos.

Left: Pulsed light scatters from a source, strikes a surface (e.g., at P_1 and P_2), and is then recorded by a sensor. Time taken by light to travel distances $z_1 + d_1$ and $z_2 + d_2$ is responsible for the existence of two different time frames and the need of computational correction to visualize the captured data in the world time frame. **Right:** Light appears to be propagating from P_2 to P_1 in camera time (before unwarping), and from P_1 to P_2 in world time, once time-unwarped. Extended, planar surfaces will intersect constant-time paths to produce either elliptical or circular fronts.

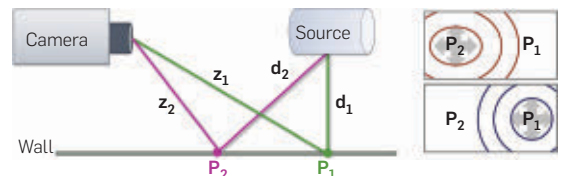
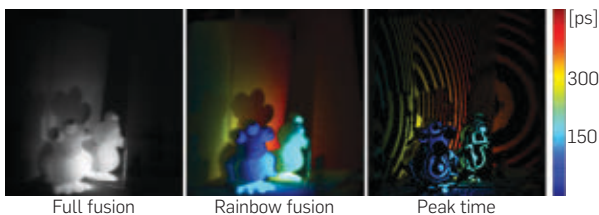


Figure 5. Three visualization methods for the *alien* scene. From left to right, more sophisticated methods provide more information and an easier interpretation of light transport in the scene.



In most of the scenes, we only have propagation of light through air, for which we take $\eta \approx 1$. For the *bottle* scene, we assume that the laser pulse travels along its longitudinal axis at the speed of light, and that only a single scattering event occurs in the liquid inside. We take $\eta = 1.33$ as the index of refraction of the liquid and ignore refraction at the bottle's surface. Our unoptimized Matlab code runs at about 0.1 s per

frame. A time-unwarped peak-time visualization of the whole of this scene is shown in Figure 1 (right). Notice how now the caustics originate from the bottle and propagate outward, energy fronts along the label are correctly depicted as straight lines, and the pulse precedes related phenomena, as expected.

7. CAPTURED SCENES

We have used our ultrafast photography setup to capture interesting light transport effects in different scenes. Figure 8 summarizes them, showing representative frames and peak time visualizations (please also refer to the full movies, which can be found in the project pages: femtocamera.info and <http://giga.cps.unizar.es/~ajarabo/pubs/femtoSIG2013/>). The exposure time for our scenes is between 1.85 ps for the *crystal* scene, and 5.07 ps for the *bottle* and *tank* scenes, which required imaging a longer time span for better visualization. Overall, observing light in such slow motion reveals both subtle and key aspects of light transport. We provide here brief descriptions of the light transport effects captured in the different scenes.

7.1. Bottle

This scene is shown in Figure 1 (bottom row), and has been used to introduce time-unwarping. A plastic bottle, filled

Figure 7. Time unwarping in 1D for a streak image (x - t slice). Left: Captured streak image; shifting the time profile down in the temporal dimension by Δt allows for the correction of path length delay to transform between time frames. Center: The graph shows, for each spatial location x , of the streak image, the amount Δt , that point has to be shifted in the time dimension of the streak image. Right: Resulting time-unwarped streak image.

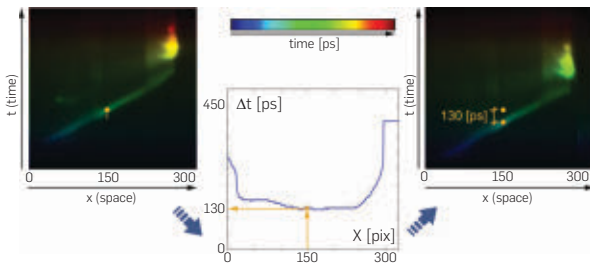
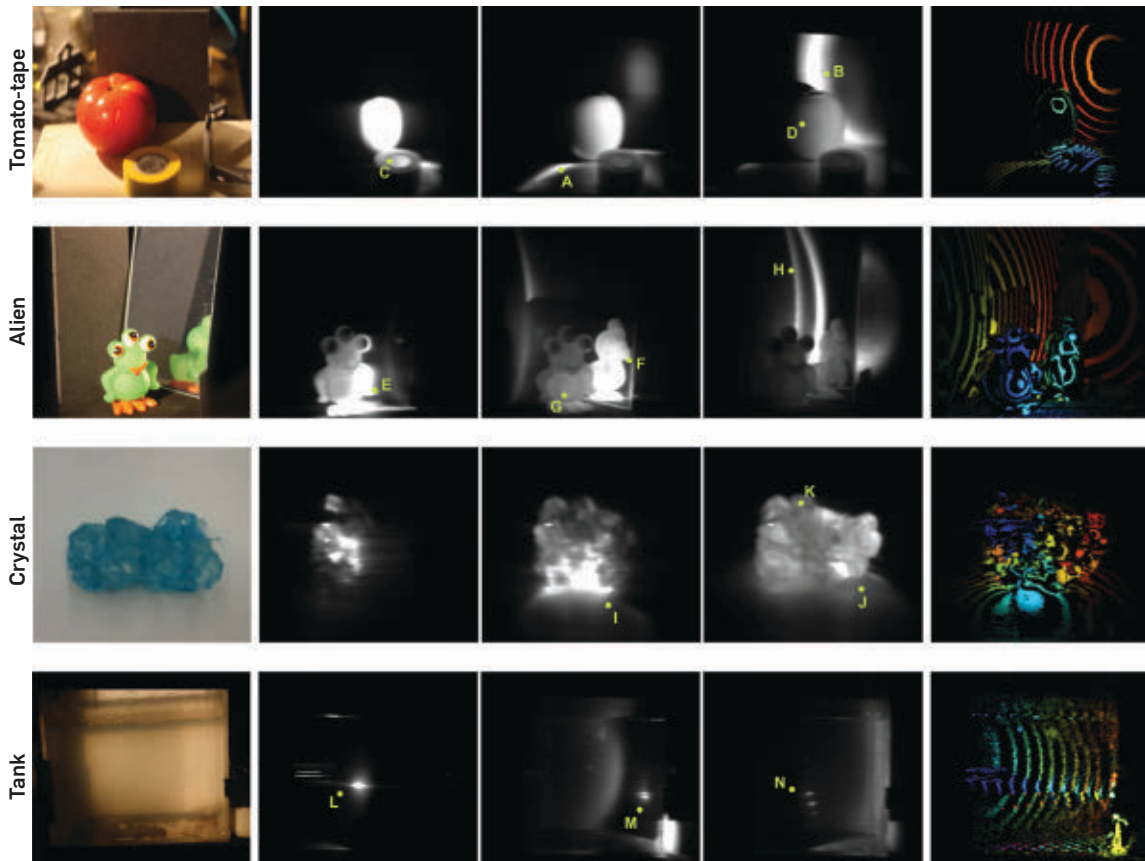


Figure 8. More scenes captured with our setup (refer to Figure 1 for the *bottle* scene). For each scene, from left to right: photograph of the scene (taken with a DSLR camera), a series of representative frames of the reconstructed movie, and peak time visualization of the data. The full movies can be found in the project pages: femtocamera.info and <http://giga.cps.unizar.es/~ajarabo/pubs/femtoSIG2013/>. Note that the viewpoint varies slightly between the DSLR and the streak sensor.



with water diluted with milk, is directly illuminated by the laser pulse, entering through the bottom of the bottle along its longitudinal axis. The pulse scatters inside the liquid; we can see the propagation of the wavefronts. The geometry of the bottle neck creates some interesting lens effects, making light look almost like a fluid. Most of the light is reflected back from the cap, while some is transmitted or trapped in subsurface scattering phenomena. Caustics are generated on the table.

7.2. Tomato-tape

This scene shows a tomato and a tape roll, with a wall behind them. The propagation of the spherical wavefront, after the laser pulse hits the diffuser, can be seen clearly as it intersects the floor and the back wall (A, B). The inside of the tape roll is out of the line of sight of the light source and is not directly illuminated. It is illuminated later, as indirect light scattered from the first wave reaches it (C). Shadows become visible only after the object has been illuminated. The more opaque tape darkens quickly after the light front has passed, while the tomato continues glowing for a longer time, indicative of stronger subsurface scattering (D).

7.3. Alien

A toy alien is positioned in front of a mirror and wall. Light interactions in this scene are extremely rich, due to the mirror, the multiple interreflections, and the subsurface scattering in the toy. The video shows how the reflection in the mirror is actually formed: direct light first reaches the toy, but the mirror is still completely dark (E); eventually light leaving the toy reaches the mirror, and the reflection is dynamically formed (F). Subsurface scattering is clearly present in the toy (G), while multiple direct and indirect interactions between wall and mirror can also be seen (H).

7.4. Crystal

A group of sugar crystals is directly illuminated by the laser from the left, acting as multiple lenses and creating caustics on the table (I). Part of the light refracted on the table is reflected back to the candy, creating secondary caustics on the table (J). Additionally, scattering events are visible within the crystals (K).

7.5. Tank

A reflective grating is placed at the right side of a tank filled with milk diluted in water. The grating is taken from a commercial spectrometer, and consists of an array of small, equally spaced rectangular mirrors. The grating is blazed: mirrors are tilted to concentrate maximum optical power in the first order diffraction for one wavelength. The pulse enters the scene from the left, travels through the tank (L), and strikes the grating. The grating reflects and diffracts the beam pulse (M). The different orders of the diffraction are visible traveling back through the tank (N). As the figure (and the captured movie) shows, most of the light reflected from the grating propagates at the blaze angle.

8. CONCLUSIONS AND OUTLOOK

Since the initial publication of this work numerous publications have advanced this field by improving numerical

models and introducing new, more accessible capture technology. Heide et al.⁵ and Kadambi et al.⁹ introduced methods of low resolution time of flight capture using inexpensive photonic mixer devices. These new devices have been used in different applications, for example to see around corners,^{6, 13} although their temporal resolution is still orders of magnitude lower than our system. Laurentzis and Velten¹¹ have recently demonstrated seeing around corners using intensified gated CCD cameras that are the state of the art in gated viewing applications and are available for example in military vehicles.

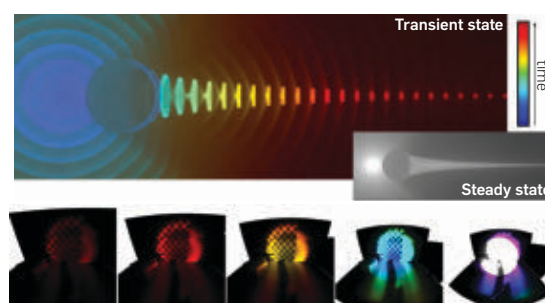
Optimization of the system hardware and software requires further advances in optics, material science, and compressive sensing. Beyond the potential in artistic and educational visualization, we hope our work will spawn new research in computer graphics and computation imaging techniques towards useful forward and inverse analysis of light interactions, which in turn will influence the rapidly emerging field of ultra fast imaging.

Future research involves investigating other ultrafast phenomena such as propagation of light in anisotropic media and photonic crystals, or novel applications in scientific visualization (to understand ultrafast processes), medicine (to image and reconstruct subsurface elements), material engineering (to analyze material properties), or quality control (to detect faults in structures). This may in turn introduce new challenges in the realm of computer graphics, to provide new insights via comprehensible simulations and new data structures to render transient light transport. For instance, our work has recently inspired a novel method for the efficient simulation on time-resolved light transport,⁷ while relativistic rendering techniques have been developed using our captured data, departing the common assumption of constant irradiance over the surfaces⁸ (Figure 9).

Acknowledgments

Belen Masia would like to acknowledge the support of the

Figure 9. Our work has inspired follow-up work in the field of computer graphics with the development of simulation frameworks departing the assumption of infinite speed of light (top, time-resolved rendering using our *peak time* visualization of a volumetric caustic),⁷ including the simulation of relativistic effects due to ultrafast camera motion (bottom, simulation of frames recorded by an accelerating camera in a scene captured using our system).⁸



Max Planck Center for Visual Computing and Communication. Diego Gutierrez would like to acknowledge the support of the Spanish Ministry of Science and Innovation (project Lightslice), the BBVA Foundation, and a Faculty Research Award from Google. 

