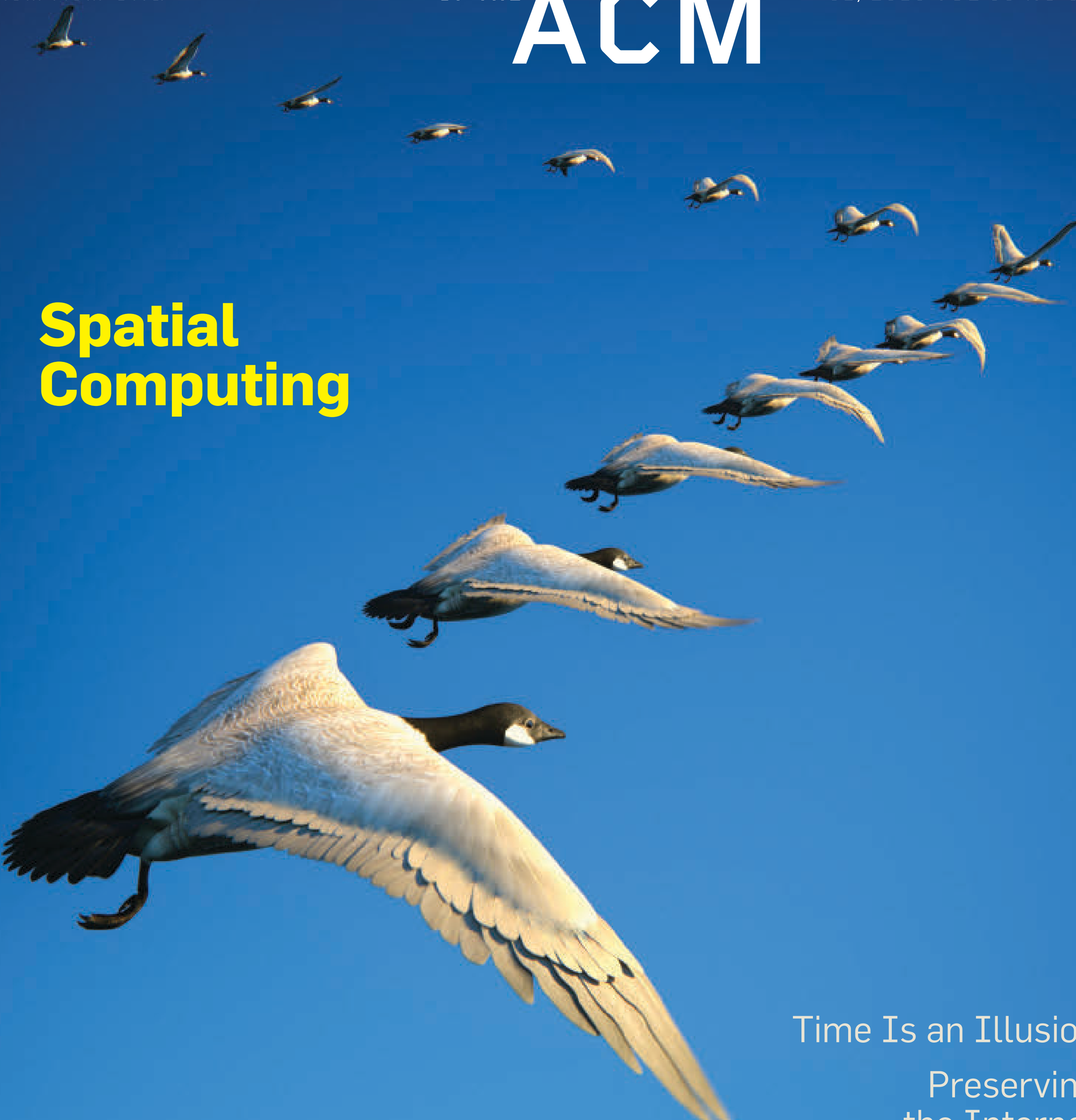
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

01/2016 VOL.59 NO.01



Spatial Computing

Time Is an Illusion

Preserving the Internet

The Chaos Machine

ACM's FY15 Annual Report

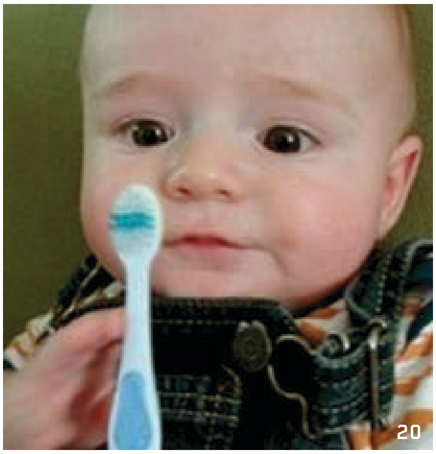
Association for Computing Machinery

COMMUNICATIONS OF THE ACM

Departments

Last Byte

News

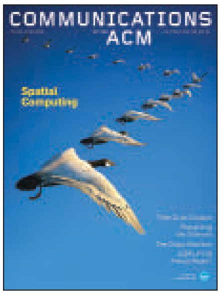

20

Viewpoints

Association for Computing Machinery
Advancing Computing as a Science & Profession

IMAGE COURTESY OF STANFORD.EDU

01/2016
VOL. 59 NO. 01



About the Cover: The integration of spatial computing into our everyday lives has been transforming, allowing us to better understand and visualize our relationship to locations and how to navigate through those locations in a far more efficient manner. This month's cover story explores the opportunities opening throughout the spatial computing landscape. Cover illustration by Peter Crowther Associates.

IMAGES FROM SHUTTERSTOCK.COM

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

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

CONTACT POINTS

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

WEBSITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/

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

Media Kit acmmediasales@acm.org

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

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $100.

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

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

COMMUNICATIONS OF THE ACM

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

POSTMASTER

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

Printed in the U.S.A.

Association for Computing Machinery

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/2851202

Mark R. Nelson

Focusing on Teacher Needs in K–12 CS Education

IT IS TRULY a great time to be in CS education, and yet CS educational needs in K–12 are greater today than ever before. A lack of access to CS among a range of demographics, including Hispanic, female, and lower-income students continues to persist. According to respondents to a recent Google and Gallup study, fewer than half of the students in these demographic groups identify any exposure to CS in school and only 1 in 4 responding schools reported having someone to teach CS. The research found general confusion as to what CS is among students, parents, and educators, with stereotypes existing at many levels.

Add to this mix a dedicated cadre of CS teachers. The 8th-grade music teacher, for instance, teaching CS for the first time this year and trying to learn where to start. The high school Spanish teacher, who is so passionate about CS, that to introduce CS into her school, she teaches CS in Spanish. The high school math teacher who has taught CS in his school for years, but lacks the confidence in his own CS skills to make substantive changes to the projects and programs he uses in class. These are but three examples of real CS teachers who are among the faces of K–12 CS education today.

Enter the Computer Science Teachers Association (CSTA). Founded by ACM just over a decade ago, CSTA now has more than 22,000 members across 130 countries. This includes more than 60 local member chapters in North America, and three international affiliates (U.K., Israel, and New Zealand). The support of ACM and its members enables CSTA to fulfill its mission.

So what is CSTA doing or have planned? Building on our strengths and listening to the needs of the K–12 CS teacher community, we will grow our capacity and programs in five interrelated areas: operations, diversity, professional development, research, and awareness.

Operations: In the coming weeks the first phase of our new website and a new association management system will go live. These establish a foundation to support improved communication, data analytics, communities of practice, grassroots advocacy, and other engagement initiatives. These are enhancements many of us look forward to seeing implemented, as they will increase our ability to support and engage our members.

Diversity: Many groups are addressing diversity in CS, but much more needs to be done. CSTA's unique contribution will be the Diversity Educational Leadership Program (DELP). DELP will seek to identify the mentoring, professional development, and support needs of teachers from underrepresented backgrounds or working with students from underrepresented groups. We will then work with others in this space to help meet those needs. The goal is to help these teachers be more successful, and enable them to become teacher leaders or advocates for CS education within their classrooms, communities, and school districts.

Professional Development (PD): CSTA remains committed to the K–12 CS teaching standards. We also plan to raise the bar on current PD offerings, such as the annual conference, and explore new PD opportunities, such as online programs, webinars, and increased PD resources for chapters and international members.

Research: Building on CSTA's existing research experience, we plan to continue to pursue research that can generate direct outcomes for the K–12 CS teacher. Sample research topics on our priority list include work related to learning assessment, pedagogy, diversity, understanding different groups of teachers (for example, new CS teachers), and innovative approaches to teacher PD.

Awareness: We will engage in more strategic outreach to the media and potential partner organizations, so that we may build awareness and diversify support for CSTA programs, and educate other groups on the nature and diversity of CS, both in terms of careers and skills. This includes promoting opportunities such as the Cutler-Bell Award, the NSA Day of Cyber, the Congressional App Challenge, the Allen Distinguished Educators Awards, the Faces of Computing Competition, and programs with partners, such as Google, College Board, Oracle Academy, and Pluralsight, among others.

Yes, it is a great time to be involved with K–12 CS education. CS is emerging as the lingua franca of the evolving world. As Jane Margolis argues in her research, access to this universal language is fundamental to future prosperity and participation in society. Computational technology is increasingly ubiquitous, and will continue to accelerate the pace of knowledge creation and societal change across the globe. Helping teachers helps students. CSTA looks forward to continued collaboration with ACM staff, volunteers, and partners as we work to support, empower, and grow the K–12 CS teacher community.

Mark R. Nelson (m.nelson@csta-hq.org) is Executive Director of the Computer Science Teachers Association.

DOI:10.1145/2854884

Vinton G. Cerf

On the Road in Latin America

It is a new year and 2020 is now only four years away. For science fiction fans like me, it feels as if we have been transported by a time machine into the future!

For the past month, I have been on the road in Latin America visiting Brazil (João Pessoa), Colombia (Bogota), Argentina (Buenos Aires), Chile (Santiago and Punta Arenas), and Uruguay (Montevideo). The continent is changing. There is more Internet access than ever before and mobiles are getting increasingly smart. In some places, the Internet penetration reaches more than 70%. The academic community in many countries here has been immersed in the Internet since the late 1980s, but it is clear there is an awakening awareness of the utility of the Internet across the continent. Some of this can be attributed to the rapid expansion of mobile access through smartphones and some to the arrival of broadband services to the business community and, increasingly, residential settings. In Montevideo, for example, we tested wireless broadband speeds of 20Mb/s down and 8–9Mb/s up just while driving around town.

While attending the Internet Governance Forum in João Passoa, Brazil, I had the pleasure of using an O3B ("Other Three Billion") ground station that delivered 400Mb/s down and 100Mb/s up in a hospitality tent adjacent to the conference facility. The system was installed in a few days and the model used was actually for shipboard applications requiring stabilizers for antenna tracking of the asynchronous, equatorial 8000KM orbit satellites.

I had the opportunity to meet with very senior officials in many of the countries I visited including President Michele Bachelet of Chile, newly elected President of Argentina Mauricio Macri, Vice President Raul Sendic of Uruguay and Simón Gaviria, Director of National Planning Development of Colombia. I met with regulators and with ministers or vice ministers of telecommunications, innovation, and information technology. Everywhere, I came away with a sense there is strong interest in Internet use in business, education, government services, and applications for the general public. Social networking uses are rising rapidly and, as in other parts of the world, smartphone applications are proliferating.

There were common themes in my many meetings: interest in domestic and global businesses facilitated by the Internet, analysis of conditions necessary for the creation and sustained growth of new business, curiosity and concerns for the way technology can disrupt old business models and challenge leaders to reinvent their companies. The future of print journalism and the arrival of Uber were topics that sparked much dialogue. Education was a common focus in every country. In Uruguay I met elementary school students, each of whom had a laptop computer (from the One Laptop Per Child program led by Nicholas Negroponte). They were using a mathematics program developed in Germany that allowed each student to progress at his or her fastest rate. When I asked about ideas for the future, one young boy said we should set up a center for ideas that can change the world—not a bad idea in itself! There is no question that every leader I met was fully aware of the importance of education at all levels to create and sustain old and new businesses.

Those of you who know me will be aware of my interest in wine and this little report about travel in Latin America would be seriously incomplete without an observation about the wines here. There are two that are particularly notable: Carmenere, of French origin, was largely lost in France in the mid-19th century and was rediscovered and nurtured in Chile more recently. Tannat is a varietal that was introduced in the late 19th century in Uruguay and was new to me.

I had the opportunity to visit several vineyards and in particular the famous Concha y Toro vineyard in Chile where the flagship label is Don Melchor. Our party met with the wine maker, Enrique Tirado, and we prepared a unique cuvee from four-barrel samplings of cabernet sauvignon and cabernet franc. The variable dimensions of wine making rival any large complex program's space of variable values and my hat is off to the wizards of wine who produce consistent products year after year. In Uruguay, I was able to visit both the Bouza and the Juanico wineries and vineyards, both not far from Montevideo. It would be difficult to overstate the remarkable quality of the tannat grape and the blends that lend lush and round flavors that challenge the best cabernet sauvignons I have tasted.

From João Pessoa to Punta Arenas, this was a trip to remember, and to recommend to you as well. Happy New Year!

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

DOI:10.1145/2851198

Let the Liable Pay

In his letter to the editor "Who Pays When Autonomous Fails?" (Nov. 2015) on Keith Kirkpatrick's news story "The Moral Challenges of Driverless Cars" (Aug. 2015), Hans Grünberger attacked my proposal for ethics review boards (http://for.tn/1NHHU1s), implying I intend certification by such boards as a mechanism for shielding software companies and vehicle manufacturers from liability for the inevitable property damage, injuries, and deaths that will result from even the best-designed autonomous vehicles. This mischaracterizes my beliefs and my proposal.

I did not (and do not) intend such boards to be a shield. On the contrary, software companies and vehicle manufacturers should be liable for the mishaps, great and small, of their products (they will then price accordingly). But the necessary ethical decisions will either be made by commercial interests in a vacuum, as generally happens today, or a way will be found to include the public interest and professional expertise, as in the mechanism I proposed—review boards comprised of manufacturers, lawyers, ethicists, and government entities. Likewise, in a related area—the basic safety of these vehicles—government regulation is necessary (http://huff.to/1MWByHW), and I testified to this effect (http://bit.ly/1Rn2Wlt) before the California Department of Motor Vehicles on January 2015, to the likely dismay of the automaker and Google representatives in attendance.

Jonathan Handel, Los Angeles, CA

Start with a Dependable Publishing Schedule

In their column "Should Conferences Meet Journals and Where? A Proposal for 'PACM'" (Sept. 2015), Joseph A. Konstan and Jack W. Davidson proposed a journal series, to be called *Proceedings of the ACM*, that would include papers from the proceedings of ACM's most prestigious conferences, recognizing conference papers are viewed by some as second class when compared to journal papers. Supporting the proposal, the column said many scientists prefer conferences due to their up-to-date take on research and that journal limitations (such as the slowness of publishing) no longer trouble potential authors. Merging both proceedings and journals thus seems a natural approach to balance the computer science community's competing interests.

A first step in this direction could instead be to reduce the overly long evaluation response schedules characteristic of iconic CS journals by establishing rigorous deadlines on the revision and editing process, from submission to publication or rejection. In my limited experience, the slow speed of journal publishing is indeed still an issue. Authors are often motivated to submit their papers to conferences just because of their shorter and more certain response times. If this assumption is correct, reducing evaluation response and editing times would initially alleviate, if not ultimately solve, the issues Konstan and Davidson outlined in proposing PACM.

Natale Patriciello, Maranello, Italy

Hold the Braces and Simplify Your Code

A. Frank Ackerman's letter to the editor "Ban 'Naked' Braces!" (Oct. 2015) actually frightened me. The problem it described—code blocks so long and complex it is difficult for even expert programmers to determine which brace closes them—is certainly an issue, but the proposed solution would not leave the code any less complex.

Please write small, simple blocks of code and save yourself having to worry about naked braces.

When dealing with legacy code, maintainers can add annotations to a long method's closing brace, as described, to help them navigate, but this should never be necessary in new code.

Modern code-writing techniques encourage small and only minimally nested blocks. See Robert C. Martin's *Clean Code* chapter 3 (Prentice Hall, 2009) for a good explanation and justification. According to Martin, and others, a function or method spanning more than a few lines is too much for an average programmer to absorb and understand quickly, especially in the context of a larger system, and should be refactored for simplicity. Blocks nested more than two levels will not fit in a short, readable function and should thus be refactored to decrease depth. Programmers who follow these practices cannot possibly lose track of closing braces and will not then need to mark them up.

I make a living reading and trying to understand unnecessarily complex code, and follow Martin's philosophy. If one writes code complex enough to require annotated closing braces, annotating those braces will not make the code less complex.

If you are a programmer, please write small, simple blocks of code and save yourself having to worry about naked braces. And if your habits run too deep, please do not encourage them in others, especially beginners.

Jamie Hale, Markham, Ontario, Canada

A. Frank Ackerman's letter to the editor (Oct. 2015) recommended tagging each construct terminator with a comment indicating which construct it terminates. This wise practice was carried further in the design technique described in the book *Principles of Program Design* (Academic Press, 1975) and the paper "Constructive Methods of Program Design" (LNCS 44, Springer, 1976). Each sequence, selection, and iteration control construct is named. The name is prefixed to start and end markers and, in a selection, to each alternative. Ackerman's coding example might thus be expanded to the following skeleton:

```
account seq
  ...
  activity itr (...)
    transaction sel (> $1000)
      ...
    transaction alt (<= $1000)
      ...
    transaction end
  activity end
  ...
account end
```

Such a discipline not only avoids simple errors. As Ackerman hinted, it also improves the program by tying its code closely to its design.

Michael Jackson,
Milton Keynes, England

The Hypothetical 5% Expert Programmers

Jack Ring's letter to the editor "Give Me the 5% Capable of Incisive Critique" (June 2015) downplayed the concerns I raised in my earlier letter "The Case of the Missing Skillset" (Apr. 2015) on Michael Walfish and Andrew J. Blumberg's article "Verifying Computations without Reexecuting Them" (Feb. 2015). The letter said all IT management needs is a certain "... 5% of the workforce at the level of expert critic ...," a statistic unsupported by any reference. Knowing its source might help us understand the deductive process that led Ring to this particular percentage. The letter further said companies need "... critical attitude and a tool that tells them whether a program or combination of programs violates established rules ..." This is beside the original point about a general lack of availability of capable expert programmers rather than the availability of automated tools.

To be practical, the letter's claim would have to satisfy three requirements: a "critical attitude," something no one would argue against, though an overly critical attitude can potentially cause the demise of an organization; a tool; and "established rules." While academic tools for formal verification are readily available, an implied assumption is there is a certain set of established rules that would be common to every type of software project. Moreover, these rules would be unlike the rules traditionally enforced by compilers, linkers, or associated virtual machines. It would again be nice see a reference to a formal proof demonstrating the universality of this hypothetical set of rules for all software projects.

In case this principle is indeed not universal but rather project-specific, programmers would have to discover and write them in some formal language from user requirements. It would be only after such a step that a tool could be used for formal verification. Moreover, even after the tool would identify errors, the cycle would begin again, with programmers correcting the code, re-verifying, and so on. All these activities cost time and resources, which are always in limited supply. It also proves the point I made in my original letter—that there are not enough programmers with the skills needed to write efficient and verifiable parallel programs.

The letter's hypothetical work setting could therefore possibly involve extended development time, an inflated budget, and delayed projects, factors all known to contribute to an organization's potential failure. Most small-to-mid-size software companies cannot afford such a development life cycle. To the best of my knowledge, the only types of organization that might sustain it would perhaps be an academic institution, research-focused enterprise, or company involved in real-time or safety-critical applications. Perhaps we should therefore take the time to re-evaluate and revise the ACM/IEEE computing curricula to ensure a plentiful supply of future expert programmers trained in these areas.

Muaz A. Niazi, Islamabad, Pakistan

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.

Calendar of Events

January 4–6
IMCOM '16: The 10th International Conference on Ubiquitous Information Management and Communication
Danang, Vietnam
Contact: Sukhan Lee
Email: lsh@ece.skku.ac.kr

January 10–13
SMA '16: The 4th International Conference on Smart Media and Applications
Danang, Vietnam
Contact: Youngchul Kim
Email: yckim@chonnam.ac.kr

January 14–16
ITCS '16: Innovations in Theoretical Computer Science
Cambridge, MA
Contact: Shafi Goldwasser
Email: shafi.goldwasser@gmail.com

February

February 14–17
TEI '16: The 10th International Conference on Tangible, Embedded, and Embodied Interaction
Eindhoven, the Netherlands
Contact: Saskia Bakker
Email: s.bakker@tue.nl

February 21–23
FPGA '16: The 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays
Monterey, CA
Contact: Deming Chen
Email: dchen@illinois.edu

Feb 22–25
WSDM 2016: The 9th ACM International Conference on Web Search and Data Mining
San Francisco, CA
Contact: Paul N. Bennett
Email: paul.n.bennett@microsoft.com

February 27–March 2
CSCW '16: Computer Supported Cooperative Work and Social Computing
San Francisco, CA
Contact: Meredith Ringel Morris
Email: merrie@microsoft.com

On Diversity and ACM's 2016 General Election

As the ACM election process begins its biannual schedule, I want to draw attention to the importance of thinking about diversity in our elected bodies. This consideration should be a matter of interest, not only to the nominations committee that I chair this year, but to the electorate as well.

In the past several years, ACM has invested time and effort to become more geographically inclusive through the creation of the ACM-Europe, ACM-China, and ACM-India Councils. We have more work to do along those lines in other regions of geographic importance. Equally important are other axes of diversity including gender and ethnicity. We want our elected officials to reflect the diversity of the computing industry and, frankly, we want to help increase the diversity in this sector. The statistics of involvement in computing are notably out of balance with too few women and too little representation among ethnic groups. The computing community will be well served to increase the participation of diverse constituents and that is equally true of ACM's elected and appointed positions.

I hope you will take these points into account as you consider recommendations for nominees for our elected posts, our awards, and member advancement opportunities. Similarly, as voting begins, please keep in mind the value of diversity on our boards and councils.

Vinton G. Cerf,
PAST PRESIDENT,
CHAIR, NOMINATING COMMITTEE

ACM's 2016 General Election

ACM Nominating Committee Report Nominees for the 2016 General Election

In accordance with the Constitution and Bylaws of the ACM, the Nominating Committee hereby submits the following slate of nominees for ACM's officers. In addition to the officers of the ACM, five Members at Large will be elected. The names of the candidates for each office are presented in random order below:

President (1 July 2016 – 30 June 2018):
Vicki L. Hanson, Rochester Institute of Technology and University of Dundee
Erik Altman, IBM T.J. Watson Research Center

Vice President (1 July 2016 – 30 June 2018):
Fabrizio Gagliardi, Polytechnic University of Catalonia
Cherri Pancake, Oregon State University

Secretary/Treasurer (1 July 2016 – 30 June 2018):
Craig Partridge, Raytheon BBN Technologies
Elizabeth Churchill, Google

Members at Large (1 July 2016 – 30 June 2020):
Susan Dumais, Microsoft Research
Eugene H. Spafford, Purdue University
Gabriele Anderst-Kostis, Johannes Kepler University
Pam Samuelson, University of California, Berkeley, School of Law
Elizabeth Mynatt, Georgia Tech
Manindra Agrawal, IIT, Kanpur
Yunhao Liu, Tsinghua University
Paul Spirakis, University of Liverpool
Asu Ozdaglar, Massachusetts Institute of Technology
Judith Gal-Ezer, The Open University of Israel

The Constitution and Bylaws provide that candidates for elected offices of the ACM may also be nominated by petition of one percent of the Members who as of **1st November** are eligible to vote for the nominee. Such petitions must be accompanied by a written declaration that the nominee is willing to stand for election. The number of Member signatures required for the offices of President, Vice President, Secretary/Treasurer and Members at Large, is **729**.

The Bylaws provide that such petitions must reach the Elections Committee before **31st of January**. Original petitions for ACM offices are to be submitted to the ACM Elections Committee, c/o Pat Ryan, COO, ACM Headquarters, 2 Penn Plaza, Suite 701, New York, NY 10121, USA, by 31 January 2016. Duplicate copies of the petitions should also be sent to the Chair of the Elections Committee, Gerry Segal, c/o ACM Headquarters. All candidates nominated by petition are reminded of the requirements stated in the Policy and Procedures on Nominations and Elections that a candidate for high office must meet in order to serve with distinction. This document is available on http://www.acm.org/about/acm-policies-procedures, or copies may be obtained from Rosemary McGuinness, Office of Policy and Administration, ACM Headquarters. Statements and biographical sketches of all candidates will appear in the May 2016 issue of *Communications*.

The Nominating Committee would like to thank all those who helped us with their suggestions and advice.

Vinton G. Cerf, CHAIR,
Michel Beaudouin-Lafon, Jennifer Chayes, PJ Narayanan, and Doug Terry

DOI:10.1145/2847218

ACM's Annual Report for FY15

As I think back over the last year, I find it a fantastic time to be a member of the world's largest professional and student membership society in computing. We continue to build on the services and resources our volunteers and staff provide to the worldwide community, from the foremost collection of computing literature to the forefront of computing education and professional development, from expert advice on public policy and professional ethics to the advancement of a balanced and representative work force.

Also as I think back over the last year, I note it was a year of important firsts.

ACM presented its first million-dollar A.M. Turing Award, bringing the financial clout of this renowned honor in computing in line with the world's most prestigious cultural and scientific prizes. Thanks to the generous support of Google, this new monetary level raises the Turing Award's visibility as the premier recognition of computer scientists and engineers who have made contributions of enduring technical importance to the computing community.

ACM hosted its first conference targeted specifically at the computing practitioner community. Indeed, the Applicative conference was so successful we expect its successor, slated for June, to double in attendance. In addition, ACM recently introduced its first *acmqueue* app—an interactive, socially networked, electronic magazine designed to more easily reach where its audience now lives and works.

ACM established its first award to recognize the computing talents of high school seniors. The ACM/CSTA Cutler-Bell Prize in High School Computing seeks to promote and encourage computing as a profession as well as empower aspiring learners to pursue computing challenges outside the classroom.

And, for the first time in 17 years, ACM welcomed a new executive director and CEO. Bobby Schnabel has a long history of service to the ACM and the computing community and I am confident he will support ACM with incredible passion and innovative ideas.

In the following pages you will find a year filled with determined initiatives, new services, and "firsts" designed to increase the value of your ACM membership.

In the following pages you will find a year filled with determined initiatives, new services, and "firsts" designed to increase the value of your ACM membership. But accomplishments are only half the story. There is still much work to do. Issues related to open access and digital library sustainability remain high priorities for ACM's leaders, where we have introduced several new options requested by our members. We continue to explore ways to better integrate the DL with other services and value-added content that might create the foundation for a new revenue model. We are exploring new ways to reach an ever-broadening and diverse audience; look for ACM's first participation in the annual South-by-Southwest (SXSW) event in March. And ACM's transformation into a global professional society remains high on our agenda.

In the coming year ACM will seize more opportunities, explore more member benefit services, and face new challenges. And we are ready. As always, we look to our devoted volunteers, members, and industry partners for the advice and guidance to keep us steadily moving forward.

Alexander L. Wolf, ACM PRESIDENT
JULY 2014–JUNE 2016

ACM's Annual Report for FY15

Highlights of ACM Activities: July 1, 2014–June 30, 2015

***ACM**, the Association for Computing Machinery, is an international scientific and educational organization dedicated to advancing the arts, sciences, and applications of information technology.*

Publications

ACM's Publications Board seeks to maintain and strengthen the association's position as the leading provider of research information and related services for the scholarly computing community and to become a leading provider of advanced-level information and services for the broader computing industry.

To achieve these goals, ACM must (1) maintain and strengthen the quality of its current publication portfolio, (2) accelerate the growth of its program into new areas of computing and established areas where ACM has not historically focused in order to provide a wider range of affordable high-quality publication venues for both the research and practitioner communities, (3) establish itself as a leader in publication technology and innovation with new products and services for the community, and lastly to (4) take a leadership role in driving publishing industry innovation for the benefit of the broader scientific community.