References

- Charbon, E. Will avalanche photodiode arrays ever reach 1 megapixel? In *International Image Sensor Workshop* (Ogunquit, ME, 2007), 246–249.
- Colaco, A., Kirmani, A., Howland, G.A., Howell, J.C., Goyal, V.K. Compressive depth map acquisition using a single photon-counting detector: Parametric signal processing meets sparsity. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Providence, RI, June 2012), IEEE, 96–102.
- Goda, K., Tsia, K.K., Jalali, B. Serial time-encoded amplified imaging for real-time observation of fast dynamic phenomena. *Nature* 458 (2009), 1145–1149.
- Gupta, O., Willwacher, T., Velten, A., Veeraraghavan, A., Raskar, R. Reconstruction of hidden 3D shapes using diffuse reflections. *Opt. Expr.* 20 (2012), 19096–19108.
- Heide, F., Hullin, M.B., Gregson, J., Heidrich, W. Low-budget transient imaging using photonic mixer devices. *ACM Trans. Graph.* 32, 4 (2013), 45:1–45:10.
- Heide, F., Xiao, L., Heidrich, W., Hullin, M.B. Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. In *CVPR* (June 2014).
- Jarabo, A., Marco, J., Muñoz, A., Buisan, R., Jarosz, W., Gutierrez, D. A framework for transient rendering. *ACM Trans. Graph.* 33, 6 (2014), 177:1–177:10.
- Jarabo, A., Masia, B., Velten, A., Barsi, C., Raskar, R., Gutierrez, D. Relativistic effects for time-resolved light transport. *Comput. Graph. Forum* (2015) to appear. DOI: 10.1111/cgf.12604.
- Kadambi, A., Whyte, R., Bhandari, A., Streeter, L., Barsi, C., Dorrington, A., Raskar, R. Coded time of flight cameras: Sparse deconvolution to address multipath interference and recover time profiles. *ACM Trans. Graph.* 32, 6 (2013), 167:1–167:10.
- Kirmani, A., Hutchison, T., Davis, J., Raskar, R. Looking around the corner using ultrafast transient imaging. *Int. J. Comp. Vision* 95, 1 (2011), 13–28.
- Laurenzis, M., Velten, A. Nonline-of-sight laser gated viewing of scattered photons. *Opt. Eng.* 53, 2 (2014), 023102–023102.
- Naik, N., Zhao, S., Velten, A., Raskar, R., Bala, K. Single view reflectance capture using multiplexed scattering and TOF imaging. *ACM Trans. Graph.* 30 (2011), 171:1–171:10.
- O'Toole, M., Heide, F., Xiao, L., Hullin, M.B., Heidrich, W., Kutulakos, K.N. Temporal frequency probing for 5D transient analysis of global light transport. *ACM Trans. Graph.* 33, 4 (2014), 87:1–87:11.
- Pandharkar, R., Velten, A., Bardagaj, A., Bawendi, M., Raskar, R. Estimating motion and size of moving non-line-of-sight objects in cluttered environments. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Colorado Springs, CO, June 2011), IEEE, 265–272.
- Qu, J., Liu, L., Chen, D., Lin, Z., Xu, G., Guo, B., Niu, H. Temporally and spectrally resolved sampling imaging with a specially designed streak camera. *Opt. Lett.* 31 (2006), 368–370.
- Velten, A., Fritz, A., Bawendi, M.G., Raskar, R. Multibounce time-of-flight imaging for object reconstruction from indirect light. In *Conference for Lasers and Electro-Optics* (OSA, 2012).
- Velten, A., Willwacher, T., Gupta, O., Veeraraghavan, A., Bawendi, M.G., Raskar, R. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nat. Commun.* 3, 745 (2012), 745:1–745:8.
- Velten, A., Wu, D., Jarabo, A., Masia, B., Barsi, C., Joshi, C., Lawson, E., Bawendi, M.G., Gutierrez, D., Raskar, R. Femtophotography: Capturing and visualizing the propagation of light. *ACM Trans. Graph.* 32, 4 (2013), 44:1–44:8.
- Velten, A., Wu, D., Jarabo, A., Masia, B., Barsi, C., Lawson, E., Joshi, C., Gutierrez, D., Bawendi, M.G., Raskar, R. Relativistic ultrafast rendering using time-of-flight imaging. In *ACM SIGGRAPH Talks* (2012).
- Wu, D., Velten, A., O'Toole, M., Masia, B., Agrawal, A., Dai, Q., Raskar, R. Decomposing global light transport using time of flight imaging. *Int. J. Comput. Vision* 107, 2 (April 2014), 123–138.
- Wu, D., Wetzstein, G., Barsi, C., Willwacher, T., O'Toole, M., Naik, N., Dai, Q., Kutulakos, K., Raskar, R. Frequency Analysis of Transient Light Transport with Applications in Bare Sensor Imaging. Springer, Berlin, Heidelberg, 2012, 542–555.

Andreas Velten (avelten@morgridge.org), University of Wisconsin, Morgridge Institute for Research, Madison, WI.

Di Wu, Belen Masia, Christopher Barsi, Chinmaya Joshi, Everett Lawson, and Ramesh Raskar (wudi@rdz.cn, [cbarsi, raskar]@mit.edu), MIT Media Lab, Cambridge, MA.

Belen Masia, Adrian Jarabo, and Diego Gutierrez (bmasiac@mpi-inf.mpg.de, [ajarabo, diegog]@unizar.es), Universidad de Zaragoza, Zaragoza, Spain.

Belen Masia MPI Informatik, Saarbrücken, Germany.

Moungi Bawendi (mgb@mit.edu) MIT Department of Chemistry, Cambridge, MA.



Watch the authors discuss their work in this exclusive *Communications* video. <http://cacm.acm.org/videos/imaging-the-propagation-of-light-through-scenes-at-picosecond-resolution>

© 2016 ACM 0001-0782/16/09 \$15.00

World-Renowned Journals from ACM

ACM publishes over 50 magazines and journals that cover an array of established as well as emerging areas of the computing field. IT professionals worldwide depend on ACM's publications to keep them abreast of the latest technological developments and industry news in a timely, comprehensive manner of the highest quality and integrity. For a complete listing of ACM's leading magazines & journals, including our renowned Transaction Series, please visit the ACM publications homepage: www.acm.org/pubs.

ACM Transactions on Interactive Intelligent Systems



ACM Transactions on Interactive Intelligent Systems (TIIS). This quarterly journal publishes papers on research encompassing the design, realization, or evaluation of interactive systems incorporating some form of machine intelligence.

ACM Transactions on Computation Theory



ACM Transactions on Computation Theory (ToCT). This quarterly peer-reviewed journal has an emphasis on computational complexity, foundations of cryptography and other computation-based topics in theoretical computer science.

PLEASE CONTACT ACM MEMBER SERVICES TO PLACE AN ORDER
Phone: 1.800.342.6626 (U.S. and Canada)
+1.212.626.0500 (Global)
Fax: +1.212.944.1318
(Hours: 8:30am–4:30pm, Eastern Time)
Email: acmhelp@acm.org
Mail: ACM Member Services
General Post Office
PO Box 30777
New York, NY 10087-0777 USA



Association for Computing Machinery

Advancing Computing as a Science & Profession

www.acm.org/pubs

Technical Perspective

Jupiter Rising

By Andrew W. Moore

DESPITE ALL OF US regularly reading of the growth of the Internet, whether it be capacity, number of users, or breadth of applications, beyond the proliferation, and ever-shortening upgrade cycle of our mobile phones, we rarely observe a physical manifestation of such growth. Yet, it is within warehouse-sized computer facilities, the datacenters that attract the superlative *hyper*, that the Internet growth is physically manifest. Not only must these facilities face operational challenges arising from a large and dynamic number of users, but the continuous growth in demand upon these facilities must also be accommodated. Such growth for the hyper-scale datacenter translates to a modest-sized facility covering 15 acres, housing over 200,000 servers, consuming mega-gallons of water and perhaps hundreds of megawatts of electricity too.

The nature and role of the datacenter has also grown; once familiar only as an organization's facility or the ever-popular co-location and hosting facility where organizations might securely site their equipment. The warehouse or hyper-scale datacenter of the type discussed in the following paper, is a wholly recent beast, with ownership and control reminiscent of a single-task supercomputer. Yet, this is only a first-glance similarity; the warehouse computer approaches superlative-rich scales of computers, bandwidth needs, memory, and storage, and is used for far more than the single application of providing a popular set of Web pages.

Any datacenter may be thought of as a set of computers: some serving storage, most providing processing, combined with a network that permits both internal and external communication; the number and density of machines may vary and the network operation seems simple yet what gives rise to the complexity of the hyper datacenter is scale. The hyper-scale datacenter is an optimization; a solution to the optimal cost of capital and operation compels ever larger datacenters—as large as is practical to manage—and the hyper-

scale facilities are the current logical conclusion: provided the operations are also scalable then the design will seek the largest system possible.

When dealing with 100,000 servers, the hyper-scale datacenter must accommodate the certainty of equipment failure, the operational variability of supporting many different systems, and the practical issues of such huge collections of computers: such as the complex network to support communications among each computer. Designing for systems at scale has long been the *raison d'être* of solutions employed within the Internet itself; yet as this paper makes clear, many of the Internet mechanisms for maintaining large-scale networks are suboptimal when, as in the case of this work, the datacenter is largely homogeneous, exhibits strong regularity within its structure, and when its bandwidth needs exceed the capabilities of those of any available equipment vendor.

The tale of developing datacenter network-fabrics able to accommodate the rapid pace of change is hugely influenced by properties of the datacenter itself: regular modular organization of systems, well-known network organization, and well-understood network needs and the authors show how their solutions are enabled by these self-same properties. This does not prevent

The hyper-scale datacenter discussed in the following paper is a wholly recent beast, with ownership and control reminiscent of a single-task supercomputer.

the lessons and process described in this paper to be more widely applicable for those outside the arena of the hyper-scale datacenter architect.

Some solutions may be a confluence of circumstance. Opportunity is created by a measurement-derived understanding of traffic growth and a lack of any practical vendor. Keen to strip any solution of unnecessary features, the availability of cheap commodity (merchant) silicon and a software-centered network-control plane leads to a scalable, bespoke, network design. Migration from old systems to new would incur mammoth challenges for a single deployment or upgrade and yet the authors describe rolling a deployment five times between 2004 and 2012, all while also limiting the impact to services. Successive network implementations are tangibly improved by this experience.

The reader is presented with other operational insights including how to achieve practical scaling by focusing upon an architecture designed in sympathy with their datacenter peculiarities. An example of this is how the design of a robust distributed system, the switch-fabric management itself, can be balanced against the benefit of a centralization of control. The authors illustrate how a centralized approach need not discard distributed advantages, an idea that is core for many scale-out designs. In turn, this provides a facility or mechanism using the resources of multiple machines all while coping with the failures inherent in any large design.

As datacenter services are central in all our lives, I commend "*Jupiter Rising*" to you. While the paper is steeped in operational experience it is far beyond a simple anecdote-ridden history tale; the authors describe solutions with the unique confluence of need, ability, and circumstances transposed into a fascinating set of insights for the reader. 

Andrew W. Moore is the Reader in Systems at the Computer Laboratory of the University of Cambridge, U.K.

Copyright held by author.

Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network

By Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Hong Liu, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat

Abstract

We present our approach for overcoming the cost, operational complexity, and limited scale endemic to datacenter networks a decade ago. Three themes unify the five generations of datacenter networks detailed in this paper. First, multi-stage Clos topologies built from commodity switch silicon can support cost-effective deployment of building-scale networks. Second, much of the general, but complex, decentralized network routing and management protocols supporting arbitrary deployment scenarios were overkill for single-operator, pre-planned datacenter networks. We built a centralized control mechanism based on a global configuration pushed to all datacenter switches. Third, modular hardware design coupled with simple, robust software allowed our design to also support inter-cluster and wide-area networks. Our datacenter networks run at dozens of sites across the planet, scaling in capacity by 100x over 10 years to more than 1 Pbps of bisection bandwidth. A more detailed version of this paper is available at Ref.²¹

1. INTRODUCTION

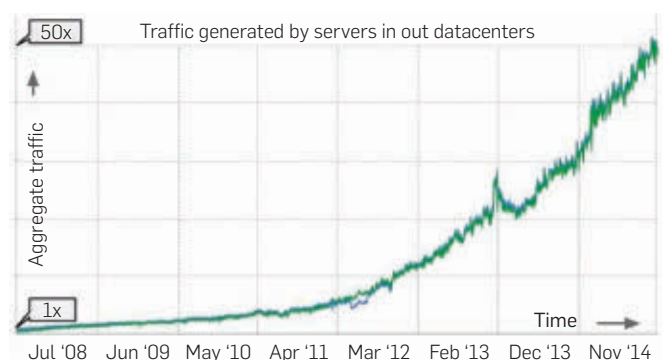
From the beginning, Google built and ran an internal cloud for all of our internal infrastructure services and external-facing applications. This shared infrastructure allowed us to run services more efficiently through statistical multiplexing of the underlying hardware, reduced operational overhead by allowing best practices and management automation to apply uniformly, and increased overall velocity in delivering new features, code libraries, and performance enhancements to the fleet.