The cornerstone of ACM publications is the ACM Digital Library (DL) serving as the primary distribution mechanism for all the association's publications as well as host to scientific periodicals and a set of conference proceedings from external organizations. With an estimated five million users worldwide, ACM's DL is available at 2,650 institutions in 190 countries. An additional 31,000 individual subscribers in 196 countries have DL access. The result of this widespread availability led to more than 20 million full-text downloads during FY15.

ACM is committed to always increasing the scope of material available via the DL. Last year, over 26,000 full-text articles were added, bringing total DL holdings to 443,000 articles. ACM's *Guide to Computing Literature* is also integrated within the DL. More than 76,000 works were added to the bibliographic database in FY15, bringing the total *Guide* coverage to more than 2.1 million works.

ACM is the publisher of 83 periodicals, including 43 journals and transactions, eight magazines, and 32 newsletters as of year-end FY15. During the year, ACM added 448 volumes of conference and related workshop proceedings to its portfolio. The ACM International Conference Proceedings Series (ICPS) added 113 new volumes, a significant increase over FY14.

The Publications Board continues to examine ways to offer authors and readers greater open access to published material. In the past year, in addition to continuing current policies offering authors a choice of copyright options and a suite of author-retained rights, the Board also approved in concept the creation of an umbrella Gold OA ACM journal.

ACM Books, a series of research monographs and graduate-level textbooks generated in partnership with Morgan & Claypool publishers, finished its first full year of operation. Six books are currently available in the ACM DL; 14 additional titles are under development.

ACM-W Connections, now a monthly online newsletter, covers ACM-W programs, celebrations, scholarships, awards, and chapters as well as contributed articles and upcoming events.

SIGAI launched a new quarterly newsletter "AI Matters," featuring articles of general interest to the artificial intelligence community. SIGMOBILE introduced *GetMobile*, a complete redesign of its former newsletter. This new venue serves as a vehicle for mobile researchers to keep abreast of developments in their community.

Education

ACM leads the computer science education community through the work of the ACM Education Board, the ACM Education Council, ACM SIGCSE, Computer Science Teachers Association (CSTA), and ACM Education Policy Committee.

The final report for CS2013, a joint effort of ACM and IEEE-CS, was presented to the ACM Council in October 2014. The report was commended for its comprehensive guidance for undergraduate CS programs. In fact, a Chinese translation of the report was recently completed.

ACM Competency Model of Core Learning Outcomes and Assessment for Associate-Degree Curriculum in Information Technology, a three-year effort, was published this year.

The second Heidelberg Laureate Forum brought together 200 young researchers, 200 mathematics and computer science laureates, including recipients of the ACM A.M. Turing Award, the Abel Prize, the Fields Medal, and the Nevanlinna Prize. The week-long event offered students a unique opportunity to share ideas with legends in the fields of computer science and mathematics.

The ACM Education Board's second Learning@Scale conference was held in Vancouver, B.C. last March, attracting over 185 attendees. Inspired by the emergence of massive online courses (MOOCs), ACM created this conference for the review and presentation of research on how learning and teaching can change and improve when done at scale.

The ACM Education Policy Committee (EPC) engages educators, industry, policymakers, and the public on public policy issues in computer science and computing-related education. ACM, through EPC, participated in the leadership council of STEM Education Coalition. This year the coalition launched a new policy forum aimed at informing stakeholders about policy issues. The forum focused on career and technical education, and common standards to preparing students for college and careers.

CSTA released a landmark report this year: *Sowing the Seeds: A Landscape Study on Assessment in Secondary Computer Science Education*. The study finds there is a dearth of valid and reliable assessments for measuring student learning in high school computer science. The report highlights the challenges faced by CS teachers in the U.S. and concludes with recommendations for advancing the state of assessment in K–12 CS education.

ACM's partnering with Code.org represents a significant investment in seeing real computer science exist and *count* in U.S. high schools. At the end of FY15, 27 states had moved to enact legislation to make computer science count as a core graduation credit in science and math. Furthermore, over 100 million students, parents, and teachers in 180+ countries participated in the Hour of Code during Computer Science Education Week.

ACM's second annual report on computing education trends confirmed a continuing growth in enrollment and degree production at participating not-for-profit U.S. academic institutions that grant bachelor's and/or master's degrees in the major computing disciplines. The ACM NDC Study is based on a survey of nearly 1,000 non-doctoral-granting academic departments and institutions in computing. The study also offered valuable pipeline data to businesses and industries that are competing in the job market for workers with skills in these areas.

In conjunction with the Association for Information Systems, SIGMIS has been developing a model curriculum for education in information systems both at the undergraduate and graduate levels. The Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems 2016 was released in June.

Professional Development

The Practitioners Board and Professional Development Committee (PDC) directed many new products and initiatives designed for computing professionals and managers.

In FY15, ACM PDC continued the successful webinar series with consistent monthly webcasts, tighter integration with SIGs, and a more diverse set of topics. The committee added 10 webinars this year. Among the featured topics: Software engineering case studies, agile pros/cons, fast data, mobile development, and the future of invention in CS.

For the fourth consecutive year, *ACM Queue*, the online practitioner's magazine spirited by the Practitioner Board, surpassed the million-pageview threshold, with 1,254 pages viewed over the 12-month period. *Queue* readership has double in the last five years.

SIGCOM and SIGMOBILE are planning a new initiative to improve interactions with industry within their discipline. SIGCOMM/SIGMOBILE Industry Day on Wireless Networking is a day-long event with industry leaders and academics from wireless and mobile networking designed to increase opportunities for collaboration between the two fields.

The Practitioners Board and Professional Development Committee directed many new products and initiatives designed for computing professionals and managers.

Public Policy

ACM's U.S. Public Policy Council (USACM) made significant progress this year in delivering on its mission to educate and inform policy leaders, ACM members, the computing community, and the public about U.S. policy issues related to IT and computing.

USACM also participated in several multi-stakeholder partnerships to help educate and raise awareness of technology issues among legislators and the public at large. USACM participated in standing committees evaluating policy issues related to accessibility, digital governance, intellectual property, law, privacy, and security.

ACM Europe Council and USACM presented a consensus position on computing policy issues to negotiators of a new EU-U.S. free-trade agreement. The proposed Transatlantic Trade and Investment Partnership agreement addresses trade issues between the EU and the U.S.

The Committee on Computers and Public Policy continues to assist ACM in a variety of internationally relevant issues pertaining to computers and public policy. CCPP's respected *ACM Forum on Risks to the Public in Computers and Related Systems*, designed to discuss potential and serious computer-related risks with a global audience, covers such issues as human safety, privacy, election integrity, and societal/legal responsibilities. Over this fiscal year, 68 issues of the *Risks* digest were produced.

The ACM Policy Award was established in FY15 to honor the contributions of an individual, or a small group, who has had a significant impact on the formation or execution of public policy related to computing. A prize of $10,000 will accompany this biennial award.

Balance Sheet: June 30, 2015 (in Thousands)

ASSETS	
Cash and cash equivalents	$43,577
Investments	80,833
Accounts receivable and other current assets	5,752
Deferred conference expenses and other assets	7,443
Fixed assets, net of accumulated depreciation and amortization	1,031
Total Assets	**$138,636**
LIABILITIES AND NET ASSETS	
Liabilities:	
Accounts payable, accrued expenses, and other liabilities	$11,396
Unearned conference, membership, and subscription revenue	24,983
Total liabilities	**$36,379**
Net assets:	
Unrestricted	94,596
Temporarily restricted	7,661
Total net assets	**102,257**
Total liabilities and net assets	**$138,636**

Optional Contributions Fund — Program Expense ($000)	
Education Board accreditation	$95
USACM Committee	10
Total expenses	**$105**

Students

The 39th Annual ACM International Collegiate Programming Contest (ACM-ICPC) was hosted by Mohammed V University, Al Akhawayn University, and Mundiapolis University with 128 teams competing in the World Finals. Earlier rounds of the competition included more than 38,000 contestants representing 2,500 universities from 101 countries. Financial and systems support for ICPC are provided by IBM. The top four teams won Gold medals as well as employment or internship offers from IBM.

The ACM Student Research Competition (SRC), sponsored by Microsoft Research, continues to offer a unique forum for undergraduate and graduate students to present their original research at well-known ACM-sponsored and co-sponsored conferences before a panel of judges and attendees. This year's SRC saw graduate and undergraduate winners compete against more than 299 participants in contests held at 22 ACM conferences.

The ACM-W Scholarship program further enhanced its support for women undergraduate and graduate students in CS and related programs. The committee awarded over 40 student scholarships in FY15 to students to attend research conferences around the world.

Internationalization

ACM Europe focused on mapping out strategies for growing ACM membership and the computing profession throughout Europe. Its plans include hosting more conferences and local chapters, developing its roster of Distinguished Speakers, and increasing the visibility of ACM Europe at the European Commission level for policy issues like education.

ACM Europe Council also held a second open panel discussion to address the need to improve CS education in Europe. The panel discussed aspects about the current education situation in Europe and especially in Greece, where the economic crisis has had an adverse impact. The Committee on European Computing Education (CECE) of ACM Europe is addressing these challenges by working on charting current situations in European countries, and examining the systems that de-

Statement of Activities: Year ended June 30, 2015 (in Thousands)

REVENUE	**Unrestricted**	**Temporarily Restricted**	**Total**
Membership dues	$7,980		$7,980
Publications	20,860		20,860
Conferences and other meetings	27,313		27,313
Interests and dividends	1,888		1,888
Net appreciation of investments	569		569
Contributions and grants	5,466	$2,575	8,041
Other revenue	128		128
Net assets released from restrictions	1,421	(1,421)	0
Total Revenue	**65,625**	**1,154**	**66,779**
EXPENSES			
Program:			
Membership processing and services	$784		$784
Publications	10,842		10,842
Conferences and other meetings	24,792		24,792
Program support and other	10,256		10,256
Total	**46,674**		**46,674**
Supporting services:			
General administration	10,863		10,863
Marketing	1,343		1,343
Total	**12,206**		**12,206**
Total expenses	**58,880**		**58,880**
Increase (decrease) in net assets	6,745	1,154	7,899
Net assets at the beginning of the year	87,851	6,507	94,358
Net assets at the end of the year	**$94,596**	**$7,661**	**$102,257***

* Includes SIG Fund balance of $43,435K

velop curricula and teacher training. The committee also hopes to develop a new conference dedicated to computing education in Europe.

ACM-W Europe is driven by a collective vision to cultivate and inspire people about the opportunities in computer science and clear the pathways throughout Europe for women in computing. Much of the efforts this year were extended to raise the visibility of computer science among European women, utilize social media channels for outreach, and to increase the number of ACM-W chapters in Europe.

The ACM China Council is committed to helping advance computing as a science and profession in China. Along with growing membership and chapters throughout the region, ACM China participates in reviewing the translation of select articles from *Communications of the ACM*, which are uploaded to the DL and distributed monthly to Chinese members as the *CACM China Edition*.

ACM India energized its efforts to promote education and the status of women in technology in India. Moreover, India-based conferences continue to grow into more dynamic and far-reaching events. ACM India launched an Eminent Speakers Program this year, which promises to gain visibility for ACM in India as well as putting

2014 ACM Award Recipients

A.M. TURING AWARD
Michael Stonebraker

ACM-INFOSYS FOUNDATION AWARD IN THE COMPUTING SCIENCES
Dan Boneh

ACM-AAAI ALLEN NEWELL AWARD
Jon Kleinberg

GRACE MURRAY HOPPER AWARD
Sylvia Ratnasamy

KARL V. KARLSTROM OUTSTANDING EDUCATOR AWARD
William A. Wulf

EUGENE L. LAWLER AWARD FOR HUMANITARIAN CONTRIBUTIONS WITHIN COMPUTER SCIENCE AND INFORMATICS
Robin Roberson Murphy

OUTSTANDING CONTRIBUTION TO ACM AWARD
Dame Wendy Hall

DISTINGUISHED SERVICE AWARD
Jeannette Wing

PARIS KANELLAKIS THEORY AND PRACTICE AWARD
James Demmel

SOFTWARE SYSTEM AWARD
Mach
Rick Rashid
Avadis Tevanian

ACM-W ATHENA LECTURER AWARD
Jennifer Widom

2015 ACM-IEEE CS ECKERT-MAUCHLY AWARD
Norman Jouppi

2015 ACM-IEEE CS KEN KENNEDY AWARD
Charles E. Leiserson

ACM PRESIDENTIAL AWARD
John R. White, ACM CEO

DOCTORAL DISSERTATION AWARD
Matei Zaharia, MIT

HONORABLE MENTION
John Criswell, University of Rochester
John Duchi, Stanford University

ACM INDIA DOCTORAL DISSERTATION AWARD
Rujurekha Sen

chapter members in direct contact with thought leaders in a variety of computing sub-disciplines.

The ACM India Education Committee conducted a two-day workshop on computing curricula. The main objective of the workshop was to contrast the success seen by the Indian IT industry with the lack of similar progress in computing education in India, and to bring into focus what can be done to advance the future of computing and to meet the needs of employers.

An ever-growing number of ACM SIGs hosted conferences or established chapters worldwide in FY15. SIGAPP's Symposium on Applied Computing was held in Spain, SIGHPC chartered a new chapter in Central Africa, SIGIR's annual conference was held in Gold Coast, Australia; a SIGSOFT chapter opened in India, and SIGWEB's Hypertext 2014 was held in Santiago, Chile. SIGCHI continues to advance internationalization, with plans to focus on Asia.

Electronic Community

ACM's e-Rights transfer application system completed its second full year of operations. This comprehensive system gives authors new options for managing rights and permissions. The system, used by all ACM journals, proceedings, and magazines, completely automates the rights transfer process.

More and more readers are accessing ACM's magazines via mobile devices. *Communications of the ACM*, *ACM Inroads*, *XRDS*, and *Interactions* are accessible as easy-to-use mobile apps for iPhones, iPads, and Android devices. These downloadable apps enable members to access ACM magazines in a new way.

A major enhancement to the ACM DL debuted this year with the rollout of new journal homepages, which now present a uniform, seamless style of ACM branding. The process will continue through 2016.

ASSETS 2014 was host to a successful experiment using Beam telepresence robots to enable attendance by two remote participants with disabilities, who would not have been able to attend otherwise. Via these robots, they participated in conference sessions, asked questions, and socialized during breaks.

ACM SIGs across the board continue to strengthen their online presence to build global awareness as well as incorporate social media into their operation at every opportunity. SIG-UCCS offered six professional development webinars to members this year; recordings of all are available on their website. SIGSIM debuted its Digest Picture Blog, which contains modeling and simulation-related news and pictures. SIGMOBILE generated ePub files for all accepted papers from all its 2014 conferences.

Conferences

ACM hosted its first conference by and for practitioners—Applicative—in February 2015. The response to the conference was overwhelming positive; indeed, conference organizers for the 2016 Applicative conference expect attendance to grow twofold.

ACM's Committee of Professional Ethics ran a workshop on teaching computer ethics and decision making at the first IEEE Ethics Conference. The event explored professional/ethical issues and the use of ACM's Ethics Codes in computing classes.

SIGGRAPH 2014 welcomed over 14,000 artists, research scientists, gaming experts and developers, filmmakers, students, and academics from 75 countries to Vancouver, B.C. Over 1,500 speakers and contributors participated in the event and SIGGRAPH's exhibition hall drew 175 industry organizations from 18 countries. In addition, SIGGRAPH Asia, in Shenzhen, China, attracted over 6,000 participants from 53 countries.

ACM-W Celebrations continue to showcase female role models, encourage mentoring and networking, supply accurate information about computing careers, and create opportunities for women to present their research, often for the first time in their careers. These small conferences run with almost 100% volunteer effort and involve considerable fundraising.

The 2014 Grace Hopper Celebration for Women in Computing India hosted a sold-out group of 1,600 attendees. The program featured many events to support women in computing in India, including sessions on entrepreneurship, building confidence, and professional development, as well as technical presentations on topics ranging from wearable technology to machine learning.

Networking Networking Women (N^2 Women), a SIGMOBILE program, fosters connections among the underrepresented women in computer network and related research fields who share the same career hurdles and attend the same conferences.

Recognition

There were 164 new chapters chartered in FY15. Of the 27 new professional chapters, five were U.S.-based and 22 were based outside the U.S.; of the 137 new student chapters, 62 were established in the U.S. and 75 outside the U.S.

The ACM Fellows Program recognized 47 members for their contributions to computing and computer science in FY15.

ACM also named 49 new Distinguished Members in FY15, of which there were five Distinguished Educators, two Distinguished Engineers, and 42 Distinguished Scientists. This year's Distinguished Members are from universities, leading international corporations, and research institutions in Austria, Germany, Switzerland, the Netherlands, Sweden, Japan, India, the U.K. and North America.

ACM and the Computer Science Teachers Association (CSTA) announced a new award this year: the ACM/CSTA Cutler-Bell Prize in High School Computing recognizing talented high school students in computer science. The program seeks to promote and encourage the field of computer science, as well as to empower young and aspiring learners to pursue computing challenges outside of the traditional classroom environment. Four winners will be selected annually and each will be awarded a $10,000 prize and cost of travel to a reception where students will demonstrate their programs and discuss their work.

The ACM History Committee fosters the collection, preservation, and interpretation of the history of ACM and its role in the development of computing. The committee continued work on its three-year project to create oral histories for all non-interviewed Turing Award recipients, and well as commissioned histories of ACM's officers, editors, among others.

Inviting Young Scientists

HEIDELBERG LAUREATE FORUM

Association for Computing Machinery

Meet Great Minds in Computer Science and Mathematics

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

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

The Heidelberg Laureate Forum is **September 18–23, 2016** in Heidelberg, Germany.

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

Who can participate?

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

How to apply:

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

What is the schedule?

Application deadline—**February 3, 2016**.

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

Successful applicants will be notified by **end of March/early April 2016**.

More information available on Heidelberg social media

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

BLOG@CACM

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

twitter

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

DOI:10.1145/2843530 http://cacm.acm.org/blogs/blog-cacm

Controlling Cyber Arms, and Creating New LEGOs

John Arquilla identifies flaws in a potential U.S.-China cyber arms control pact, while Joel C. Adams suggests an unusual way of preserving computer science history.

John Arquilla
"A Farewell to (Virtual) Arms?"
http://bit.ly/1RkiAfA
October 2, 2015

Much attention has been focused recently on the budding possibility of a Sino-American cyber arms control agreement, whose foundation would be a mutual pledge of "no first use" of bits and bytes to cripple critical civilian infrastructure. It is an intriguing development, despite having three troubling flaws.

The first problem afflicts the agreement's logical basis, given that both sides pledge not to mount such attacks "in peacetime." But what if such an attack, a "digital Pearl Harbor," were to be the opening act of war—when "peacetime" would have been thereby ended? A bit of a conundrum, complicated further by the fact that most advanced militaries rely, to varying degrees, on civil infrastructures they do not own or control for much of their communications, logistics, and other functions. So, in a sense, civil infrastructure can actually be viewed as consisting of a range of strategic, military-related targets.

Next there is the major perceptual problem that lies at what might be called the "boundary layer" of this agreement that does not explicitly extend to cyber espionage. The difficulty here is that the sorts of actions, exploits, and intrusions that go with virtual spying are observationally equivalent to the preparatory access to the adversary's systems that would be sought prior to launching an actual attack. Thus the cyber peace would always be poised on a knife-edge of instability. A related perceptual complication is that the ultimate identity of the attacker is not always clearly or easily distinguished—and so the potential for a third party, C, to attack A anonymously, or to finger innocent B as the culprit, is a very real risk, one that might lead to escalation to war in the physical world—which was the scenario I unfolded in my short story in *Wired* back in 1998, "The Great Cyberwar of 2002" (http://bit.ly/1XMUSfy).

The third difficulty with the Sino-American cyber arms control initiative lies in its scope. The initially narrow focus on infrastructure protection does little or nothing to deal with the large-scale theft of intellectual property that constitutes what can be called the realm of "strategic crime." U.S. President Barack Obama has said much about this over the past few years, and has explicitly called out China as a culprit. In a public statement growing out of a meeting between him and Chinese President Xi Jinping, both leaders affirmed neither country would *knowingly* engage in intellectual property theft.

When asked during his recent testimony before the Senate Armed Services Committee whether there was any real chance of curtailing intellectual property theft, the director of National Intelligence, former general James Clapper, gave a one-word answer: "No." He went on to make critical comments about the possibility of cyber arms control, indicating instead his preference for a focus on improving defenses. His only nod to any sort of agreement was an allusion to Ronald Reagan's approach to engaging in arms-reduction talks with the Russians back in the 1980s: "Trust but verify." So it seems, even in American officialdom, the window of opportunity for cyber arms control has only been opened a crack.

Yet it may prove enough of an opening to move ahead, for the "no first use" doctrine has caught on in the nuclear realm—though it took many decades for the U.S. to decide to move in this direction (there are still some extreme conditions noted in the American nuclear posture statement that would allow first use, but for all practical purposes this is no longer a usable first option).

Issues of verification aside, nations—not just China and the U.S., but others, too—have incentives to behave circumspectly about starting a strategic cyberwar that would incur huge economic costs and run the risk of a virtual conflict escalating into a shooting war in the physical world. Full disclosure: I introduced the idea of a cyber no-first-use doctrine in an article in the journal *Ethics and Information Technology* back in 1999 ("Can Information Warfare Ever Be Just?" http://bit.ly/1kpQRPq), so I am hardly impartial. It has been a long wait to hear leading heads of state talking about such a possibility, and we must allow the discourse to unfold, rather than simply to dismiss it as idealistic or quixotic.

The best way to envision cyber arms control may be to think of it as analogous to other controlled activities in areas in which diffusion of the enabling technology itself is unstoppable. In the varied realms of chemical and biological weapons, for example, countless nations have access to the materials required to craft such weapons. And yet there are behavior-based arms control agreements in force, to which nearly all countries subscribe, that forbid their use. In the main, there is strong compliance with few violations. Such compliance may well be possible in the cyber arena, too. It is an approach well worth exploring.

With regard to the logical possibility that a "peacetime" pledge is not violated if a strategic cyber attack *starts* a war, the response to this concern is that such an attack could still be limited to military-related targets. To return to the nuclear analogy, this would be very much like the "counterforce" strategic doctrine of the Cold War era that sought to target missiles and other military targets, not population centers. In this way, it was thought, a nuclear war could be waged without massive civilian deaths.

Only a small portion of critical infrastructure is essential for military operations, so cyber combatants would have good chances of operating against armed forces without imposing too much civilian suffering. To be sure, a conflict of this sort would inflict much costly, disruptive collateral damage, but far less than would be the case in a city-busting, apocalyptic general nuclear war. Thank God counterforce nuclear doctrine was never put to use. But cyberwar is much more thinkable than an atomic Armageddon, so the counterforce doctrine that never had to be used for its original purpose may well be dusted off when thinking about how to conduct conflict in the virtual domain.

The most nettlesome problem, of course, is the veil of anonymity in which cyber aggressors—nations or networks—may be inclined to enshroud themselves. Clearly, forensics must continue to improve so as to identify attackers accurately. And just as clearly, a great deal of work is needed to bring forensics up to the needed level of accuracy. Also, strategic deception about the identity of the perpetrator, as mentioned earlier, must be guarded against. But these challenges are no reason to give up on the promise of cyber arms control.

On balance, the emerging, maturing discourse about applying notions of arms control to the cyber realm is a "net positive" (no pun). There are indeed obstacles to overcome, but the potential gains for peace and cybersecurity make the efforts to master these challenges more than worth the while.

Joel C. Adams
"A Lovelace, Babbage, and Analytical Engine LEGO Set?"

http://bit.ly/1JvpkC6
August 29, 2015

LEGO has a crowdsourcing ideas site, at https://ideas.lego.com/, where LEGO fans can pitch ideas for new LEGO sets. What a great way to let your audience help you conduct market research!

Hugh McGuire was kind enough to send me a note about a Lovelace and Babbage set (https://ideas.lego.com/projects/102740) that Stewart Lamb Cromar has proposed. The set would include LEGO figurines for Ada Lovelace and Charles Babbage, LEGO pieces to build a representation of the Analytical Engine, punch cards, and related pieces. The various pieces would be styled with "a steampunk aesthetic" to capture the imaginations of young builders. The set would thus let young LEGO builders realize Babbage's vision by completing his Analytical Engine, and learn about the historical roles played by Babbage and Lovelace.

(For those who have forgotten their early computing history: back in 1837, Charles Babbage designed a general-purpose (that is, programmable with punch cards) mechanical computer he called the Analytical Engine. Although a working Analytical Engine was never built, Ada [the Countess of] Lovelace understood the design's potential and corresponded with Babbage about it. She developed a detailed algorithm for using the Analytical Engine to compute Bernoulli numbers, for which she has been dubbed the first computer programmer. In honor of her contributions, the Ada programming language was named after her. Those interested in more details should read "Lovelace and Babbage and the Creation of the 1843 'Notes'" (http://inroads.acm.org/article.cfm?aid=2810201) by Fuegi and Francis.)

Many stories from the "steampunk" genre take place in alternative universes where Babbage actually built an Analytical Engine powered by steam and Ada wrote programs for it. Such stories generally explore the question, "What if … the power of computing was unleashed in the Victorian era?"

Back in our universe, the dimensions of the LEGO Analytical Engine would be sufficient to accommodate a Raspberry Pi 2 (https://www.raspberrypi.org/products/raspberry-pi-2-model-b/), if one wishes to put a computer inside. That would be fun to see: a LEGO Analytical Engine driving an LCD display, mouse, and keyboard!

One of the motivations for the set is to commemorate the 200th anniversary of Ada's birth (Dec. 10, 1815). The set would thus teach young LEGO builders some early computing history, and that women have been involved in computing since its origins. It would thus help to counter the popular misperception that only men belong in computer science.

If an idea on the LEGO site receives 10,000 supporting votes, they will consider building the set. To support a project, you must register on their site, but registration only takes a minute, so if you want to raise awareness of computer science in our society, and help young boys and girls realize computer science is not limited to males, I encourage you to support this proposal by clicking the blue button on the proposal page (https://ideas.lego.com/projects/102740).

John Arquilla is a professor at the U.S. Naval Postgraduate School. **Joel C. Adams** is a professor at Calvin College.

news

Science | DOI:10.1145/2843532

Neil Savage

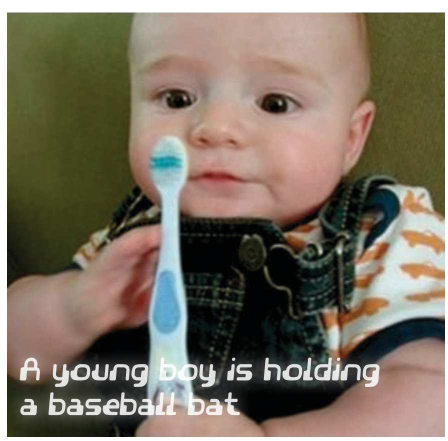
Seeing More Clearly

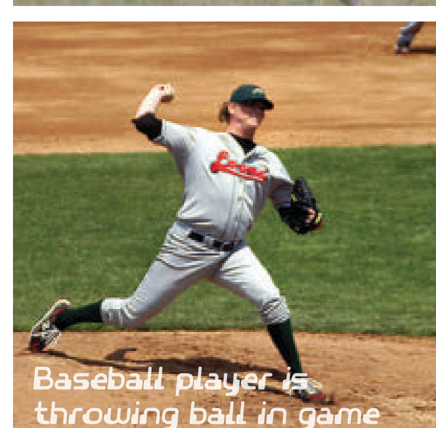
Computer understanding of images has improved rapidly, but true visual intelligence is still a long way off.

THE PHOTO SHOWS a baby, dressed in a striped shirt and denim overalls, staring intently at the toothbrush he is grasping in his left hand while he pokes at it with his right. The caption underneath reads, "A young boy is holding a baseball bat."

The computer at Stanford University in California that generated that description failed in this case, but in others its captions were much more accurate, if not terribly exciting. The phrases "black and white dog jumps over bar," "little girl is eating piece of cake," and "baseball player is throwing ball in game" are all correct characterizations of the photos in question. Even if it occasionally mistakes a toothbrush for a baseball bat, or a ferret for a cat, the computer today can do a much better job of explaining what is happening in a photograph than it could even a couple of years ago. The field of computer understanding of images has made remarkable progress in the last few years.