We architect all these cloud services and applications as large-scale distributed systems.^{3, 6, 7, 10, 13} The inherent distributed nature of our services stems from the need to scale out and keep pace with an ever-growing global user population. Our services have substantial bandwidth and low-latency requirements, making the datacenter network a key piece of our infrastructure. Ten years ago, we found the cost and operational complexity associated with traditional datacenter network architectures to be prohibitive. Maximum network scale was limited by the cost and capacity of the highest-end switches available at any point in time.²³ Bandwidth demands in the datacenter

have been doubling every 12–15 months (Figure 1), even faster than the wide area Internet. Essentially, we could not buy a network at any price that could meet our scale and performance needs.

More recently, Google has been expanding aggressively in the public Cloud market, making our internal hardware and software infrastructure available to external customers. This shift has further validated our approach to building scalable, efficient, and highly available data center networks. With our internal cloud deployments, we could vertically integrate and centrally plan to co-optimize application and network behavior. A bottleneck in network performance could be alleviated by redesigning Google application behavior. If a single site did not provide requisite levels of availability, the application could be replicated to multiple sites. Finally, bandwidth requirements could be precisely measured and projected to inform future capacity requirements. With the external cloud, application bandwidth demands are highly bursty and much more difficult to adjust. For example, an external customer may in turn be running third party software that is difficult to modify. We have found that our data center network architecture substantially eases

Figure 1. Aggregate server traffic in our datacenter fleet.



The original version of this paper was published in the *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ACM, 183–197.

the management overhead of hosting external applications while simultaneously delivering new levels of performance and scale for Cloud customers.

Inspired by the community's ability to scale out computing with parallel arrays of commodity servers, we sought a similar approach for networking. This paper describes our experience with building five generations of custom data center network hardware and software by leveraging commodity hardware components, while addressing the control and management requirements introduced by our approach. We used the following principles in constructing our networks:

Clos topologies: To support graceful fault tolerance, increase the scale/bisection of our datacenter networks, and accommodate lower radix switches, we adopted Clos topologies^{4, 9, 15} for our datacenters. Clos topologies can scale to nearly arbitrary size by adding stages to the topology, principally limited by failure domain considerations and control plane scalability. They also have substantial in-built path diversity and redundancy, so the failure of any individual element can result in relatively small capacity reduction. However, they introduce substantial challenges as well, including managing the fiber fanout and more complex routing across multiple equal-cost paths.

Merchant silicon: Rather than use commercial switches targeting small-volume, large feature sets and high reliability, we targeted general-purpose merchant switch silicon, commodity priced, off the shelf, switching components. To keep pace with server bandwidth demands which scale with cores per server and Moore's law, we emphasized bandwidth density and frequent refresh cycles. Regularly upgrading network fabrics with the latest generation of commodity switch silicon allows us to deliver exponential growth in bandwidth capacity in a cost-effective manner.

Centralized control protocols: Control and management become substantially more complex with Clos topologies because we dramatically increase the number of discrete switching elements. Existing routing and management protocols were not well-suited to such an environment. To control this complexity, we observed that individual datacenter switches played a pre-determined forwarding role based on the cluster plan. We took this observation to one extreme by collecting and distributing dynamically changing link state information from a central, dynamically elected, node in the network. Individual switches could then calculate forwarding tables based on current link state relative to a statically configured topology.

Overall, our software architecture more closely resembles control in large-scale storage and compute platforms than traditional networking protocols. Network protocols typically use soft state based on pair-wise message exchange, emphasizing local autonomy. We were able to use the distinguishing characteristics and needs of our datacenter deployments to simplify control and management protocols, anticipating many of the tenets of modern Software Defined Networking (SDN) deployments.¹² The datacenter networks described in this paper represent some of the largest in the world, are in deployment at dozens of sites across the planet, and support thousands of internal and external

services, including external use through Google Cloud Platform. Our cluster network architecture found substantial reuse for inter-cluster networking in the same campus and even WAN deployments¹⁷ at Google.

2. BACKGROUND AND RELATED WORK

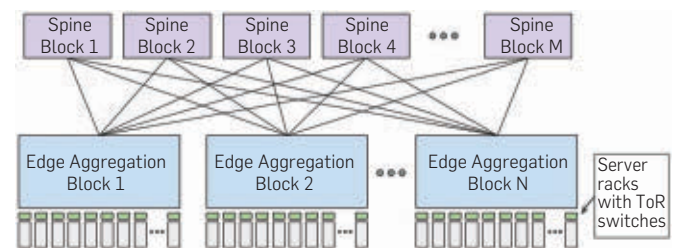
The tremendous growth rate of our infrastructure served as key motivation for our work in datacenter networking. Figure 1 shows aggregate server communication rates since 2008. Traffic has increased 50x in this time period, roughly doubling every year. A combination of remote storage access,^{5, 13} large-scale data processing,^{10, 16} and interactive web services³ drive our bandwidth demands. More recently, the growth rate has increased further with the popularity of the Google Cloud Platform¹⁴ running on our shared infrastructure.

In 2004, we deployed traditional cluster networks similar to Ref.⁴ This configuration supported 40 servers connected at 1Gb/s to a Top of Rack (ToR) switch with approximately 10:1 oversubscription in a cluster delivering 100Mb/s among 20k servers. High bandwidth applications had to fit under a single ToR to avoid the heavily oversubscribed ToR uplinks. Deploying large clusters was important to our services because there were many affiliated applications that benefited from high bandwidth communication. Consider large-scale data processing to produce and continuously refresh a search index, web search, and serving ads as affiliated applications. Larger clusters also substantially improve bin-packing efficiency for job scheduling by reducing stranding from cases where a job cannot be scheduled in any one cluster despite the aggregate availability of sufficient resources across multiple small clusters.²⁴

While our traditional cluster network architecture largely met our scale needs, it fell short in terms of overall performance and cost. With bandwidth per host limited to 100Mbps, packet drops associated with incast⁸ and outcast²⁰ were severe pain points. Increasing bandwidth per server would have substantially increased cost per server and reduced cluster scale.

We realized that existing commercial solutions could not meet our scale, management, and cost requirements. Hence, we decided to build our own custom data center network hardware and software. We started with the key insight that we could scale cluster fabrics to near arbitrary size by leveraging Clos topologies (Figure 2) and the then-emerging (ca. 2003) merchant switching silicon industry.¹¹ Table 1 summarizes a number of the top-level challenges we faced in constructing and managing building-scale network fabrics.

Figure 2. A generic three tier Clos architecture with edge switches (ToRs), aggregation blocks and spine blocks. All generations of Clos fabrics deployed in our datacenters follow variants of this architecture.



The following sections explain these challenges and the rationale for our approach in detail.

Our topological approach, reliance on merchant silicon, and load balancing across multipath are substantially similar to contemporaneous research.^{1, 15} Our centralized control protocols running on switch embedded processors are also related to subsequent substantial efforts in SDN.¹² Based on our experience in the datacenter, we later applied SDN to our Wide Area Network.¹⁷ For the WAN, more CPU intensive traffic engineering and BGP routing protocols led us to move control protocols onto external servers with more powerful CPUs.

3. NETWORK EVOLUTION

3.1. Firehose

Table 2 summarizes the multiple generations of our cluster network. With our initial approach, Firehose 1.0 (or FH1.0), our nominal goal was to deliver 1Gbps of nonblocking bisection bandwidth to each of 10K servers. Figure 3 details the FH1.0 topology using 8x10G switches in both the aggregation blocks as well as the spine blocks. The ToR switch delivered 2x10GE ports to the fabric and 24x1GE server ports.

Each aggregation block hosted 16 ToRs and exposed 32x10G ports towards 32 spine blocks. Each spine block had 32x10G towards 32 aggregation blocks resulting in a fabric

that scaled to 10K machines at 1G average bandwidth to any machine in the fabric.

Since we did not have any experience building switches but we did have experience building servers, we attempted to integrate the switching fabric into the servers via a PCI board. See top right inset in Figure 3. However, the uptime of servers was less than ideal. Servers crashed and were upgraded more frequently than desired with long reboot times. Network disruptions from server failure were especially problematic for servers housing ToRs connecting multiple other servers to the first stage of the topology.

The resulting wiring complexity for server to server connectivity, electrical reliability issues, availability and general issues associated with our first foray into switching doomed the effort to never seeing production traffic. At the same time, we consider FH1.0 to be a landmark effort internally. Without it and the associated learning, the efforts that followed would not have been possible.

Our first production deployment of a custom datacenter cluster fabric was Firehose 1.1 (FH1.1). We had learned from FH1.0 not to use servers to house switch chips. Thus, we built custom enclosures that standardized around the compact PCI chassis each with six independent linecards and a dedicated Single-Board Computer (SBC) to control the linecards using PCI. See insets in Figure 4. The fabric chassis did

Table 1. High-level summary of challenges we faced and our approach to address them.

Challenge	Our approach (section discussed in)
Introducing the network to production	Initially deploy as bag-on-the-side with a fail-safe big-red button (3.1)
High availability from cheaper components	Redundancy in fabric, diversity in deployment, robust software, necessary protocols only, reliable out of band control plane (3.1, 3.2, 5.1)
Individual racks can leverage full uplink capacity to external clusters	Introduce Cluster Border Routers to aggregate external bandwidth shared by all server racks (4.1)
Routing scalability	Scalable in-house IGP, centralized topology view and route control (5.2)
Interoperate with external vendor gear	Use standard BGP between Cluster Border Routers and vendor gear (5.2.5)
Small on-chip buffers	Congestion window bounding on servers, ECN, dynamic buffer sharing of chip buffers, QoS (6.1)
Routing with massive multipath	Granular control over ECMP tables with proprietary IGP (5.1)
Operating at scale	Leverage existing server installation, monitoring software; tools build and operate fabric as a whole; move beyond individual chassis-centric network view; single cluster-wide configuration (5.3)
Inter cluster networking	Portable software, modular hardware in other applications in the network hierarchy (4.2)

Table 2. Multiple generations of datacenter networks.

Datacenter generation	First deployed	Merchant silicon	ToR config	Aggregation block config	Spine block config	Fabric speed	Host speed	Bisection BW
Legacy network	2004	Vendor	48x1G	–	–	10G	1G	2T
Firehose 1.0	2005	8x10G 4x10G (ToR)	2x10G up 24x1G down	2x32x10G (B)	32x10G (NB)	10G	1G	10T
Firehose 1.1	2006	8x10G	4x10G up 48x1G down	64x10G (B)	32x10G (NB)	10G	1G	10T
Watchtower	2008	16x10G	4x10G up 48x1G down	4x128x10G (NB)	128x10G (NB)	10G	nx1G	82T
Saturn	2009	24x10G	24x10G	4x288x10G (NB)	288x10G (NB)	10G	nx10G	207T
Jupiter	2012	16x40G	16x40G	8x128x40G (B)	128x40G (NB)	10/40G	nx10G/nx40G	1.3P

B, Indicates blocking; NB, Indicates nonblocking.

not contain any backplane to interconnect the switch chips. All ports connected to external copper cables.

A major concern with FH1.1 in production was deploying an unproven new network technology for our mission critical applications. To mitigate risk, we deployed Firehose 1.1 in conjunction with our legacy networks as shown in Figure 5. We maintained a simple configuration; the ToR would forward default traffic to the legacy network (e.g., for connectivity to external clusters/data centers) while more specific intra-cluster traffic would use the uplinks to Firehose 1.1. We built a *Big Red Button* fail-safe to configure the ToRs to avoid Firehose uplinks in case of catastrophic failure.

3.2. Watchtower and Saturn: Global deployment

Our deployment experience with Firehose 1.1 was largely positive. We showed that services could enjoy substantially more bandwidth than with traditional architectures, all with lower cost per unit bandwidth. The main drawback to Firehose 1.1 was the deployment challenges with the external copper cabling.

Figure 3. Firehose 1.0 topology. Top right shows a sample 8x10G port fabric board in Firehose 1.0, which formed Stages 2, 3 or 4 of the topology.