"Things are moving really fast right now, because we are dealing with the earliest successes of object recognition and tasks," says Fei-Fei Li, director of the Vision Lab and the Artificial Intelligence Lab at Stanford, who developed the program that created those captions. Part of what has made such successes possible is the development





IMAGES COURTESY OF STANFORD.EDU

of large sets of training data. Li, for instance, began in 2007 to build ImageNet, a collection that has grown to nearly a billion images hand-labeled by almost 50,000 people through Amazon Mechanical Turk, which provides small payments to people willing to perform such tasks. Microsoft, meanwhile, developed the Common Objects in Context (COCO) dataset, which contains more than 300,000 images, each labeled with five captions.

The other advance is that computers have become powerful enough to apply convolutional neural networks to the task of understanding images. Each neuron in the network is a filter that looks at a small image segment a few pixels wide and computes a value expressing how confident the computer is that a given object is within that segment. The segments overlap to cover the entire image, and the network repeats this process through many layers, with each layer's output providing input for the next layer. "With careful training, these things actually work very well," says Rob Fergus, a scientist in Facebook's AI Research Group in Menlo Park, CA. "Prior to these models, these recognitions systems didn't really work in practice."

Five years ago, Fergus says, recognition systems had an error rate of approximately 26%; today, that is down to 5%–6%. "It's not to say they can do visual recognition as well as a human, but in this slightly artificial setting, they do pretty well," he says. Eventually, computers should surpass humans. Thanks to their vast access to information, they should be able to always identify a breed of dog or the make and model year of a car, something most humans cannot manage.

Facebook—whose users upload 400 million images each day, according to Fergus—is very interested in automating image understanding. That social network wants to know whether to put a given photo into a user's newsfeed, based on what it shows or whether it depicts friends of the user; it also wants to detect and automatically delete objectionable content before people see it. One area Facebook is exploring is how to identify people whose faces are not visible, or partly obscured. While a human can look at a picture of President Obama with his face turned away from the camera, or Mahatma Gandhi with his head bowed in prayer, and instantly recognize them, a computer generally cannot.

Facebook researchers have developed a program called Pose Invariant Person Recognition (PIPER) that looks for "poselets"—a hand next to a hip, how a head and shoulders look from the back—that will tell it that it is seeing a person, even when it cannot identify a face. Given a starting image of an individual, the system was able to identify the same person in other photos 83% of the time. When a frontal image of a face was available, it improved the accuracy of Facebook's facial recognition software to 89%–94%.

"We aren't really at the point of [a computer] understanding what's going on in that image. At a glance, a person can figure that out."

Identifying discrete objects, however, is only a small part of understanding a scene. The next step toward visual intelligence is recognizing the relationship between those objects and noticing if action is taking place, and that is where attempts to write captions come in. Researchers at Microsoft trained their system by presenting the computer with images that were accompanied by human-generated captions. One advantage of that approach is that the humans who write the simple descriptions will tend to focus on the most important details. If the caption contains the word "horse," for instance, it is more likely that a horse will feature prominently in the picture, rather than be a small part of the background. The computer might also find a high correlation between the use of the word "riding" and images of a person on top of a horse.

Once trained, the computer follows a three-step process, says Xiadong He, a researcher in Microsoft's Deep Learning Technology Center in Richmond, WA. First, it identifies objects within an image to create a list of words most likely to apply to the image. Then it uses those words to construct up to 500 candidate sentences. Finally, it ranks the sentences in order of likelihood, coming up with one that is most likely to describe what is in the picture. To do that, it creates a map of probabilities, called semantic vectors, ranking various words and phrases to decide which are more likely to go together. It also creates a similar ranking for words it has assigned to portions of the image. It compares the semantic values of the text to those of the image, and declares that those from each group that are closest to each other are most likely correct. "The overall picture should have the same semantic value as the description," says He.

In the MS COCO Image Captioning Challenge 2015, in which 15 groups from industry and academia competed to see who could do the best job of getting computers to generate descriptions, the Microsoft team ranked first according to a Turing-style test, in which judges were asked to determine whether the captions had been created by a human or a machine. Slightly more than 32% of the machine-generated captions were thought by the judges to have been written by people; only 68% of human-generated captions were attributed to humans by the judges, so the software is almost halfway toward passing the Turing test, He says. "If you looked at this problem two years ago, the outcome was almost garbage. It was so simple to tell which captions came from a human and which came from a computer," says He.

In another category based on the percentage of captions thought to be equal to or better than captions written by people, researchers from Google came out ahead; combining the scores left the two companies finishing in a tie.

Microsoft researchers taking a slightly different tack tied for third place with researchers from the University of Toronto, Canada, whose software also analyzes images and sentences to find the best match. The UT approach, developed with researchers from the University of Montreal, Canada, includes the concept of attention. The computer identifies the most important object in a given region of an image, then moves sequentially through the regions, with its decisions about what is most important in each influenced by what it saw in the others. Like Microsoft's system, it analyzes both the image and the sentence and tries to put together the two that best match.

While the program can correctly identify, say, that there are people in a boat in a particular photo, that is about as far as it goes, says Richard Zemel, a computer scientist at the University of Toronto. "What are they actually doing? Are they rowing the boat? Are they falling out of the boat?" he says. "We aren't really at the point of understanding what's going on in that image. At a glance, a person can figure that out."

To move closer to such understanding, Zemel is working on training the computer to answer arbitrary questions about an image, such as "what color is the shirt?" or "what's in front of the sofa?" Answering such questions requires a more detailed description of an image than simply what is in it and where. "If you really understood the image, you could answer a question about it," Zemel says. "I think that's more indicative of true understanding." The work is still in its early stages, in part because the existing database of human-generated questions and answers about images the computer can learn from is not very large.

Li says true visual intelligence could be important in a lot of areas. Self-driving cars, for instance, have to be able to do more than simply drive down a road in a particular direction. If a computer-driven car comes across a construction zone, for instance, the new information it needs to proceed safely will be mostly visual, from such sources as officers directing traffic. "You have to read their gestures. You have to read detour signs. You have to see the orange cones," she says.

Electronic eyes with visual intelligence in a hospital could alert nurses to problems with a patient that the nurses missed, or remind them to wash their hands before they touch an IV. In an airport, they might identify an unattended backpack as a security issue.

Though recent progress has been rapid, Li believes it could slow as cientists try to tackle more complicated challenges than object recognition and move into reasoning about relationships among objects and actions. That will require building datasets that reflect a much more complex level of interconnectedness in the world. She compares it to the challenge of basing computer searches on natural language. Google does an excellent job when given discrete search terms, but ask it a long question, such as "give me the names of painkillers that do not have stomach side effects," and it stumbles.

Right now, computers are roughly as good at describing the content of images as a three-year-old human, Li says. "The complete level, on par with an adult, college-degreed human, I think is going to be a long way off." ■

Further Reading

Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Plattz J.C., Zitnick, C.L., Zweig, G.
From Captions to Visual Concepts and Back, *Computer Vision and Pattern Recognition*, Boston, MA, 2015.

Zhang, N., Paluri, M., Taigman, Y., Fergus, R., and Bourdev, L.
Beyond Frontal Faces: Improving Person Recognition Using Multiple Cues, *Computer Vision and Pattern Recognition*, Boston, MA, 2015.

Ren, M, Kiros, R., and Zemel, R.
Exploring Models and Data for Image Question Answering, *Deep Learning Workshop at the International Conference on Machine Learning*, Lille, France, 2015.

Vinyals, O., Toshev, A., Bengio, S., and Erhan D.
Show and Tell: A Neural Image Caption Generator, *Computer Vision and Pattern Recognition*, Boston, MA, 2015.

How We Teach Computers to Understand Pictures, Fei-Fei Li TED Talk. https://www.youtube.com/watch?v=40riCqvRoMs

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

Milestones

American Academy Adds CS Fellows

Among the 181 Fellows and 16 Foreign Honorary Members recently elected to the American Academy of Arts and Sciences were seven computer scientists and a mathematician who is also a computer scientist.

Honorees in the Computer Sciences category were:

- **Sanjeev Arora, Charles C. Fitzmorris Professor of Computer Science at Princeton University.**
- **Susan T. Dumais, a Distinguished Scientist at Microsoft and affiliate professor at the University of Washington Information School.**
- **Laura M. Haas, IBM Fellow and director of the IBM Accelerated Discovery Lab at the IBM Almaden Research Center.**
- **Joseph Y. Halpern, a professor of computer science at Cornell University and administrator for the Computing Research Repository of arXIV.org.**
- **Maurice P. Herlihy, a computer scientist at Brown University.**
- **Ravindran Kannan, Principal Researcher at Microsoft Research Labs, Bangalore, India, and first adjunct faculty of the Computer Science and Automation Department of the India Institute of Science.**
- **Nicholas W. McKeown, a professor in the electrical engineering and computer science departments of Stanford University.**

In addition, 2007 ACM A.M. Turing Award laureate Joseph Sifakis, a computer scientist who works for CNRS at the VERIMAG laboratory, and as coordinator of Artist2, the European Network of Excellence for research on embedded Systems, was named Foreign Honorary Member in the Computer Sciences category.

New Fellows in the Mathematics, Applied Mathematics, and Statistics category included László Babai, a professor in the departments of Computer Science and Mathematics of the University of Chicago.

GUGGENHEIM FOUNDATION NAMES COMPUTER SCIENTISTS AS FELLOWS

The John Simon Guggenheim Memorial Foundation recently awarded Fellowships to 175 scholars, artists, and scientists in the U.S. and Canada on the basis of prior achievement and exceptional promise. Among these were two computer scientists, Vincent Conitzer and Krishna V. Palem.

The research of Conitzer, who is Sally Dalton Robinson Professor of Computer Science and professor of economics at Duke University, focuses on computational aspects of microeconomics, in particular game theory, mechanism design, voting/social choice, and auctions, using techniques from, and including applications to, artificial intelligence and multi-agent systems.

Palem is the Ken and Audrey Kennedy Professor of Computing at Rice University, where he is director and founder of the Nanyang Technological University-Rice Institute of Sustainable and Applied Infodynamics, with appointments in computer science and electrical and computer engineering. Palem is a leader in Embedded Systems research, having founded one of the earliest laboratories for research in academia dedicated to this field in 1994.

Technology | DOI:10.1145/2843555

Samuel Greengard

Better Memory

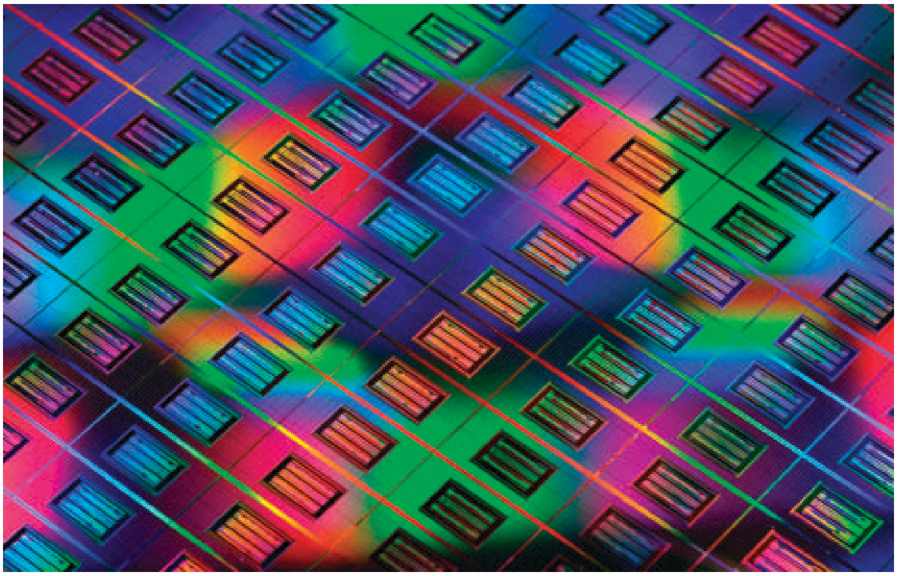
Advances in non-volatile memory are changing the face of computing and ushering in a new era of efficiencies.

SINCE THE DAWN of computing, an ongoing challenge has been to build devices that balance the need for speed and persistent storage. While dynamic random-access memory (DRAM) is fast, it can only hold data as long as it receives an electrical current; when the computing device is switched off, the data disappears. And although storage devices such as hard drives are efficient for holding large volumes of data, they are relatively slow. The result? "A performance or persistence choice that doesn't give you the best of both worlds," states David Andersen, an associate professor in the computer science department at Carnegie Mellon University.

Over the last few years, engineers have resolved some of these challenges through solid state drives (SSDs) that contain no disk or other moving parts, yet continue to store data when the devices are switched off. What is more, SSDs use less power and provide higher reliability than hard disk drives. However, they are far from ideal. For one thing, they're still relatively expensive. For another, while SSD is often an improvement over older technologies and sometimes reduces the need for DRAM, it still does not provide the level of speed, flexibility, and lifespan that users desire.

"There is a desire for more advanced technology, particularly in high-performance computing systems," says Jim Handy, memory analyst at market research firm Objective Analysis.

All of this is leading researchers down the path to faster and more advanced non-volatile random-access memory (NVRAM) technologies. These technologies—some of them radically different than today's flash storage technologies—could usher in speed and performance efficiencies that change computing. Unlike DRAM, these systems do not necessarily store the ones and zeros of binary code on a capacitor; they instead use memristors, which rely on electrical resistance. This produces efficiency gains, but also energy savings. These technologies could ultimately replace today's flash, SSD, static random-access memory (SRAM) and dynamic random-access memory (DRAM).



Hewlett-Packard Enterprise Memristor devices on a 300mm wafer.

IMAGE COURTESY OF HEWLETT-PACKARD LABS

They have names like 3D XPoint, MRAM, MeRAM, Memistor, NRAM, STT-RAM, PCM, CBRAM, RRAM, Millipede, and Racetrack. Much faster persistent memory is a potential game-changer for high-performance clusters and transaction-oriented systems because checkpoints must become persistent before an operation completes.

These technologies could usher in speed and performance efficiencies that change computing.