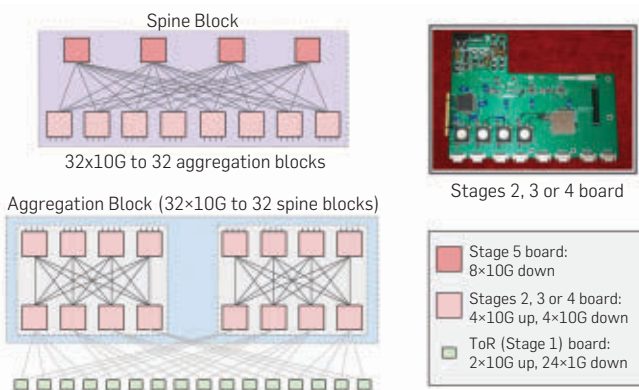
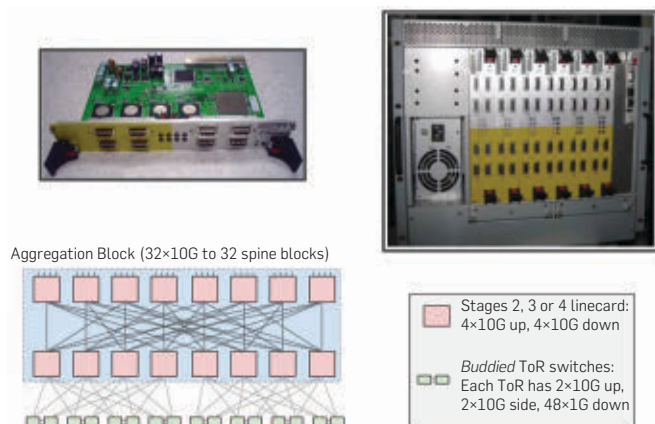


Figure 4. Firehose 1.1 packaging and topology. The top left picture shows a linecard version of the board from Figure 3. The top right picture shows a Firehose 1.1 chassis housing six such linecards. The bottom figure shows the aggregation block in Firehose 1.1, which was different from Firehose 1.0.



We used these experiences to design Watchtower, our third-generation cluster fabric. The key idea was to leverage the next-generation merchant silicon switch chips, 16x10G, to build a traditional switch chassis with a backplane. Figure 6 shows the half rack Watchtower chassis along with its internal topology and cabling. Watchtower consists of eight line cards, each with three switch chips. Two chips on each linecard have half their ports externally facing, for a total of 16x10GE SFP+ ports. All three chips also connect to a backplane for port to port connectivity. Watchtower deployment, with fiber cabling as seen in Figure 6 was substantially easier than the earlier Firehose deployments. The higher bandwidth density of the switching silicon also allowed us to build larger fabrics with more bandwidth to individual servers, a necessity as servers were employing an ever-increasing number of cores.

Saturn was the next iteration of our cluster architecture. The principal goals were to respond to continued increases in server bandwidth demands and to further increase maximum cluster scale. Saturn was built from 24x10G

Figure 5. Firehose 1.1 deployed as a bag-on-the-side Clos fabric.

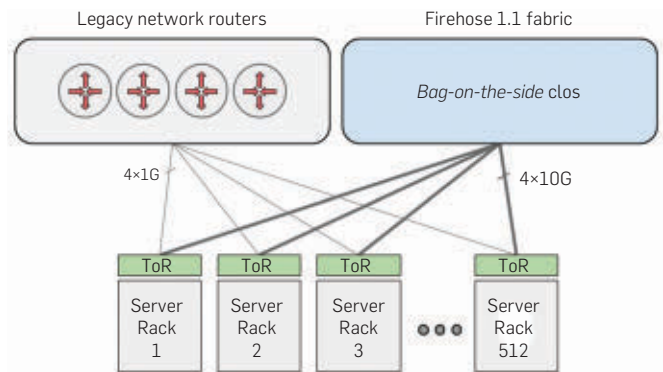
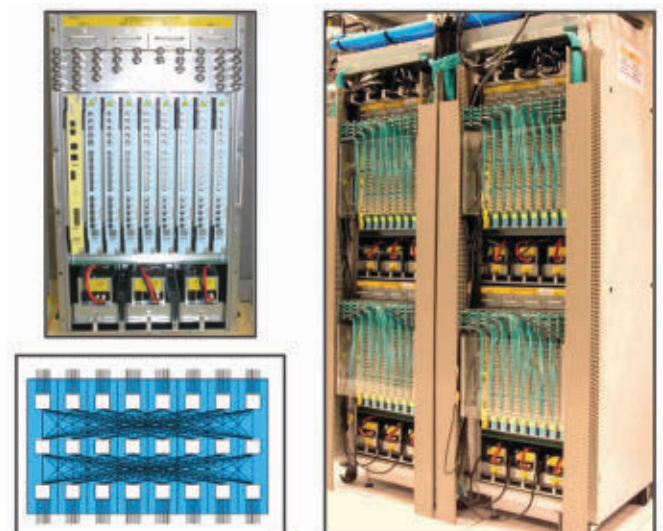


Figure 6. A 128x10G port Watchtower chassis (top left). The internal non-blocking topology over eight linecards (bottom left). Four chassis housed in two racks cabled with fiber (right).



merchant silicon building blocks. A Saturn chassis supports 12-linecards to provide a 288 port non-blocking switch. These chassis are coupled with new Pluto single-chip ToR switches; see Figure 7. In the default configuration, Pluto supports 20 servers with 4x10G provisioned to the cluster fabric for an average bandwidth of 2 Gbps for each server. For more bandwidth-hungry servers, we could configure the Pluto ToR with 8x10G uplinks and 16x10G to servers providing 5 Gbps to each server. Importantly, servers could burst at 10 Gbps across the fabric for the first time.

3.3. Jupiter: A 40G datacenter-scale fabric

As bandwidth requirements per server continued to grow, so did the need for uniform bandwidth across all clusters in the datacenter. With the advent of dense 40G capable merchant silicon, we could consider expanding our Clos fabric across the entire datacenter subsuming the inter-cluster networking layer. This would potentially enable an unprecedented pool of compute and storage for application scheduling. Critically, the unit of maintenance could be kept small enough relative to the size of the fabric that most applications could now be agnostic to network maintenance windows unlike previous generations of the network.

Jupiter, our next generation datacenter fabric, needed to scale more than 6x the size of our largest existing fabric. Unlike previous iterations, we set a requirement for incremental deployment of new network technology because the cost in resource stranding and downtime was too high. Upgrading networks by simply forklifting existing clusters stranded hosts already in production. With Jupiter, new technology would need to be introduced into the network *in situ*. Hence, the fabric must support heterogeneous hardware and speeds.

At Jupiter scale, we had to design the fabric through individual building blocks, see Figure 8. Our unit of deployment is a Centauri chassis, a 4RU chassis housing two linecards, each with two switch chips with 16x40G ports controlled by a separate CPU linecard. Each port could be configured in

4x10G or 40G mode. There were no backplane data connections between these chips; all ports were accessible on the front panel of the chassis.

We employed the Centauri switch as a ToR switch with each of the four chips serving a subnet of machines. In one ToR configuration, we configured each chip with 48x10G to servers and 16x10G to the fabric. Servers could be configured with 40G burst bandwidth for the first time in production (see Table 2). Four Centauris made up a Middle Block (MB) for use in the aggregation block. The logical topology of an MB was a 2-stage blocking network, with 256x10G links available for ToR connectivity and 64x40G available for connectivity to the rest of the fabric through the spine (Figure 9).

Each ToR chip connects to eight such MBs with dual redundant 10G links. The dual redundancy aids fast

Figure 8. Building blocks used in the Jupiter topology.

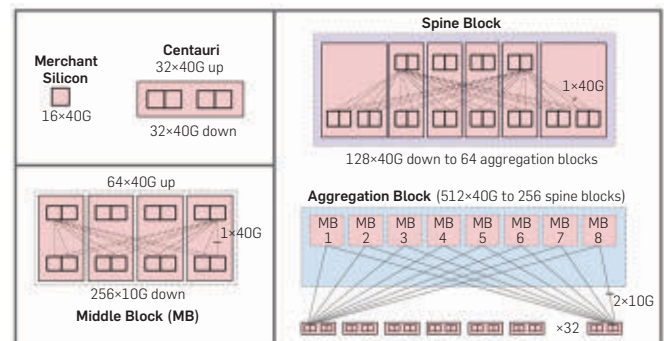
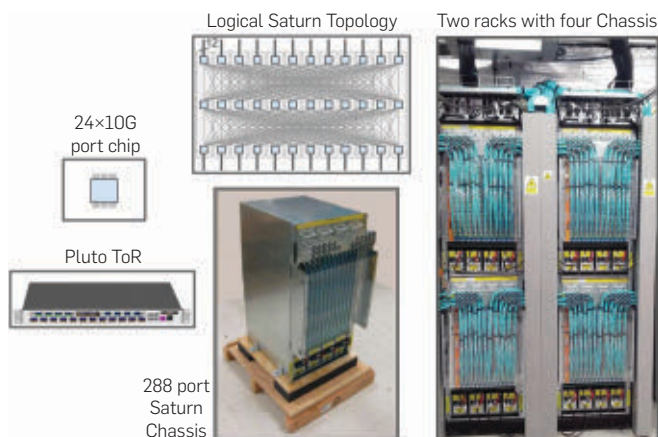


Figure 9. Jupiter Middle blocks housed in racks.



Figure 7. Components of a Saturn fabric. A 24x10G Pluto ToR Switch and a 12-linecard 288x10G Saturn chassis (including logical topology) built from the same switch chip. Four Saturn chassis housed in two racks cabled with fiber (right).



reconvergence for the common case of single link failure or maintenance. Each aggregation block exposes up to 512x40G links towards the spine blocks. Jupiter employs six Centauris in a spine block exposing 128x40G ports towards the aggregation blocks. We limited the size of Jupiter to 64 aggregation blocks for dual redundant links between each spine block and aggregation block pair at the largest scale, once again for local reconvergence on single link failure. In its largest configuration, Jupiter supports 1.3 Pbps bisection bandwidth among servers.

4. EXTERNAL CONNECTIVITY

4.1. WCC: Decommissioning legacy routers

Through the first few Watchtower deployments, all cluster fabrics were deployed as *bag-on-the-side* networks coexisting with legacy networks (Figure 5). Time and experience ameliorated safety concerns, tipping the balance in favor of reducing the operational complexity, cost, and performance limitations of deploying two parallel networks.

Hence, our next goal was to decommission our legacy routers by connecting the fabric directly to the inter-cluster networking layer with Cluster Border Routers (CBRs). This effort was internally called WCC.

To deliver high external bandwidth, we chose to build separate aggregation blocks for external connectivity, physically and topologically identical to those used for ToR connectivity. However, we reallocated the ports normally employed for ToR connectivity to connect to external fabrics. This mode of connectivity enabled an isolated layer of switches to peer with external routers, limiting the blast radius from an external facing configuration change. Moreover, it limited the places where we would have to integrate our in-house IGP (Section 5.2) with external routing protocols.

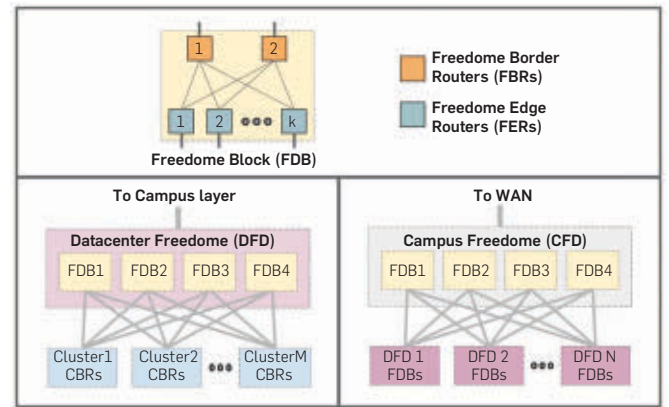
4.2. Inter-cluster networking

We deploy multiple clusters within the same building and multiple buildings on the same campus. Given the relationship between physical distance and network cost, our job scheduling and resource allocation infrastructure leverages campus-level and building-level locality to co-locate loosely affiliated services as close to one another as possible. Each CBR block developed for WCC supported 2.56Tbps of external connectivity in Watchtower and 5.76Tbps in Saturn. However, our external networking layers were still based on expensive and port-constrained vendor gear. Freedom, the third step in the evolution of our network fabrics, involved replacing vendor-based inter cluster switching.

We employed the BGP capability we developed for our CBRs to build two-stage fabrics that could speak BGP at both the inter cluster and intra campus connectivity layers. See Figure 10. The Freedom Block shown in the top figure is the basic building block for Freedom and is a collection of routers configured to speak BGP.

A Datacenter Freedom typically comprises four independent blocks to connect multiple clusters in the same datacenter building. Inter-cluster traffic local to the same building would travel from the source cluster's CBR layer to

Figure 10. Two-stage fabrics used for inter-cluster and intra-campus connectivity.



the Datacenter Freedom and finally to the CBR layer of the destination cluster. We connect the top stage router ports of the Freedom Block to the campus connectivity layer to the north. The bottom left figure in Figure 10 depicts a Datacenter Freedom.

Recursively, a Campus Freedom also typically comprises four independent Freedom Blocks to connect multiple Data-center Freedoms in a campus on the south and the WAN connectivity layer on the north-facing side. The bottom right figure in Figure 10 depicts a Campus Freedom. This same approach would later find application for our WAN deployments.¹⁷

5. SOFTWARE CONTROL

5.1. Discussion

As we set out to build the control plane for our network hardware, we faced the following high level trade-off: deploy traditional decentralized routing protocols such as OSPF/IS-IS/BGP to manage our fabrics or build a custom control plane to leverage some of the unique characteristics and homogeneity of our cluster network.

We chose to build our own control plane for a number of reasons. First, and most important, existing routing protocols did not, at the time, have good support for multi-path, equal-cost forwarding. Second, there were no high quality open source routing stacks a decade ago. Third, we were concerned about the protocol overhead of running broadcast-based routing protocols across fabrics with potentially thousands of switching elements. Scaling techniques like OSPF Areas¹⁹ appeared hard to configure and to reason about.²² Fourth, network manageability was a key concern and maintaining hundreds of independent switch stacks and, for example, BGP configurations seemed daunting.

Our approach was driven by the need to route across a largely static topology with massive multipath. Each switch had a predefined role according to its location in the fabric and could be configured as such. A centralized solution where a route controller collected dynamic link state information and redistributed this link state to all switches over a reliable

out-of-band Control Plane Network (CPN) appeared to be substantially simpler and more efficient from a computation and communication perspective. The switches could then calculate forwarding tables based on current link state as deltas relative to the underlying, known static topology that was pushed to all switches.

Overall, we treated the datacenter network as a single fabric with tens of thousands of ports rather than a collection of hundreds of autonomous switches that had to dynamically discover information about the fabric. We were, at this time, inspired by the success of large-scale distributed storage systems with a centralized manager.¹³ Our design informed the control architecture for both Jupiter datacenter networks and Google's B4 WAN,¹⁷ both of which are based on OpenFlow¹⁸ and custom SDN control stacks.

5.2. Routing

We now present the key components of Firepath, our routing architecture for Firehose, Watchtower, and Saturn fabrics. A number of these components anticipate some of the principles of modern SDN, especially in using logically centralized state and control. First, all switches are configured with the baseline or intended topology. The switches learn actual configuration and link state through pair-wise neighbor discovery. Next, routing proceeds with each switch exchanging its local view of connectivity with a centralized Firepath master, which redistributes global link state to all switches. Switches locally calculate forwarding tables based on this current view of network topology. To maintain robustness, we implement a Firepath master election protocol. Finally, we leverage standard BGP only for route exchange at the edge of our fabric, redistributing BGP-learned routes through Firepath.

Neighbor discovery to verify connectivity. Building a fabric with thousands of cables invariably leads to multiple cabling errors. Moreover, correctly cabled links may be re-connected incorrectly after maintenance. Allowing traffic to use a miscabled link can lead to forwarding loops. Links that fail unidirectionally or develop high packet error rates should also be avoided and scheduled for replacement. To address these issues, we developed *Neighbor Discovery* (ND), an online liveness and peer correctness checking protocol. ND uses the configured view of cluster topology together with a switch's local ID to determine the expected peer IDs of its local ports and verifies that via message exchange.

Firepath. We support Layer 3 routing all the way to the ToRs via a custom Interior Gateway Protocol (IGP), Firepath. Firepath implements centralized topology state distribution, but distributed forwarding table computation with two main components. A Firepath client runs on each fabric switch, and a set of redundant Firepath masters run on a selected subset of spine switches. Clients communicate with the elected master over the CPN. Figure 11 shows the interaction between the Firepath client and the rest of the switch stack. Figure 12 illustrates the protocol message exchange between various routing components.

At startup, each client is loaded with the static topology of the entire fabric called the cluster config. Each client

Figure 11. Firepath component interactions. (A) Non-CBR fabric switch and (B) CBR switch.

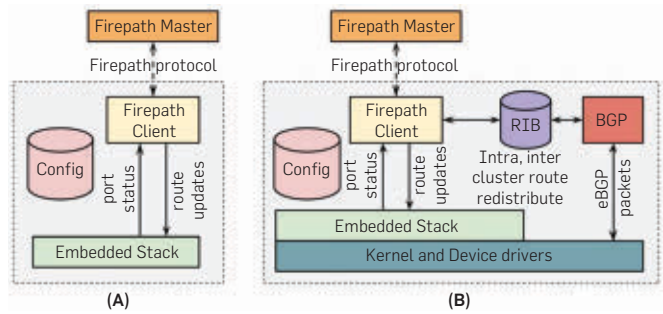
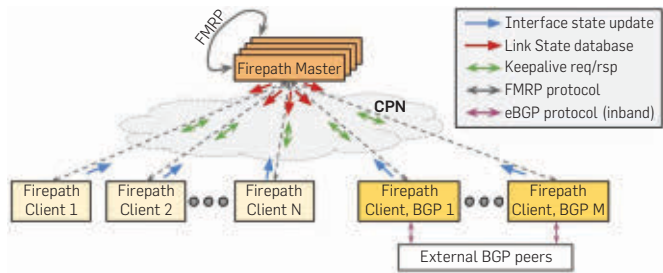


Figure 12. Protocol messages between Firepath client and Firepath master, between Firepath masters and between CBR and external BGP speakers.



collects the state of its local interfaces from the embedded stack's interface manager and transmits this state to the master. The master constructs a Link State Database (LSD) with a monotonically increasing version number and distributes it to all clients via UDP/IP multicast over the CPN. After the initial full update, a subsequent LSD contains only the diffs from the previous state. The entire network's LSD fits within a 64 KB payload. On receiving an LSD update, each client computes shortest path forwarding with Equal-Cost Multi-Path (ECMP) and programs the hardware forwarding tables local to its switch.

Path diversity and convergence on failures. For rapid convergence on interface state change, each client computes the new routing solution and updates the forwarding tables independently upon receiving an LSD update. Since clients do not coordinate during convergence, the network can experience small transient loss while the network transitions from the old to the new state. However, assuming churn is transient, all switches eventually act on a globally consistent view of network state.

Firepath LSD updates contain routing changes due to planned and unplanned network events. The frequency of such events observed in a typical cluster is approximately 2000 times/month, 70 times/day, or 3 times/hour.

Firepath master redundancy The centralized Firepath master is a critical component in the Firepath system. It collects and distributes interface states and synchronizes the Firepath clients via a keepalive protocol. For availability, we run redundant master instances on pre-selected spine switches. Switches know the candidate masters via their

static configuration. The Firepath Master Redundancy Protocol (FMRP) handles master election and bookkeeping between the active and backup masters over the CPN.

FMRP has been robust in production over multiple years and many clusters. Since master election is *sticky*, a misbehaving master candidate does not cause changes in mastership and churn in the network. In the rare case of a CPN partition, a multi-master situation may result, which immediately alerts network operators for manual intervention.

Cluster border router. Our cluster fabrics peer with external networks via BGP. To this end, we integrated a BGP stack on the CBR with Firepath. This integration has two key aspects: (i) enabling the BGP stack on the CBRs to communicate inband with external BGP speakers, and (ii) supporting route exchange between the BGP stack and Firepath. Figure 11B shows the interaction between the BGP stack, Firepath, and the switch kernel and embedded stack.

A proxy process on the CBR exchanges routes between BGP and Firepath. This process exports intra-cluster routes from Firepath into the BGP RIB and picks up inter-cluster routes from the BGP RIB, redistributing them into Firepath. We made a simplifying assumption by summarizing routes to the cluster-prefix for external BGP advertisement and the /0 default route to Firepath. In this way, Firepath manages only a single route for all outbound traffic, assuming all CBRs are viable for traffic leaving the cluster. Conversely, we assume all CBRs are viable to reach any part of the cluster from an external network. The rich path diversity inherent to Clos fabrics enables both these assumptions.

5.3. Configuration and management

Next, we describe our approach to cluster network configuration and management prior to Jupiter. Our primary goal was to manufacture compute clusters and network fabrics as fast as possible throughout the entire fleet. Thus, we favored simplicity and reproducibility over flexibility. We supported only a limited number of fabric parameters, used to generate all the information needed by various groups to deploy the network, and built simple tools and processes to operate the network. As a result, the system was easily adopted by a wide set of technical and non-technical support personnel responsible for building data centers.

Configuration generation approach. Our key strategy was to view the entire cluster network top-down as a single static fabric composed of switches with pre-assigned roles, rather than bottom-up as a collection of switches individually configured and assembled into a fabric. We also limited the number of choices at the cluster-level, essentially providing a simple menu of fabric sizes and options, based on the projected maximum size of a cluster as well as the chassis type available.

The configuration system is a pipeline that accepts a specification of basic cluster-level parameters such as the size of the spine, base IP prefix of the cluster and the list of ToRs and their rack indexes and then generates a set of output files for various operations groups.

We distribute a single monolithic cluster configuration to all switches (chassis and ToRs) in the cluster. Each switch

simply extracts its relevant portion. Doing so simplifies configuration generation but every switch has to be updated with the new config each time the cluster configuration changes. Since cluster configurations do not change frequently, this additional overhead is not significant.

Switch management approach. We designed a simple management system on the switches. We did not require most of the standard network management protocols. Instead, we focused on protocols to integrate with our existing server management infrastructure. We benefited from not drawing arbitrary lines between server and network infrastructure; in fact, we set out to make switches essentially look like regular machines to the rest of fleet. Examples include large scale monitoring, image management and installation, and syslog collection and alerting.

Fabric operation and management. For fabric operation and management, we continued with the theme of leveraging the existing scalable infrastructure built to manage and operate the server fleet. We built additional tools that were aware of the network fabric as a whole, thus hiding complexity in our management software. As a result, we could focus on developing only a few tools that were truly specific to our large scale network deployments, including link/switch qualification, fabric expansion/upgrade, and network troubleshooting at scale. Also important was collaborating closely with the network operations team to provide training before introducing each major network fabric generation, expediting the ramp of each technology across the fleet.

Troubleshooting misbehaving traffic flows in a network with such high path diversity is daunting for operators. Therefore, we extended debugging utilities such as traceroute and ICMP to be aware of the fabric topology. This helped with locating switches in the network that were potentially blackholing flows. We proactively detect such anomalies by running probes across servers randomly distributed in the cluster. On probe failures, these servers automatically run traceroutes and identify suspect failures in the network.

6. EXPERIENCE

6.1. Fabric congestion

Despite the capacity in our fabrics, our networks experienced high congestion drops as utilization approached 25%. We found several factors contributed to congestion: (i) inherent burstiness of flows led to inadmissible traffic in short time intervals typically seen as incast⁸ or outcast²⁰; (ii) our commodity switches possessed limited buffering, which was sub optimal for our server TCP stack; (iii) certain parts of the network were intentionally kept oversubscribed to save cost, for example, the uplinks of a ToR; and (iv) imperfect flow hashing especially during failures and in presence of variation in flow volume.

We used several techniques to alleviate the congestion in our fabrics. First, we configured our switch hardware schedulers to drop packets based on QoS. Thus, on congestion we would discard lower priority traffic. Second, we tuned the hosts to bound their TCP congestion window for intracluster traffic to avoid overrunning the small buffers in our switch chips. Third, for our early fabrics, we employed link-level pause at

ToRs to keep servers from over-running oversubscribed uplinks. Fourth, we enabled Explicit Congestion Notification (ECN) on our switches and optimized the host stack response to ECN signals.² Fifth, we monitored application bandwidth requirements in the face of oversubscription ratios and could provision bandwidth by deploying Pluto ToRs with four or eight uplinks as required. Sixth, the merchant silicon had shared memory buffers used by all ports, and we tuned the buffer sharing scheme on these chips so as to dynamically allocate a disproportionate fraction of total chip buffer space to absorb temporary traffic bursts. Finally, we carefully configured switch hashing functionality to support good ECMP load balancing across multiple fabric paths.

Our congestion mitigation techniques delivered substantial improvements. We reduced the packet discard rate in a typical Clos fabric at 25% average utilization from 1% to < 0.01%. Further improving fabric congestion response remains an ongoing effort.