Says Andrew Wheeler, vice president and deputy director of HP Labs: "Today's computer architecture is fundamentally unchanged. It's the same architecture we've been using for 60 years—processors with a fixed amount of local memory, connected to storage and memory over an I/O bus. NVRAM becomes really interesting when you introduce the opportunity to simultaneously reinvent the architecture."

Flash Forward

The ability to design a more advanced memory architecture would have a profound impact on everything from high-performance computing clusters to smartphones and devices that comprise the Internet of Things. The technology could change basic computing architectures and storage designs, and address issues such as battery life, power requirements, in-memory database (IMDB) designs, and the way applications are coded. "Today, flash (memory) occupies the middle ground between speed and durability. It isn't as fast as DRAM and it isn't as durable as a disk drive. The goal is to close the gap further so that it's possible to address the challenges related to large databases and increasingly complex computing problems, as well as consumer devices," Andersen explains.

At HP, for example, researchers are working to develop memristor technology that uses electrons for processing, photons for communication, and ions for storage. "The Machine" creates a vast pool of fast NVRAM, connected to task-specific processors over a high-bandwidth, low-latency photonic fabric. The goal, Wheeler says, is to build a system that better optimizes logic gates while delivering long-term storage. HP refers to the approach as Memory-Driven Computing (MDC). "Every buffer copy or block move that we can design out saves energy, reduces the chances for interception or corruption, and shrinks the security attack surface," he says. The technology, which the company hopes to have commercially available by 2016, would tackle petascale datasets that are beyond reach today.

Memristor technology would consume a fraction of the power of today's memory systems. "At the tiny scale, having tens of terabytes of virtually zero-power memory allows us to build a new class of smart, secure IoT devices that can store their experience to know what's normal and what's novel or to satisfy a query from a neighboring peer or central intelligence," Wheeler explains. "Applications and operating systems that are fully adapted to pervasive non-volatile memory could enable perpetual computing where there is no more 'off switch'. When sufficient energy is present, information is processed; otherwise, the current state is preserved." HP hopes to have The Machine available in a range of form factors over the coming decade, based on price and performance.

HP is not the only player in the space. Intel and Micron are collaborating on a technology called 3D XPoint memory, which the companies claim is 1,000 times faster than the NAND flash storage used in current memory cards and in solid state drives. The dual in-line memory modules (DIMMs) are designed to be compatible with today's DDR4 SDRAM (double data rate fourth-generation synchronous dynamic random-access memory) but deliver a fourfold capacity increase. The proprietary solution offers performance gains without modifications to the underlying operating system or applications. However, the platform would require a redesigned central processing unit (CPU) and new extensions in order to take advantage of the 3D XPoint technology. Analysts say the technology would benefit organizations running large numbers of servers in a datacenter. 3D XPoint, for instance, would anticipate when data is required and transfer it in advance to the 3D XPoint component.

Other technologies are emerging as well. For instance, Crossbar has produced a working test chip for its RRAM (resistive random-access memory) technology. The company claims the system delivers 100 times lower read latency than NAND flash storage, along with 20 times faster writes without any block erase design constraints. It also delivers up to 1 terabyte of storage on a single chip, in an architecture that is 3D-stackable and scalable to sub-10 nanometers. Within the chip, each cell surrounds an insulating switching medium between electrode layers. When electrodes receive voltage nanoparticles in the switching medium, they form a conductive filament. The design supports stacking and it can be scaled down to fabrication nodes smaller than five nanometers.

Another technology, MeRAM (magnetoelectric random access memory), replaces the electrical current of spin-transfer torque (STT) with voltage to write data. This nanoscale approach results in 10 times to 1,000 times greater energy efficiency.

Memristor technology, using electronics for processing, photons for communication, and ions for storage, would consume a fraction of the power of today's systems.

"At this point, nobody knows which of the horses in the NVRAM game will win or how things will play out, but the bottom line is that the technology will very likely make a big impact on computing," Andersen says.

Making it All Compute

The practical benefits of next-generation NVRAM could be profound. Los Alamos National Laboratory began using NAND flash storage for high checkpointing and other high-performance computing in its Trinity system in September 2015. The National Energy Research Scientific Computing Center (NERSC) Cori system also will utilize the concept. Trinity holds nearly 2 petabytes of DRAM in main memory and 4 petabytes of NVRAM to support an I/O enhancement—essentially a new storage layer—know as a burst buffer. Gary Grider, division leader for the Los Alamos High Performance Computing Division, says more advanced versions of the technology will be incorporated into future CORAL (Collaboration of Oak Ridge, Argonne, and Livermore) supercomputers that will tap into the knowledge gained from the new tier of storage in Trinity.

Grider says NVRAM advances will have a major impact on supercomputers and also on consumer devices, including laptop computers, smartphones, and cameras. As prices drop and the technology advances, "It will become far more ubiquitous."

He also believes next-generation NVRAM could make today's data storage hierarchies obsolete. He points to the Intel-led U.S. Department of Energy Storage Fast Forward DAOS (Distributed Application Object Storage) Project, which targets HPC applications with scalable, transactional, versioned, and end-to-end reliable exploitation of multiple tiers of non-volatile storage. It will exploit non-volatile storage on compute node, in system burst buffer nodes, and on remotely attached parallel disks systems.

In addition to I/O use cases, next-generation NVRAM could be harnessed by some applications for out-of-core direct use in order to tap into a slower but larger memory pool.

Andersen believes NVRAM could help make future devices smaller and cheaper, as well as speeding start-up times for certain types of devices and sensors. "A suspend-resume mode has a lot of advantages for sensors and actuators that are part of the Internet of Things. The goal for these devices is to be insanely cheap and efficient."

NVRAM promises to deliver benefits at an equivalent or cheaper per-gigabyte price point as today's flash technology, he says. It could also deliver improvements to today's battery-backed database technology and in NOR flash, which is often used in mobile phones because it consumes minimal energy during the write process.

Marc Staimer, president of Dragon Slayer Consulting, says next-generation NVRAM will introduce new functions and capabilities "that will be developed over time." He believes the technology, like early flash technology, will initially "show up at the consumer end and prove itself out" before enterprises and others begin using it for high-end data center requirements. "You will likely see it in smartphones, tablets, and laptops before you see it in servers and storage systems on a widespread basis."

When the technology does move into data center systems, it will not make NAND flash storage immediately obsolete, just as NAND flash storage did not make hard drives immediately obsolete. "There will be a cost differential that will place these new NVRAM technologies (in a) high-performance solutions tier with a higher cost," he says. "Over time that will gradually change, and variations of the new NVRAM technologies will move downstream to lower tiers, squeezing out slower NAND flash storage."

For now, Handy says the industry must begin to define standards for how these memory technologies will communicate with standard programming interfaces. Meanwhile, system manufacturers will have to place their bets on which new NVRAM technology makes it to market first with the desired characteristics.

Nevertheless, the writing appears on the wall or, perhaps, in the chips. Says Staimer: "These technologies are not just an iteration of existing technology; they are a breakthrough.

> **"Nobody knows which of the horses in the NVRAM game will win or how things will play out, but the bottom line is that the technology will very likely make a big impact on computing."**

This is not just another generation of NAND flash; it is a significant leap forward in performance and wear-life well above today's flash. It will change computer architectures, break down the barriers between memory and storage, and ultimately change how we do computing."

For more on non-volatile memory, see the article by Nanavati et al. in this issue on page 56. C

Further Reading

Pelley, S., Wenisch, T.F., Gold, B.T., and Bridge, B.,
Storage Management in the NVRAM Era, *Proceedings of the VLDB Endowment*, Volume 7 Issue 2, October 2013 Pages 121-132. http://dl.acm.org/citation.cfm?id=2732231.

Oh, S., and Ryu, Y.,
Multi-core Scheduling Scheme for Wireless Sensor Nodes with NVRAM-Based Hybrid Memory, Ubiquitous Computing Application and Wireless Sensor, Volume 331 of the series *Lecture Notes in Electrical Engineering*, pp 45-52. http://link.springer.com/chapter/10.1007/978-94-017-9618-7_5#

Van Essen, B., Pearce, R., and Gokhale, M.,
On the Role of NVRAM in Data-intensive Architectures: An Evaluation, *Center for Appl. Sci. Comput.*, Lawrence Livermore Nat. Lab., Livermore, CA, USA, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6267871&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6267871.

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

ACM Member News

ABSTRUSE REASONING IDENTIFIES SOLUTIONS TO PRACTICAL PROBLEMS

Alan Bundy is a problem solver.

A professor of Automated Reasoning in the School of Informatics at the University of Edinburgh, Bundy utilizes abstruse mathematical reasoning to find solutions to practical problems in the development and maintenance of computing systems.

Author of more than 270 publications and recipient of more than 60 research grants, Bundy's research focuses on the automation of mathematical reasoning, combining artificial intelligence with theoretical computer science.

His current research interest involves the need for automated mechanisms of representational change in autonomous agents "that have to cope with a changing environment to find a mechanism that aids in efficient solutions to problems," he says.

Bundy is proudest of his work on automating the conjecturing of intermediate lemmas—a side-theorem required to prove a main theorem. Bundy's lemma illustrated that the failure of an initial proof attempt of the main theorem could sometimes be diagnosed to suggest the form of the missing lemma. "It's like creatively plotting a cross-country journey," he explains. "We developed an idea of a plan; when it went wrong you could figure out what you needed."

That "generic theorem" resulted in an entire family of proofs that took Bundy over five years to construct.

A fellow of the Royal Society of Edinburgh and of the Royal Academy of Engineers, Bundy was named a Commander of the Most Excellent Order of the British Empire (CBE) by Queen Elizabeth in 2012. "I was nervous and forgot the proper form to address Her Majesty. She's a game lady and pretended not to notice," he laughed.

—Laura DiDio

Society | DOI:10.1145/2843553

Esther Shein

Preserving the Internet

Is the Internet ephemeral by its nature, or can it be archived?

Additional storage awaiting use by the Internet Archive.

IMAGE BY JASON "TEXTFILES" SCOTT

THERE WAS A time when an interested party could find five White House press releases online detailing the number of countries participating in the "Coalition of the Willing" in support of the 2003 invasion of Iraq. That list drew controversy, however, because when the war did not go so well, countries like Togo and Costa Rica objected to having their names associated with it. Consequently, the data was manipulated, says Kaylev Leetaru, who co-authored the report "Airbrushing History, American Style."

Today, two of those five documents are unchanged, two have been deleted, and one is accessible but has been edited from its original form, according to Leetaru. "There was a lot of manipulation, and what came out of that was the White House was altering press releases on a regular basis,'' he says. "It came out that they didn't view these as government documents, but rather propaganda material, so there was no reason to mark that a change was made."

Leetaru, creator of the GDELT (global database of events, language and tone) project, which monitors broadcast, print, and Web news worldwide in over 100 languages, also found an instance in which the White House issued a statement from then-President George Bush about the economy improving. "When it didn't, they went back and edited a press release from a few years prior so they could say, 'look, we were right.'" Altering history online is often done by the government and elected officials when that information turns out to be factually incorrect, Leetaru says.

"For government and elected government officials in particular, they don't want things preserved because people can go back and see what the government actually said, so there's actually a perverse effort not to preserve it,'' says Leetaru, who is also a senior fellow at George Washington University. Such selective archiving is not limited to the government, however; many companies also do not archive information "because there is no benefit and it would create confusion to have old material up,'' he says.

Not to mention embarrassment. Leetaru also cites the (in)famous statement "there is no market for home computers," which can be found online in the 1988 *University of Michigan Computing News*, Volume 3, (http://bit.ly/1PZ7k74).

It is not surprising some people might not want certain statements preserved for posterity, and that websites are updated and information altered; after all, the Internet is ephemeral by nature. Yet in doing so, we are doing a disservice to future generations by not passing on relevant societal information that provides an accurate view of the past. Past ACM president Vint Cerf, long considered the "father of the Internet" and now a vice president and chief Internet evangelist at Google, expressed concern last year that we are in a "digital Dark Age," and that future generations will have little or no record of the 21st century, because so much data is kept in a digital format and technology advances so quickly that old files will be inaccessible.

Leetaru notes that in the European Union there is a "Right to be Forgotten" law that basically says if someone wants information on them removed from the Internet, major search engines like Google must delete it from their search index. Leetaru worries other countries might adopt similar regulations. If something cannot be found on Google, he points out, it essentially does not exist to Internet users, and history is being rewritten. "So if a website is removed from Google and all the major search engines, for all intents and purposes it's removed from the world. That's a really scary situation in terms of our collective remembrance of society."

However, "The Web was never designed for being archived,'' observes Brewster Kahl, founder and digital librarian of the Internet Archive, a not-for-profit digital library whose goal is to preserve the Internet's past for the use of future historians. The average lifespan of a Web page is a little under 100 days before it is changed and/or deleted, he says.

Kahl felt strongly that as the Web grew, the Internet Archive should help provide universal access to all knowledge. "Libraries were useful in the past, and as we're changing a publishing system from paper-based to digital, we should do the same,'' says Kahl. The idea was further fleshed out to provide access to materials no longer available online "and to make it so people could compute on these materials."

About 140 people work at Internet Archive scanning hundreds of books per day, worldwide. The organization receives about $12 million per year in funding, mostly from libraries paying the Internet Archive for its services, as well as donations from foundations, he says.

Kahl also created the Wayback Machine, whose name is a nod to a fictional time machine from the "Rocky and Bullwinkle" cartoon show in the 1960s. It is a three-dimensional index that archives and allows browsing of current and older Web documents. "We started by being a kind of robot that crawled everything we could find," says Kahl, "but it's evolved since then." Some 1,000 librarians build custom Web collections from about 350 different libraries, museums, and archives worldwide, "so now there are lots of people helping select and make sure the Wayback Machine has the right things in it,'' he says. "The Web is not just one thing anymore; it's lots of different collections. At least, that's the way we think of it."

Today, 600,000 people access the Wayback Machine each day, searching for approximately 450 billion Web objects, which include images. About one billion pages are added each week, Kahl says. Leetaru has collaborated with the Internet Archive and Flickr to extract images from 600 million digitized book pages that date back 500 years from over 1,000 libraries worldwide and make them all browseable and searchable via both the metadata of the original book and the text surrounding each image.