6.2. Outages

While the overall availability of our datacenter fabrics has been satisfactory, our outages fall into three categories representing the most common failures in production: (i) control software problems at scale; (ii) aging hardware exposing previously unhandled failure modes; and (iii) misconfigurations of certain components.

Control software problems at large scale. A datacenter power event once caused the entire fabric to restart simultaneously. However, the control software did not converge without manual intervention. The instability took place because our liveness protocol (ND) and route computation contended for limited CPU resources on embedded switch CPUs. On entire fabric reboot, routing experienced huge churn, which, in turn, led ND not to respond to heartbeat messages quickly enough. This in turn led to a snowball effect for routing where link state would spuriously go from up to down and back to up again. We stabilized the network by manually bringing up a few blocks at a time.

Going forward, we included the worst case fabric reboot in our test plans. Since the largest scale datacenter could never be built in a hardware test lab, we launched efforts to stress test our control software at scale in virtualized environments. We also heavily scrutinized any timer values in liveness protocols, tuning them for the worst case while balancing slower reaction time in the common case. Finally, we reduced the priority of non-critical processes that shared the same CPU.

Aging hardware exposes unhandled failure modes. Over years of deployment, our inbuilt fabric redundancy degraded as a result of aging hardware. For example, our software was vulnerable to internal/backplane link failures, leading to rare traffic blackholing. Another example centered around failures of the CPN. Each fabric chassis had dual redundant links to the CPN in active-standby mode. We initially did not actively monitor the health of both the active and standby links. With age, the vendor gear suffered from unidirectional failures of some CPN links exposing unhandled corner cases in our routing protocols. Both these problems would have been easier to mitigate had

the proper monitoring and alerting been in place for fabric backplane and CPN links.

Component misconfiguration. A prominent misconfiguration outage was on a Freedom fabric. Recall that a Freedom chassis runs the same codebase as the CBR with its integrated BGP stack. A CLI interface to the CBR BGP stack supported configuration. We did not implement locking to prevent simultaneous read/write access to the BGP configuration. During a planned BGP reconfiguration of a Freedom block, a separate monitoring system coincidentally used the same interface to read the running config while a change was underway. Unfortunately, the resulting partial configuration led to undesirable behavior between Freedom and its BGP peers.

We mitigated this error by quickly reverting to the previous configuration. However, it taught us to harden our operational tools further. It was not enough for tools to configure the fabric as a whole; they needed to do so in a safe, secure and consistent way.

7. CONCLUSION

This paper presents a retrospective on ten years and five generations of production datacenter networks. We employed complementary techniques to deliver more bandwidth to larger clusters than would otherwise be possible at any cost. We built multi-stage Clos topologies from bandwidth-dense but feature-limited merchant switch silicon. Existing routing protocols were not easily adapted to Clos topologies. We departed from conventional wisdom to build a centralized route controller that leveraged global configuration of a predefined cluster plan pushed to every datacenter switch. This centralized control extended to our management infrastructure, enabling us to eschew complex protocols in favor of best practices from managing the server fleet. Our approach has enabled us to deliver substantial bisection bandwidth for building-scale fabrics, all with significant application benefit.

Acknowledgments

Many teams contributed to the success of the datacenter network within Google. In particular, we would like to acknowledge the Platforms Networking (PlaNet) Hardware and Software Development, Platforms Software Quality Assurance (SQA), Mechanical Engineering, Cluster Engineering (CE), Network Architecture and Operations (NetOps), Global Infrastructure Group (GIG), and Site Reliability Engineering (SRE) teams, to name a few. 

References

1. Al-Fares, M., Loukissas, A., Vahdat, A. A scalable, commodity data center network architecture. In *ACM SIGCOMM Computer Communication Review*. Volume 38 (2008), ACM, 63–74.
2. Alizadeh, M., Greenberg, A., Maltz, D.A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., Sridharan, M. Data center TCP (DCTCP). *ACM SIGCOMM Comput. Commun. Rev.* 41, 4 (2011), 63–74.
3. Barroso, L.A., Dean, J., Holzle, U. Web search for a planet: The Google cluster architecture. *Micro. IEEE* 23, 2 (2003), 22–28.
4. Barroso, L.A., Hölzle, U. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Syn. Lect. Comput. Architect.* 4, 1 (2009), 1–108.
5. Calder, B., Wang, J., Ogus, A., Nilakantan, N., Skjolsvold, A., McKelvie, S., Xu, Y., Srivastav, S., Wu, J., Simitci, H., et al. Windows Azure storage: A highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (2011), ACM, 143–157.

6. Chambers, C., Raniwala, A., Perry, F., Adams, S., Henry, R.R., Bradshaw, R., Weizenbaum, N. Flumejava: Easy, efficient data-parallel pipelines. In *ACM Sigplan Notices*. Volume 45 (2010), ACM, 363–375.
7. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.* 26, 2 (2008), 4.
8. Chen, Y., Griffith, R., Liu, J., Katz, R.H., Joseph, A.D. Understanding TCP incast throughput collapse in datacenter networks. In *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking* (2009), ACM, 73–82.
9. Clos, C. A study of non-blocking switching networks. *Bell Syst. Tech. J.* 32, 2 (1953), 406–424.
10. Dean, J., Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.
11. Farrington, N., Rubow, E., Vahdat, A. Data center switch architecture in the age of merchant silicon. In *Proceedings of the 17th IEEE Symposium on HOT Interconnects, 2009* (2009), 93–102.
12. Feamster, N., Rexford, J., Zegura, E. The road to SDN: An intellectual history of programmable networks. *ACM Queue* 11, 12 (December 2013), 87–98.
13. Ghemawat, S., Gobioff, H., Leung, S.-T. The Google file system. In *ACM SIGOPS Operating Systems Review*. Volume 37 (2003), ACM, 29–43.
14. Google Cloud Platform. <https://cloud.google.com>.
15. Greenberg, A., Hamilton, J.R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D.A., Patel, P., Sengupta, S. VL2: A scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM Computer Communication Review* (2009), 51–62.
16. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D. Dryad: Distributed data-parallel programs from sequential building blocks. In *Proceedings of the ACM SIGOPS Operating Systems Review* (2007), 59–72.
17. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., Vahdat, A. B4: Experience with a globally-deployed software defined WAN. In *Proceedings of the ACM SIGCOMM* (2013), 3–14.
18. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* 38, 2 (2008), 69–74.
19. Moy, J. OSPF version 2. STD 54, RFC Editor, April 1998. <http://www.rfc-editor.org/rfc/rfc2328.txt>.
20. Prakash, P., Dixit, A.A., Hu, Y.C., Kompella, R.R. The TCP outcast problem: Exposing unfairness in data center networks. In *Proceedings of the NSDI* (2012), 413–426.
21. Singh, A., Ong, J., Agarwal, A., Anderson, G., Armistead, A., Bannon, R., Boving, S., Desai, G., Felderman, B., Germano, P., Kanagala, A., Provost, J., Simmons, J., Tanda, E., Wanderer, J., Hölzle, U., Stuart, S., Vahdat, A. Jupiter rising: A decade of clos topologies and centralized control in Google's datacenter network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (2015), ACM, 183–197.
22. Thorup, M. OSPF areas considered harmful. IETF Internet Draft 00, individual, April 2003. <http://tools.ietf.org/html/draft-thorup-ospf-harmful-00>.
23. Vahdat, A., Al-Fares, M., Farrington, N., Mysore, R.N., Porter, G., Radhakrishnan, S. Scale-out networking in the data center. *IEEE MICRO* 30, 4 (August 2010), 29–41.
24. Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E., Wilkes, J. Large-scale cluster management at Google with Borg. In *Proceedings of the Tenth European Conference on Computer Systems* (2015), ACM, 18.

Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand

Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat (jupiter-sigcomm@google.com) Google, Inc.

Copyright held by authors/owners.

ACM Transactions on Parallel Computing

Solutions to Complex Issues in Parallelism

Editor-in-Chief: Phillip B. Gibbons, Intel Labs, Pittsburgh, USA



ACM Transactions on Parallel Computing (TOPC) is a forum for novel and innovative work on all aspects of parallel computing, including foundational and theoretical aspects, systems, languages, architectures, tools, and applications. It will address all classes of parallel-processing platforms including concurrent, multithreaded, multicore, accelerated, multiprocessor, clusters, and supercomputers.

Subject Areas

- Parallel Programming Languages and Models
- Parallel System Software
- Parallel Architectures
- Parallel Algorithms and Theory
- Parallel Applications
- Tools for Parallel Computing



Association for
Computing Machinery

Advancing Computing as a Science & Profession

For further information or to submit your manuscript, visit topc.acm.org

Subscribe at www.acm.org/subscribe

CAREERS

Bradley University Assistant Professor

The Computer Science and Information Systems Department at **Bradley University** invites applications for a tenure-track **Assistant Professor** position starting in August 2017. The tenure-track Assistant Professor requires a PhD in Computer Science or a closely related field; candidates working on their dissertation with anticipated completion date before August 2017 will be considered. **Please** visit www.bradley.edu/humanresources/ for full position description and application process.

Lamar University Department of Computer Science Tenure-track Assistant Professor

Lamar University's Department of Computer Science seeks applications for a tenure-track Assistant Professor position beginning Spring 2017. Applicants must have a PhD in Computer Science and a strong commitment to teaching and research in cyber and/or software security, or high performance computing. Lamar is an AA/EEO state supported university of approximately 15,000 students. It offers the B.S. and M.S. in Computer Science. There are 9 full-time faculty and 500 undergraduate and graduate majors. Review of applications will begin on October 1, 2016, and continue until the position is filled. Apply at <http://jobs.lamar.edu/postings/3118>.

If you have additional questions, please address them at Stefan.Andrei@lamar.edu

San Diego State University Department of Computer Science Chair of Computer Science

Department of Computer Science at SDSU seeks candidates for the Chair position with a PhD in Computer Science or a related discipline, and a sustained record of supported research. The Department is a dynamic and growing unit looking for a visionary Chair to lead its future development.

We strive to build and sustain a welcoming environment for all. SDSU is seeking applicants with commitment to working effectively with individuals from diverse backgrounds and members of underrepresented groups.

For more details and application procedures, please apply via <https://apply.interfolio.com/36248>.

SDSU is a Title IX, equal opportunity employer. A full version of this ad can be found at: <http://cs.sdsu.edu/>

Stanford University Graduate School of Business Faculty Positions in Operations, Information and Technology

The Operations, Information and Technology (OIT) area at the Graduate School of Business, Stanford University, is seeking qualified applicants for full-time, tenure-track positions, starting September 1, 2017. All ranks and relevant disciplines will be considered. Applicants are considered in all areas of Operations, Information and Technology (OIT) that are broadly defined to include the analytical and empirical study of technological systems, in which technology, people, and markets interact. It thus includes operations, information systems/technology, and management of technology. Applicants are expected to have rigorous training in management science, engineering, computer science, economics, and/or statistical modeling methodologies. The appointed will be expected to do innovative research in the OIT field, to participate

in the school's PhD program, and to teach both required and elective courses in the MBA program. Junior applicants should have or expect to complete a PhD by September 1, 2017.

Applicants should submit their applications electronically by visiting the web site <http://www.gsb.stanford.edu/recruiting> and uploading their curriculum vitae, research papers and publications, and teaching evaluations, if applicable, on that site. **For an application to be considered complete, all applicants must have three letters of recommendation, CV and job market paper submitted by November 15, 2016.** For questions regarding the application process, please send an email to Faculty_Recruiter@gsb.stanford.edu.

Stanford University is an equal opportunity employer and is committed to increasing the diversity of its faculty. It welcomes nominations of and applications from women, members of minority groups, protected veterans and individuals with disabilities, as well as from others who would bring additional dimensions to the university's research, teaching and clinical missions.



上海科技大学
ShanghaiTech University

TENURE-TRACK AND TENURED POSITIONS IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

The newly launched **ShanghaiTech University** is built as a world-class research university, which locates in Zhangjiang High-Tech Park. We invite highly qualified candidates to fill tenure-track/tenured faculty positions as its core team in the School of Information Science and Technology (SIST). Candidates should have exceptional academic records or demonstrate strong potential in cutting-edge research areas of information science and technology. They must be fluent in English. Overseas academic connection or background is highly desired.

Academic Disciplines:

We seek candidates in all cutting edge areas of information science and technology. Our recruitment focus includes, but is not limited to: computer architecture and technologies, nano-scale electronics, high speed and RF circuits, intelligent and integrated signal processing systems, computational foundations, big data, data mining, visualization, computer vision, bio-computing, smart energy/power devices and systems, next-generation networking, as well as interdisciplinary areas involving information science and technology.

Compensation and Benefits:

Salary and startup funds are highly competitive, commensurate with experience and academic accomplishment. We also offer a comprehensive benefit package to employees and eligible dependents, including housing benefits. All regular ShanghaiTech faculty members will be within its new tenure-track system commensurate with international practice for performance evaluation and promotion.

Qualifications:

- A detailed research plan and demonstrated record/potentials;
- Ph.D. (Electrical Engineering, Computer Engineering, Computer Science, or related field);
- A minimum relevant research experience of 4 years.

Applications:

Submit (in English, PDF version) a cover letter, a 2-page research plan, a CV plus copies of 3 most significant publications, and names of three referees to: sist@shanghaitech.edu.cn (until positions are filled). For more information, please visit Job Opportunities on <http://sist.shanghaitech.edu.cn/>

Deadline: September 30, 2016

University of Alabama in Huntsville
Department of Computer Science
Assistant Professor

The Department of Computer Science of The University of Alabama in Huntsville (UAH) invites applicants for a tenure-track faculty position at the **Assistant Professor** level beginning January 2017. The incumbent will augment the department's emphases in at least one of the following areas: cloud computing, particularly secure cloud computing; mobile computing, particularly secure mobile computing; or data science, particularly big data applications. Outstanding candidates who couple cybersecurity with other areas of computing could also be considered.

A Ph.D. in computer science or a closely related area is required. The successful candidate will have a strong academic background, perform funded research, be able to carry out research in areas typical for publication in well-regarded academic conference and journal venues, and be keen on undergraduate education.

The department has a strong commitment to excellence in teaching, research, and service; the hire should have good communication, strong teaching potential, and research accomplishments.

UAH is located in an expanding, high technology area, next door to one of the largest research parks in the nation. Nearby are the NASA Marshall Space Flight Center, the Army's Redstone Arsenal, and many high-tech industries. UAH also

has an array of research centers, including in information technology, modeling and simulation, etc. In short, collaborative research opportunities are abundant, and many well-educated and highly technically skilled people are in the area. There is also access to excellent public schools and housing.

UAH has approximately 8000 students. UAH Computer Science offers the BS, MS, and PhD degrees in Computer Science and the MS and PhD degrees in modeling and simulation. Approximately 200 undergraduate majors and 175 graduate students are associated with the unit. Faculty research interests are many and include cybersecurity, mobile computing, data science, software engineering, visualization, graphics and game computing, multimedia, AI, image processing, pattern recognition, and distributed systems. Recent NSF figures indicate the department ranks 30th in the nation in overall federal research funding.

Interested parties should submit a detailed resume with references to Chair, Search Committee, Dept. of Computer Science, The University of Alabama in Huntsville, Huntsville, AL 35899. Qualified female and minority candidates are encouraged to apply. Initial review of applicants will begin starting in late Summer 2016 and will continue until a suitable candidate is found.

The University of Alabama in Huntsville is an affirmative action/equal opportunity employer/ minorities/ females/ veterans/ disabled.



ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to acmm mediasales@acm.org. Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will be typeset according to CACM guidelines. **NO PROOFS** can be sent. Classified line ads are **NOT** commissionable.

Rates: \$325.00 for six lines of text, 40 characters per line. \$32.50 for each additional line after the first six. The **MINIMUM** is six lines.

Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact:

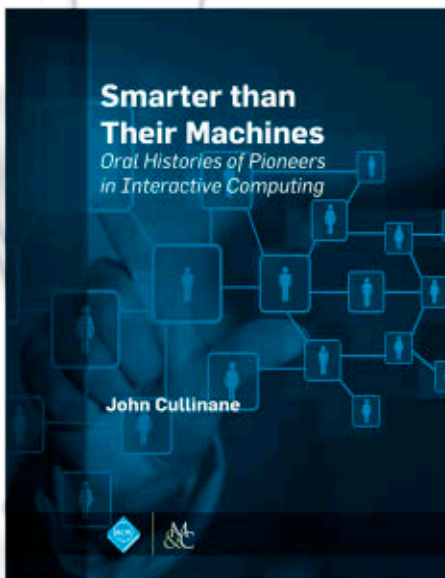
acmm mediasales@acm.org

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at: <http://jobs.acm.org>

Ads are listed for a period of 30 days.

For More Information Contact:

ACM Media Sales
at 212-626-0686 or
acmm mediasales@acm.org



**A personal walk down the
computer industry road.
BY AN EYEWITNESS.**

Smarter Than Their Machines: Oral Histories of the Pioneers of Interactive Computing is based on oral histories archived at the Charles Babbage Institute, University of Minnesota. These oral histories contain important messages for our leaders of today, at all levels, including that government, industry, and academia can accomplish great things when working together in an effective way.



ISBN: 978-1-62705-550-5 DOI: 110.1145/2663015

<http://books.acm.org>

<http://www.morganclaypoolpublishers.com/acm>



The Universität der Bundeswehr München (Bundeswehr University Munich) is significantly expanding its Cyber Defence Research Center (CODE). CODE was established in 2013 with the objective to bring together experts from different faculties and scientific disciplines as well as expertise from industry and government agencies to conduct research in the cyber and information space. CODE pursues a comprehensive, integrated, and interdisciplinary approach to implement technical innovations and concepts for the protection of data, software, and ICT infrastructures in accordance with legal and commercial framework conditions. It has already established important strategic partnerships in this area. The objective of the expansion is to unite the research initiatives of the Bundeswehr and the Federal Government in the area of Cyber Defence and Smart Data and to establish the CODE Research Center as the primary point of contact in the cyber and information domain of the Bundeswehr and the Federal Government.

Research and teaching in the area of cyber security is already being carried out as part of the Bachelor's and Master's programs in the Computer Science Department. According to current planning, a new international Master's program in Cyber Security will be launched on January 1st, 2018.

The Universität der Bundeswehr München will therefore be appointing **eleven Professors** for its Computer Science Department on **October 1st, 2017**.

The Universität der Bundeswehr München is looking for personalities with outstanding scientific qualifications to fill these professorial positions, who will also contribute actively to the CODE research center. Besides excellent research work, the new professors are expected to develop demanding lectures, practicals, and seminars for the new Master's program in Cyber Security and to provide excellent teaching in their respective specialist area. Applicants are also expected to carry out teaching in the Bachelor's programs in Computer Science and Business Informatics, and to work closely with the other departments at the Universität der Bundeswehr München.

The Professorships will be provided with eight excellently equipped laboratories housed in a new building that is to be completed in the near future.

The candidates must have an excellent scientific track record, as demonstrated by a habilitation or equivalent scientific achievements, as well as significant excellent publications in academic journals. Proven teaching experience in their respective specialist area is highly desired. The new Professors should have an international perspective, e.g., based on participation in international research projects, and experience in acquiring third-party funding. The duties will also include active participation in the university's academic self-administration.

The Computer Science Department at the Universität der Bundeswehr München is seeking Professors for the following specialist areas of its Cyber Defence and Smart Data Research Center:

University Professorship (W3) in Cryptography

When it comes to transmitting, storing, and processing data, cryptographic methods are crucial to ensure that the data remains confidential, authenticated, and uncorrupted.

The remit of the Professorship includes encryption algorithms, random number generators and key management as well as their practical application within communication protocols – both from a provider's and an attacker's perspective.

University Professorship (W3) in ICT Threat and Malware Analysis

The complexity and heterogeneity of communication networks and ICT infrastructures requires a systematic assessment of potential attack vectors in order to derive priorities for protection mechanisms.

Besides malware as a mass phenomenon in the form of viruses, Trojan horses and encryption-based ransomware, the Professorship also focuses on constructing sandbox analysis environments to identify malicious code in third-party software, such as smartphone apps and other software downloaded from the web.

University Professorship (W3) in Business Intelligence Security

Networked applications, IT services, and operating environments generate vast amounts of data that can be used for various purposes, including the detection of potential attacks, but the sheer amount of such data prevents any manual processing.

The research tasks involved in the Professorship in Business Intelligence Security include big data and smart data algorithms for the aggregation, the correlation, and the analysis of large data amounts associated with security events for the purpose of providing specific support for decision processes concerning the prevention, the detection, and the response to attacks.

University Professorship (W3) in Cyber Physical System Security

The combination of networked applications with mechanical and electronic components, such as in industrial production facilities, assistance systems, and energy supply systems, has many advantages – but it also involves the risk that these cyber physical systems can be compromised or sabotaged through cyber attacks.

The Professorship focuses on the information security characteristics of cyber physical systems with their specific framework conditions, such as the constrained resources of embedded systems, real-time capabilities, and physical access by attackers.

University Professorship (W3) in Data Protection and Compliance

The development and procurement of complex software systems must comply with legal and sector-specific regulations, which must already be taken into account during the requirements analysis and planning stage. Ultimately, proof of compliance must also be demonstrated, e.g., by means of a certification.

The Professorship in Data Protection and Compliance focuses on the methods and tools needed for the technical implementation of requirements resulting from, for example, the European General Data Protection Regulation and the new German IT Security Act, and it will also develop IT concepts for the implementation of the privacy-by-design paradigm.

University Professorship (W3) in Forensic Methods and IT Security

In order to fully investigate and resolve security incidents, evidence must be gathered and analyzed while maintaining the integrity of the chain of custody. In order to deal with sophisticated attacks that are increasingly leaving fewer traces in compromised IT systems, the concealment of traces, the use of encryption and, for example, requirements regarding the automation and the scalability of digital forensics, the Professorship will research new approaches to analyzing the main and background memories of mobile and stationary systems as well as networked applications, among others topics.

University Professorship (W3) in Open Source Intelligence and Situation Assessment

The situation assessment and documentation in cyber space as a basis for planning and decision processes can no longer be completely achieved using dedicated sensors and traditional reporting channels. Instead, the analysis of public sources, such as social media and internet communities, has become crucial to ensure the timeliness and correct focus of situation reports.

The Professorship in Open Source Intelligence and Situation Assessment will develop methods for continuous information gathering in networked environments using experience and findings from previous operations, as well as methods to compile clear and concise situation reports.

University Professorship (W3) in Privacy Enhancing Technologies

As users often act carelessly with their personal data and internet providers frequently favor commercial interests over protective measures, it is becoming easier and less risky for organized crime to commit, among other things, identity theft.

The Professorship in Privacy Enhancing Technologies will research new approaches to eliminate or at least minimize the unnecessary or unwanted disclosure of information during the use of networked applications and to ensure that stored data is not analyzed and forwarded to third parties without user consent.

University Professorship (W3) in IT Vulnerability Management and Security Testing

Programming errors, insufficient adaptation to new operating environments, and negligent use of IT systems often lead to vulnerabilities that allow attackers to obtain unauthorized access to processed data or even take control of entire systems.

The Professorship in IT Vulnerability and Security Testing will deal with the systematic handling of such vulnerabilities in IT systems and testing methods for their identification and assessment, so that, for instance, penetration tests of networked applications can be employed to determine areas in which the security levels need to be improved.

University Professorship (W3) in Secure Software Development

The security of software crucially depends on the priority assigned to information security attributes during the requirements analysis, system design, and programming as well as the associated testing and approval procedures.

In the Professorship in Secure Software Development, the two disciplines of software engineering and security engineering overlap. It deals with methods, algorithms, and tools required for the implementation of software according to the secure-by-design, secure-by-default, and secure-in-deployment paradigms.

University Professorship (W3) in Usable Security and Privacy

Information security and data protection necessitate the implementation and use of technical mechanisms that are too complex to apply for many users. E-mail encryption procedures, for instance, have only been utilized by IT experts for decades because the time and effort necessary for their use is much too high compared to their subjective benefits.

The Professorship in Usable Security and Privacy will investigate user-friendly approaches to security and data protection procedures and to the implementation of graphical interfaces for human-computer interaction in order to enhance the usability and therefore ensure the extensive employment of important protective measures.

The Universität der Bundeswehr München offers academic programs directed primarily at Officer Candidates and Officers, who can obtain Bachelor's and Master's degrees within a trimester system. Depending on spare capacity, civilian students are allowed to enroll. The study course is complemented by interdisciplinary elements in an integrated program entitled „studium plus“.

Preconditions of employment and the legal duty positioning of Professors are based upon the “Bundesbeamtengesetz”. Employment as a “Beamten/Beamtin” requires that the candidate is not older than 50 at the date of appointment.

The University seeks to increase the number of female Professors and thus explicitly invites women to submit applications. Severely disabled candidates with equal qualifications will receive preferential consideration.