Among the other organizations working with the Internet Archive is the U.S. Library of Congress (LOC), which took some heat early last year for failing to digitize its collections. Since 2000, the LOC has been preserving Web content not only from its own Website, but also the sites of other organizations and citizens as part of its Web Archiving program (http://www.loc.gov/webarchiving/), says Abbie Grotke, lead information technology specialist with the LOC's Web Archiving Team. In addition to event-based archives on topics including the U.S. national elections, the Iraq War, and the events of September 11, 2001, the team's focus is on preserving content deemed "at risk." Grotke says this typically includes an event that has occurred but the documentation of it is no longer needed, such as campaign websites that are taken down after an election. It might also include government documents from unstable regions around the world.

The LOC archives different types of collections, such as online news sources, Web comics, and folklore, which are selected by "recommending officers," Grotke says. The team uses a custom

The Wayback Machine is a three-dimensional index that archives and allows browsing of current and older Web documents.

Milestones

Yelick Receives ACM/IEEE Award

Katherine Yelick, a professor of Electrical Engineering and Computer Sciences at the University of California at Berkeley and faculty scientist and Associate Laboratory Director for Computing Sciences at Lawrence Berkeley National Laboratory, was awarded the 2015 ACM/IEEE Computer Society Ken Kennedy Award for innovative research contributions to parallel computing environments that have been used in both the research community and in production environments.

Yelick, who has authored more than 170 technical papers and reports on parallel languages, algorithms, libraries, architecture, and storage, also was cited for her strategic leadership of national research libraries, and for developing novel educational and mentoring tools.

Yelick's work has improved the programmability of high-performance computing through innovations to parallel languages and runtime systems. Her contributions to compiler research and open source software were key to the success of partitioned global address space. She also developed automatic performance tuning techniques and runtime systems that maximize performance across a variety of computer architectures.

An ACM Fellow, Yelick was named 2013 Athena Lecturer by the ACM Women's Council (ACM-W). A member of the National Academies of Sciences, Engineering, and Medicine's Computer Science and Telecommunications Board, Yelick has served on the California Council on Science and Technology and on the University of California Science and Technology Council.

ACM and IEEE co-sponsor the Kennedy Award to recognize contributions to programmability and productivity in computing and significant service or mentoring contributions.

curator tool it built called Digiboard to nominate URLs to a collection, manage them, and track project activities. The tool is also used to request permission to crawl a particular site.

Universities are also making efforts to archive their websites. The Massachusetts Institute of Technology (MIT) has just started doing so, since it did not previously have the tools and technology in place, says Kari Smith, digital archivist at the MIT Libraries' Institute Archives and Special Collections, which uses an international file standard called WARC (Web ARChive) to combine different digital resources into an archival file.

MIT has eight full-time archivists working to preserve not only internal Websites, but also reports and attachments that might come from an administrative or faculty office, Smith says. "We know every semester information changes, because information on a website is about what's happening in a particular semester, so we look at dates that are most likely to change,'' such as those for specific courses. Although some sites change daily, "from an archival perspective, the question is, what can we capture on a daily basis and what is the future value of that for historical purposes?"

Professional archivists at the university decide what gets preserved, she says. "It's a collaborative decision based on our resources and what staffing and technology do we have available."

Other organizations also have as their mission the preserving information for future generations, among them the International Internet Preservation Consortium (IIPC) (http://www.netpreserve.org/) and Perma.cc (https://perma.cc), which helps create permanent links to online sources cited in works by scholars, courts, and journals.

The Digital Public Library of America (DPLA) works with Perma.cc and the Internet Archive, and also preserves the metadata it brings into its online repository through exhibitions, says Kenny Whitebloom, manager of special projects. "We have a small array of digital objects that we've acquired permissions to reproduce online,'' he explains. "We further facilitate access to those materials and help in their preservation on our site in that way, but we don't have a mandate to preserve Web pages or links."

"The World Wide Web is effectively infinite and has parts and corners that can't be archived, and there's large swaths of it that shouldn't be archived, that weren't meant for the ages."

Nearly all Web archivists say it is virtually impossible to preserve all the content on the Web.

"The Web is so massive you can't crawl and digest everything," says Leetaru, "but I'm creating a very targeted collection of worldwide news media," especially local media, and providing URLs to make a permanent copy.

Leetaru does not hesitate when asked if enough government organizations and companies that should be preserving content for future generations are doing their due diligence. "Absolutely not … the average company doesn't care at all. Most companies delete content when it gets old. If you make a product you don't want to keep marketing material [online] because it will create confusion, so they delete material as part of their routine processes." Whereas it used to be costly to put that information on the Web, now companies just want to "clean up" their sites, he says.

Archivists generally do not preserve everything, says Smith, and resources are limited. As a result, the priority is on information that is ephemeral and the "output of people in the world that we can acquire and keep to tell the stories that are incredibly important—especially the stories that most often don't get told. We definitely want to make sure we're collecting actively about minority organizations or small groups that are having a big impact in their community," but may not get wide distribution.

Grotke also says it is possible to preserve portions of the Web, but notes that a lot of sites are constructed in a way that crawlers are unable to access them. "One reason is because of passwords, and current crawler technology is probably five to 10 years behind what the current Web is doing, so a lot of dynamic or flash-based content" such as art Websites that museums are trying to capture, is too complex and challenging for current tools, she says.

Kahl says the key is to capture what we want people in the future to know about the past.

"The World Wide Web is effectively infinite and has parts and corners that can't be archived, and there's large swaths of it that shouldn't be archived, that weren't meant for the ages,'' he maintains. "It's important to be selective. But on the other hand, we try to cast a very wide net, because we're interested in trying to help preserve the creative works of humankind." C

Further Reading

Althaus, S. and Leetaru, K.,
University of Illinois Urbana-Champaign, 2008. Airbrushing History, American Style. http://courseweb.lis.illinois.edu/~katewill/spring2011-502/502%20and%20other%20readings/althous%20and%20leetaru%20Airbrushing%20History.pdf

"Blue Magic": A Review
***University of Michigan Computing News*, Volume 3, Number 19, p. 14, 1988, http://bit.ly/1NoZst2**

Venkataraman, B.,
The Race to Preserve Disappearing Data, *The Boston Globe*, May 17, 2015, http://bit.ly/1XGpKiS

O'Keefe, E.,
"Obama wants better digital archive of federal records," *The Washington Post*, Nov. 28, 2011, http://wapo.st/1LVlx45

Ghosh, P.,
"Google's Vint Cert warns of 'digital Dark Age,'" *BBC News*, http://bbc.in/1LVlB41

Executive Office of the President, "Memorandum for the Heads of Executive Departments and Agencies and Independent Agencies on Managing Government Records Directive," 2012, https://www.whitehouse.gov/sites/default/files/omb/memoranda/2012/m-12-18.pdf

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

In Memoriam | DOI:10.1145/2845948

CACM Staff

Gene Amdahl, 1922–2015

Gene Amdahl, who formulated Amdahl's Law and worked with IBM and others on developments related to mainframe computing, died recently from complications of pneumonia.

AMERICAN COMPUTER ARCHITECT and high-tech entrepreneur Gene Myron Amdahl died recently at the age of 92.

Amdahl's wife Marian said he had suffered from Alzheimer's disease for about five years, before succumbing to pneumonia. "We are thankful for his kind spirit and brilliant mind. He was a devout Christian and a loving father and husband. I was blessed with having him as my husband and my best friend."

Born in South Dakota, Amdahl served in the U.S. Navy during World War II. He completed a bachelor's degree in engineering physics at South Dakota State University in 1948 and went on to study theoretical physics at the University of Wisconsin-Madison, where he received his doctorate in 1952.

Amdahl joined IBM in 1952, where he worked on the IBM 704, the IBM 709, and the Stretch project, the basis for the IBM 7030. He left IBM in 1955 but returned in 1960 and became chief architect of the System/360 mainframe computer. Amdahl was named an IBM Fellow in 1965, and head of the IBM Advanced Computing Systems Laboratory in Menlo Park, CA. He left IBM again in 1970 and set up Amdahl Corporation, which specialized in IBM mainframe-compatible computer products, with the help of Fujitsu.

The company manufactured "plug-compatible" mainframes, starting in 1975 with the Amdahl 470V/6, a less expensive, more reliable, faster alternative to IBM's System 370/168. Amdahl's software team developed Virtual Machine/Performance Enhancement (VM/PE) software to optimize the performance of IBM's Multiple Virtual Storage (MVS) operating system when running under IBM's VM operating system. Within four years, the corporation had sold more than $1 billion worth of V6 and V7 mainframes and had more than 6,000 employees worldwide.

PHOTO BY LOUIS FABIAN BACHRACH

At ACM's Spring Joint Computer Conference in 1967, Amdahl participated in a discussion on future architectural trends, arguing for performance limitations in any special feature or mode introduced to new machines. This resulted in what came to be known as Amdahl's Law regarding sequential vs. parallel processing.

Amdahl left his company in 1979 to set up Trilogy Systems, a company aimed at designing an integrated chip for even cheaper mainframes. When the chip development failed within months of the company's $60 million public offering, Trilogy focused on developing its VLSI technology, which also did not do well. In 1985, Trilogy was merged into microcomputer manufacturer Elxsi (now Tata Elxsi), but poor results there had Amdahl leaving in 1989 for a company he had founded in 1987 to produce mid-sized mainframes, Andor International, which by 1995 had been driven into bankruptcy by production problems and strong competition.

In 1996, Amdahl co-founded Commercial Data Servers, again developing mainframe-like machines, but this time with super-cooled processor designs aimed at physically smaller systems. The company, Xbridge Systems, develops software to scan mainframe datasets and database tables for sensitive information such as credit card numbers, government identification numbers, and medical diagnosis information.

Amdahl was a member of the National Academy of Engineering and the recipient of honorary doctorates from four institutions. He also was the recipient of the IEEE's Harry H. Goode Memorial Award, a Fellow of the Computer History Museum, and recipient of the ACM Special Interest Group on Design Automation (SIGDA) Pioneering Achievement Award.

Said David Patterson, a professor of computer sciences at the University of California, Berkeley, and a computer pioneer in his own right, "The IBM System/360 was one of the greatest computer architectures of all time, being both a tremendous technical success and business success. It invented a computer family, which we would call binary compatibility today. When he left to form his own company, his mainframes were binary-compatible with the System/360."

Patterson noted Amdahl's Law was based on a brief paper (http://bit.ly/1rUSzL7) submitted to ACM's Spring Joint Computer Conference "basically offering a critique to enthusiasts about the parallel supercomputers of the era." He added that Amdahl's Law "may be obvious, but architects still keep being caught by it, seeing their hopes dashed for their cool ideas."

In addition to Amdahl's Law, Patterson said, "Less well-known are Amdahl's rules of thumb for a balanced computer system," such as "A system needs a bit of IO per second and one byte of main memory for each instruction per second."

Software engineer, computer scientist, and 1999 ACM A.M. Turing Award recipient Fred Brooks, a colleague of Amdahl's during his time at IBM, recalled a number of instances of Amdahl's "architectural and implementation brilliance" that allowed his IBM 704 to "utterly dominate its chief competitor, the Univac 1103A, in the scientific computing market."

Brooks added, "We have indeed lost one of the truly great computer designers."

DOI:10.1145/2846082

Jonathan T. Weinberg

Law and Technology

Biometric Identity

Assessing the promises and dangers of biometric identity plans.

THREE YEARS AGO, the U.S. Senate passed a comprehensive immigration reform bill. The drafters of that bill pushed for a requirement that every employed person in the U.S.—whether citizen or noncitizen, native-born or immigrant—should have to get a federal government-issued ID card. The holder's biometric information, either fingerprints or a different technology, would be encrypted on the card. Every time a U.S. worker took a new job, the employer would take her fingerprints or other biometric, so as to check her physical characteristics against the information on the card. If the biometric information matched, it would establish the job applicant was the card's rightful bearer. The employer would then transmit the identity information on the card to a central database, to verify she was legally authorized to work. In the end, though, the drafters dropped the ID card proposal from the bill.

In India, the government is undertaking to assign to residents 1.2 billion unique "Aadhaar" ID numbers, linked to each person's biometrics—photograph, 10 fingerprints, and two iris scans. The government aims to make use and verification of one's Aadhaar number an inseparable part of daily life. The card is accepted as identification and proof of address for banking purposes; authorities are pushing forward with plans to use Aadhaar to scrub voting lists; and a host of government agencies are making it mandatory under their programs, all notwithstanding an interim order by India's Supreme Court forbidding such requirements.

Biometric information helps connect abstract legal status to the physical individual.

Both of these stories involve databases with two features. First, they include entries for all or the vast majority of a country's residents. Second, there is a mechanism to tie the data entries to the subjects' biometric characteristics, which can be checked or verified in the field. In that way, the physical person—showing up for work, or presenting herself at an ATM, or seeking health benefits from a government clinic—can be connected to her identity and description in the database. The U.S. plan has been the pet project of a few senators for years, but has never become law. The government of India, by contrast, has invested 50 billion rupees (US$775,000,000) in its project and has collected biometric information from 800 million people so far. That country's Supreme Court, though, is currently pondering the constitutionality of the plan.

Are plans like these desirable? They present some policy advantages; biometric identification techniques enable governments to achieve certain popularly supported goals more successfully. In the U.S., the law forbids employers to hire people who are in the country illegally or on temporary visas, unless the Department of Homeland Security has granted them work authorization. But whether a person at a particular moment has legal work authorization (or legal immigration status) is not apparent when looking at her; the information resides in a database in Washington, D.C. She may present government-issued documents, but those docu-

An employee uses an Aadhaar-based entry system to verify identity at a building in New Delhi, India.

PHOTO BY VIPIN KUMAR/HINDUSTAN TIMES VIA GETTY IMAGES

ments may be somebody else's; taking her biometric information helps connect abstract legal status to the physical individual.

Pakistan was able to rely on biometric (fingerprint-based) ID cards to provide reconstruction grants to families affected by severe flooding, without too much money going astray. Some countries take voters' biometric data in order to de-duplicate voting lists (that is, to ensure single individuals do not appear on voter registration lists multiple times). Integration of biometric identification into the system for paying government employees in Nigeria is said to have helped uncover more than 60,000 "ghost workers."