Please submit your application documents marked as Confidential Personnel Matter to the Department Head of the Computer Science Department at *the Universität der Bundeswehr München, 85577 Neubiberg*, by **October 15th, 2016**.

CALIFORNIA STATE UNIVERSITY

EAST BAY

CALIFORNIA STATE UNIVERSITY, EAST BAY

FACULTY EMPLOYMENT OPPORTUNITY

DEPARTMENT OF COMPUTER SCIENCE

FULL-TIME TENURE-TRACK

OAA POSITION NO. 16-17 COMPUTER SCIENCE - TT (2 Positions)

THE UNIVERSITY: California State University, East Bay is known for award-winning programs, expert instruction, a diverse student body, and a choice of more than 100 career-focused fields of study. The ten major buildings of the Hayward Hills campus, on 342 acres, contain over 150 classrooms and teaching laboratories, over 177 specialized instructional rooms, numerous computer labs and a library, which contains a collection of over one million items. The University also has campuses in Contra Costa County, Online, and in Oakland, California. With an enrollment of approximately 13,000 students and 600 faculty, CSUEB is organized into four colleges: Letters, Arts, and Social Sciences; Business and Economics; Education and Allied Studies; and Science. The University offers bachelor's degrees in 50 fields, minors in 61 fields, master's degrees in 37, and 1 doctoral degree program. <http://www.csueastbay.edu/>

THE DEPARTMENT: The Department of Computer Science has 13 full-time faculty members, with a wide range of backgrounds and interests. The faculty is committed to teaching its undergraduate and Master's level students. In a typical quarter, the Department will offer over 30 undergraduate and about 20 graduate classes. Classes are offered both in day and evening. Classes are generally small, with many opportunities for faculty-student contact. The Department offers a variety of degrees: B.S. in Computer Science (with possible options in Networking and Data Communications, Software Engineering, or Computer Engineering), and M.S. in both Computer Science and Computer Networks. Currently, there are more than 400 undergraduate majors and over 250 students in the M.S. programs.

DUTIES OF THE POSITION (2 positions currently available): Teaching courses at B.S. and M.S. levels, curriculum development at both levels, and sustaining a research program. Please note that teaching assignments at California State University, East Bay include courses at the Hayward, Concord and Online campuses. In addition to teaching, all faculty have advising responsibilities, assist the department with administrative and/or committee work, and are expected to assume campus-wide committee responsibilities. The ideal candidate for this position is able to:

1. Teach a wide range of computer science courses including most or all of the core subject matter at both the undergraduate and graduate level. Preference will be given to those with expertise in emerging areas.
2. Support offerings for undergraduate C.S. students including teaching courses, developing the undergraduate curriculum, and engaging undergraduate students in research.
3. Support offerings for graduate C.S. students including teaching courses, guiding M.S. theses, developing the graduate comprehensive examination, etc.
4. Advise Computer Science students.

5. Participate in departmental activities such as curriculum development, assessment, outreach, etc.
6. Develop and continue ongoing research activities, service and leadership.

RANK AND SALARY: Assistant Professor. Salary is dependent upon educational preparation and experience. Subject to budgetary authorization.

DATE OF APPOINTMENT: Fall Quarter, 2017

QUALIFICATIONS: Applicants must have a Ph.D. in Computer Science by September 2017. Applicants must be able to teach undergraduate and master's level courses in most of the standard computer science core subjects. Candidates should demonstrate experience in teaching, mentoring, research, or community service that has prepared them to contribute to our commitment to diversity and excellence. Additionally, applicants must demonstrate a record of scholarly activity. Candidate's accomplishments should be commensurate with their professional level.

This University is fully committed to the rights of students, staff and faculty with disabilities in accordance with applicable state and federal laws. For more information about the University's program supporting the rights of our students with disabilities see: <http://www20.csueastbay.edu/af/departments/as/>

APPLICATION DEADLINE: The deadline for applications is October 31, 2016; review of applications will begin November 1, 2016. The position, however, will be considered open until filled. Please submit a letter of application, which addresses the qualifications noted in the position announcement; a complete and current vita; names and email addresses of three references, three letters of recommendation, a statement of teaching philosophy, and evidence of teaching and research abilities.

Note: California State University, East Bay hires only individuals lawfully authorized to work in the United States. All offers of employment are contingent upon presentation of documents demonstrating the appointee's identity and eligibility to work in accordance with provisions of the Immigration Reform and Control Act. A background check (including a criminal records check and prior employment verification) must be completed and cleared prior to the start of employment.

For instructions on how to apply, please visit:

<https://help.interfolio.com/hc/en-us/articles/203701176-Job-Applicant-s-Guide-to-ByCommittee-Faculty-Search>

As an Equal Opportunity Employer, CSUEB does not discriminate on the basis of any protected categories: age, ancestry, citizenship, color, disability, gender, immigration status, marital status, national origin, race, religion, sexual orientation, or veteran's status. The University is committed to the principles of diversity in employment and to creating a stimulating learning environment for its diverse student body.

[CONTINUED FROM P. 104] really bother us. In another sense it doesn't work, because 99% of the email that's transited is still spam, and that would only be true if people were actually clicking and buying.

So what makes spam possible?

The fascinating part about spam filters is that they are part of the solution and part of the problem. For drugs, 30% of revenue comes from people who actively go into their spam folders, find the Viagra spam, and click on it. So their spam folder is a classifier.

What happens after someone clicks? Did the drugs you bought really show up?

Without fail.

Were they what they claimed to be?

We analyzed a subset. We didn't do a hardcore chemical assay, but we did put it under a mass spectrometer and look at the distribution of the underlying chemicals. And when you match up the spectral patterns of the counterfeit stuff and the reference stuff, there's no difference.

So counterfeit drug producers are not interested in ripping off their customers.

Right. We were also privy to a variety of leaked email and text messages, and to a first approximation, these people do not think of themselves as criminals. Much of the counterfeit drug market is eastern European, where there's a certain worldview that intellectual property protection is a tool of the bourgeois West, and that in fact they are shipping a quality product, lowering prices, and satisfying a need.

The people doing credit card theft think of themselves as criminals. They might justify it by saying that their victims have so much money. But a lot of these other scammers don't take any heart in their status as outlaws. They think of themselves as businesspeople.

In the end, you discovered that 95% of pharmaceutical and software counterfeiters relied on a handful of banks, and you were able to work with Visa and MasterCard to shutter their accounts. What has happened since?

"The fascinating part about spam filters is that they are part of the solution and part of the problem."

Everywhere these techniques have been applied, there's been a serious impact to the ecosystems. It has also shifted substantially where the money goes. In software, drugs, and especially in counterfeit goods like Gucci handbags, it's all moving to the Bank of China. And that remains a bit of an open question. China is in a special position vis-à-vis the financial world. It's a little different when it's some small Azerbaijani bank.

You have also done some work in the automotive sector.

This notion that your car is a Henry Ford-style thing that happens to have a computer in it turns out to be totally wrong. The mental model everyone should have is that the most complicated distributed system you use is the one you drive in, and it happens to have wheels on it.

Here, too, your work was informed by your attempts to understand the broader structure of the automotive industry.

Far too often, when people do this kind of vulnerability research, there is a tendency to name and shame, and I'm not convinced that it does anyone any good. The most interesting part of our research was not the technical details—it wasn't like there was a class of vulnerability that no one had ever heard of. What was eye-opening was trying to figure out why we found what we found.

Most of the vulnerabilities you discovered were at the interface between two

pieces of code that were written by different organizations.

One of the things that surprised me the most is that no one has access to the source code for a car. The modern automotive industry has a very deep supply chain, and the OEMs are really just integrators. The suppliers own the IP, and they are not about to give the source code to the OEM. And that happens on down the chain: the suppliers have sub-suppliers, and there's no one who can look through the final product. There is also no agency in the government that's set up to audit the source code of automobiles.

The other thing I thought was fascinating is that there were bugs that were really egregious. They were the kinds of bugs you could not get away with if you worked at a company that sold PC software. You would not be allowed to use those functions because they were so known to be prone to fragility.

But in a certain sense, automotive software is like PC software from the early 1990s, before the Internet took off. The thing that made PC software better is that people started attacking it. From a pure Darwinian sense, it had to get better, or we couldn't use it anymore. Thus far, most embedded systems just haven't had an adversary. Security costs money, which is really hard to justify when you don't have an adversary out there forcing your hand. 

Leah Hoffmann is a technology writer based in Piermont, NY.

Q&A

Hello, Underworld

Stefan Savage's innovative research has focused on strengthening the security, privacy, and reliability of networks.

CONSIDER THE SPAMMER: that is what University of California, San Diego (UCSD) professor and this year's ACM-Infosys Foundation Award recipient Stefan Savage did nearly 10 years ago, when he began to expand his research beyond the technical aspects of network security. What he found was not just a fascinating glimpse into an underground community and a dynamic global marketplace; he also gained insights that led to valuable techniques for combating spam, reducing vulnerabilities in automotive software, and strengthening cybersecurity.

You have spent the bulk of your career in network security. What sparked your interest?

When I got to UCSD, I was working on network protocols, and for a variety of self-serving reasons, I began wondering if you could abuse those protocols to do different kinds of measurements. It turns out that you could.

That is what opened your eyes?

All of this stuff presumes that everyone is well behaved, and if you're not, you can do all kinds of strange things. It's almost like those sci-fi horror movies where you get the special glasses and you can see aliens. You start looking at the world like an adversary.

This was during the Internet worm outbreak era, when worms were taking over hundreds of thousands of machines.



"At the end of the day, the only reason you care about security is that you actually think someone is out to get you."

For totally serendipitous reasons, we ended up having an amazing monitoring capability at UCSD where we could measure how a worm was spreading over the Internet and look at different DOS attacks. I started working with Geoff Voelker and Vern Paxson, and we did a lot of purely technical work until about 2007.

Since then, you have taken a more holistic approach, in which you look beyond the technical components of computer security to economic and social motives.

At the end of the day, the only reason you care about security is that you actually think someone is out to get you. But that's not always reasonable. No one is hacking my kid's Barney the Dinosaur doll; there's no incentive. All of the conflict we have in the cyber realm is conflict that just generally exists. It's not like if there weren't computers, people wouldn't still want your money or want to steal your intellectual property. It's just that that's the medium through which they can do it in the 21st century, given that we've shoved a bunch of money and valuable information online.

One of the first domains you looked into using this lens was spam.

When we began our research, we realized that we tend to think about spam as a filtering problem: how do I recognize the email that I want from the email that I don't want? To a certain extent that works—most of us have a modest amount of spam, and it doesn't [CONTINUED ON P. 103]

Call for Nominations

The ACM Doctoral Dissertation Competition

Rules of the Competition

ACM established the Doctoral Dissertation Award program to recognize and encourage superior research and writing by doctoral candidates in computer science and engineering. These awards are presented annually at the ACM Awards Banquet.

Submissions

Nominations are limited to one per university or college, from any country, unless more than 10 Ph.D.'s are granted in one year, in which case two may be nominated.

Sponsorship

Each nomination shall be forwarded by the thesis advisor and must include the endorsement of the department head. A one-page summary of the significance of the dissertation written by the advisor must accompany the transmittal.

Deadline

Submissions must be received by **October 31, 2016** to qualify for consideration.

For Eligibility and Submission Procedure see http://awards.acm.org/doctoral_dissertation/

Publication Rights

Each nomination must be accompanied by an assignment to ACM by the author of exclusive publication rights. (Copyright reverts to author if not selected for publication.)

Publication

Winning dissertations will be published by ACM in the ACM Books Program and appear in the ACM Digital Library. Honorable mention dissertations will appear in the ACM Digital Library.

Selection Procedure

Dissertations will be reviewed for technical depth and significance of the research contribution, potential impact on theory and practice, and quality of presentation. A committee of individuals serving staggered five-year terms performs an initial screening to generate a short list, followed by an in-depth evaluation to determine the winning dissertation.

The selection committee will select the winning dissertation in early 2017.

Award

The Doctoral Dissertation Award is accompanied by a prize of \$20,000 and the Honorable Mention Award is accompanied by a prize of \$10,000. Financial sponsorship of the award is provided by Google.



Association for
Computing Machinery



SIGGRAPH
ASIA 2016
MACAO

SA2016.SIGGRAPH.ORG

Register online by
15 October 2016, 15:00 UTC/GMT
& enjoy early bird discounts of up to **20%!**

👉 sa2016.siggraph.org/registration



KEY TO THE FUTURE

CONFERENCE: 5 - 8 DECEMBER 2016

EXHIBITION: 6 - 8 DECEMBER 2016

THE VENETIAN MACAO, **MACAO**

Sponsored by