More generally, people need some way of verifying their identity so governments will provide them with services and businesses will enter into relationships with them. Governments want satisfactory proof of identity, and often proof of residency or citizenship (which in turn is predicated on proof of identity) before they provide payments such as pension or welfare benefits, or allow individuals to vote, or grant passports or register property transfers. Private actors require people to verify their identity before they can take such steps as opening a bank account, renting an apartment, or cashing a check.

In the industrialized West, these concerns have been addressed primarily through birth registration: Children are registered with the state at birth, and are entitled to documents as proof. They can use those documents to get others such as driver's licenses and passports. All of those documents, tied to an entry in some official database, can be used to verify the holder's identity. But in some poorer countries that does not work, because as many as 70% of all births go unregistered. That is why projects like Aadhaar, in a variety of less-industrialized countries, are exploring the use of biometrics as a way of tying individuals directly into identity-verifying databases.

The connection of our physical bodies to entries in government databases, though, is also problematic. Consider the main episodes in U.S. history where government not only issued biometric ID, but required persons to carry that ID. Before the Civil War, free blacks were sometimes required to carry certificates that recited their names and employers and included the mid-19th-century version of biometrics: they described the worker's physical characteristics, including such matters as age, complexion, build, height, and scars. A free black without adequate identification risked being arrested or enslaved. After 1892, U.S. law required all Chinese persons in the U.S. to carry "certificates of residence" validating their immigration status, on penalty of deportation. Congress mandated that each card contained the holder's photograph; that biometric, said a senator, was "the only effective method" for identifying Chinese migrants.

When the U.S. government next told a group of people they had to carry cards with biometric identifiers at all times, it was 1952. The card in question was the "green card" issued to noncitizen residents in the U.S.; the motivation was fear of the Communist threat. Congress members, worrying that outsiders sympathetic to enemy countries would act as a fifth column, mandated that all noncitizens carry their immigration documentation wherever they went. That law is still on the books today.

That is not a confidence-inspiring record. When U.S. law has imposed requirements that certain people carry biometric ID at all times, it has been so a target could be required to show a document linking him to a dataset telling law enforcement officers whether to enslave, detain, or deport him. That is the promise and the danger of biometric ID systems. ID systems without a biometric component have limited law-enforcement value, because they lack good mechanisms by means of which police can connect the persons standing in front of them to the documents they produce. Biometric ID systems enable better identification, but more effective policing carries risks of its own.

It is perhaps not coincidental, then, that modern U.S. thinking incorporates a severe allergy to anything that looks like a biometric national ID card. Americans have accepted the Social Security number, which in practice serves as a unique common identifier linking them to entries in a variety of federal and private databases. They have accepted a requirement that they carry—and often produce—driver's licenses while driving. But they need not carry or display driver's licenses at other times, and driver's licenses do not display a unique common identifier that could reliably identify the holder across federal databases. In particular, it is illegal for a driver's license or any other state-issued identification document to display the holder's Social Security number.

Besides the baseline concerns associated with police being able to easily and effectively identify citizens, one can identify a wide range of more nuanced risks flowing from government identity systems' coming to rely on biometric identifiers and a central database in some manner associated with them. One risk relates to data security: Can the government keep this

It is important to be mindful of the substantial privacy risks associated with biometric identity plans.

information safe, avoiding either privacy breaches or identity theft? One of the claims made by plaintiffs in the Aadhaar litigation is that data security for the submitted information is unacceptably weak. A second risk relates to the damage done by (the inevitable) bad information in the database, especially if use of the card or biometrics becomes ubiquitous. Will the database become so useful it is treated as presumptively correct, with bad information difficult or impossible to change? Intentionally planting bad information would then be an excellent route to identity theft or worse.

Another concern: To the extent the use of a biometric or card to verify identity becomes routine in everyday transactions, it would be easy to structure the system so each use of the card adds information to the relevant databases. That would fatten the data portfolios maintained on each citizen and would limit individuals' ability to undertake everyday activities free of surveillance. Finally, the government might gain leverage over the citizenry through its power to revoke or limit the use of card or biometric data to verify identity—what happens when government decides to flag the database entries of undesirable citizens so their biometrics or cards can no longer be used to obtain services?

The implementation details of any biometric identity plan are key. The designers of the proposed U.S. worker ID plan, thus, sought to forestall objections by ensuring no biometrics would be stored in the central database; rather, their plan was that biometrics would be stored only on individuals' cards, to be checked against their physical characteristics using card readers in the field. That way, central government authorities would not have access to the biometrics at all. The Aadhaar plan, by contrast, does not rely on cards: biometrics are stored centrally so a person's merely presenting his fingerprints identifies him to the system.

Another approach avoiding some risks would tie citizens' biometrics only to limited-purpose databases—designed for particular functions and not calling up other information in the government's possession—rather than to an all-encompassing database (or a linked set of databases) containing multiple classes of information. Such functional structures can be less expensive than multipurpose identity platforms (although not if a country ends up establishing multiple, separate biometric systems serving separate goals). Some countries, for example, have done biometric registration for the limited purpose of enabling voting in national elections. But limited-purpose biometric identity plans can find their focus shifting; in several countries that have set up single-purpose voter registration systems, the voter registration card has become a de facto national ID card. In the U.S., the Social Security number, created for a limited purpose, rapidly became a unique common identifier.

It is important to be mindful of the substantial privacy risks associated with biometric identity plans. This does not mean they are always a bad idea; in a country where many individuals have no means of identity verification at all, some form of appropriately structured biometric identification system can make people better off. But we have not done well in the past in the U.S. imposing biometric ID requirements, and—given the strength of our existing systems for identity verification—the risks (and costs) of any such plan in the U.S. would likely far outweigh the benefits.

Jonathan T. Weinberg (weinberg@wayne.edu) is a Professor of Law at Wayne State University in Detroit, MI.

DOI:10.1145/2846084

Michael A. Cusumano and David B. Yoffie

Technology Strategy and Management

Extrapolating from Moore's Law

Behind and beyond Microsoft, Intel, and Apple.

ALL GREAT STRATEGIES start with a vision of the future. For entrepreneurs and company leaders, the vision should include a sense of what opportunities lay ahead, what kind of organization can exploit those opportunities, and what products or services customers are likely to buy. The devil, of course, lies in the details.

To get all the details right, successful leaders rely on both extrapolation and interpretation to "look forward" into the future and then "reason back" to what they need to do today and over the next several months. Extrapolation is relatively easy: information on industry trends is widely available. However, someone has to interpret that information—identify key changes, opportunities, and threats for a specific organization and market. Interpretation is where visionary leaders make their mark, as we can see in the companies once led by Andy Grove, Bill Gates, and Steve Jobs.[a]

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

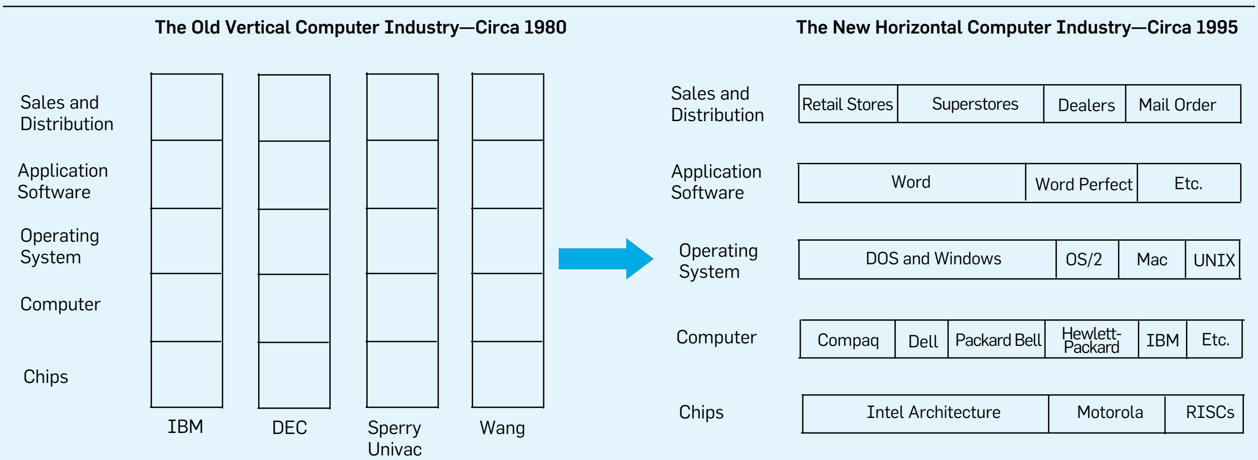
Grove and Intel

Andy Grove, trained as a Ph.D. in chemical engineering at Berkeley, was employee number one at Intel, founded in 1968. He quickly took charge of engineering and then other operations. He became president in 1979 and CEO in 1987, when he had to clarify the company's strategy. Grove would base his vision on an extrapolation from "Moore's Law." Recall that, in a 1965 article, Gordon Moore, who co-founded Intel with Robert Noyce, predicted the number of transistors on an integrated circuit would double every 18 to 24 months.[b] Some people saw Moore's Law as just another example of progress in engineering. Grove interpreted it as a strategy that would

a This column is adapted from a new book by David B. Yoffie and Michael A. Cusumano, *Strategy Rules: Five Timeless Lessons from Bill Gates, Andy Grove, and Steve Jobs* (Harper Business, 2015), 23–59.

b See also Michael S. Malone, *The Intel Trinity: How Robert Noyce, Gordon Moore, and Andy Grove Built the World's Most Important Company* (Harper Business, 2014).

The transformation of the computer industry (not to scale).



Source: Re-created from Andy Grove Intel presentation, with permission.

transform the structure of the computer industry. He concluded that, if Intel continued to pursue Moore's Law, competitors would need massive scale economies to produce integrated circuits. Inevitably, this would topple vertically integrated giants like IBM and Digital Equipment Corporation (DEC) that had dominated the industry for decades.

Several years before it became obvious to the world, Grove foresaw the rise of an industry organized in horizontal layers—chips, hardware, operating systems, applications, distribution—each dominated by a small number of powerful companies (see the accompanying figure).[c] Based on this vision, he focused Intel's strategy on leadership in the microprocessor segment. The top priority became the innovations needed to double the transistors on an integrated circuit every 18 to 24 months.

This evolution in Grove's thinking did not happen all at once. In 1987, he proclaimed 50% of Intel's business should be "systems" of fully assembled computers. By 1990, he believed the company should focus on its core strength—microprocessors. In the future, Intel would make products such as motherboards and chipsets that would help sell microprocessors. But Intel would steer far away from layers of the computer industry dominated by large product companies with scale economies on their side.

Gates and Microsoft

Bill Gates also built his vision of the future on Moore's Law, but in a different way. Gates saw the future through his own "personal anchor" in software, with a deep understanding of how to program the early microprocessors Intel was producing. Gates believed the repeated doubling of computing power would turn hardware into a commodity, leaving software as the true source of value. In a 1994 interview, he recalled his thinking when launching Microsoft in 1975: "When you have the microprocessor doubling in power every two years, in a sense you can think of computer power as almost free. So you ask, why be in the business of making something that's almost free? What is the scarce resource? What is it that limits being able to get value out of that infinite computing power? Software." [d]

This insight was revolutionary and prophetic, as was Gates' conviction there would one day be a personal computer on every desk and in every home. Gates proclaimed this vision when industry luminaries such as Ken Olsen of DEC and even Gordon Moore of Intel believed home computers were a silly idea. Gates disagreed, and in 1975 dropped out of Harvard to make his vision of the future happen. Later in his career, Gates delegated some of the work of extrapolating from the present to others. But, until he stepped down as CEO in 2000, Gates led the way when it came to interpreting how new trends such as the Internet would impact Microsoft's strategy and product portfolio.

Like Grove, Gates was highly disciplined when it came to strategy and execution. Co-founder Paul Allen originally wanted to produce hardware and software, but Gates insisted they should focus on software. Microsoft started with programming languages and then set out to dominate PC operating systems, first through MS-DOS and then Windows. As secondary fronts, Microsoft added applications and then servers, browsers, and other software products that complemented the Windows platform, largely ignoring hardware until the Xbox in 2001.

Steve Jobs and Apple

Like Andy Grove and Bill Gates, Steve Jobs took inspiration from the advances in computing power described by Moore's Law. Unlike Grove and Gates, however, he was not a technologist by training and wanted to see computing devices made as simple to use as a toaster or a typewriter. This

c Grove also published this graph in his best-seller book, *Only the Paranoid Survive* (Currency/Doubleday, New York, 1996), 44.

d "Playboy Interview: Bill Gates," *Playboy* (July 1994), 63.

focus led to Jobs' goal of transforming complex personal computers into "insanely great" consumer products defined by simplicity and ease of use. Eventually, his vision for Apple (founded in 1976) expanded beyond creating individual products to designing the entire digital experience.

By the late 1990s, Jobs and others had come to realize the explosion of devices was creating a digital Babel, made worse by poor usability and connectivity. He also had a solution. In 2001, Jobs told MacWorld attendees the Macintosh (originally introduced in 1984) "can become the 'digital hub' of our new emerging digital lifestyle, with the ability to add tremendous value to these other digital devices."[e] With its focus on consumers and the user experience, Apple was uniquely suited to deliver on this vision. Ron Johnson, former head of Apple retail, explained how Jobs' concept of a digital hub set Apple on a new path: "[The digital hub vision] created a mental roadmap for products ... how Apple would win in the marketplace. Apple had been locked into a PC model for most of its history and this liberated the company to be relevant in all emerging categories from music players, to cameras and beyond. It really became how we allocated resources."[f]

Jobs also saw focus as a central element in a successful strategy, explaining that, "the way we've succeeded is by choosing which horses to ride very carefully."[g] While Jobs was out of the company during 1985–1997, Apple did not follow this rule. When he returned in 1997, Jobs found the company's product portfolio too broad and weak. In one meeting, out of frustration he drew a simple grid, labeling the columns "Consumer" and "Professional" and the rows "Desktop" and "Portable." He insisted that, going forward, Apple focus on just four products, one for each quadrant in the grid. And even within the professional segment, Jobs later told company executives to abandon the enterprise market.[h]

What will be the next equivalent of Moore's Law? Will it again transform the world?

From Vision to Strategy

The visions of Gates, Grove, and Jobs are noteworthy not only for their ambition, but also for their clarity and simplicity. Clarity and simplicity, however, are not the same as immutability. These visions did not spring fully grown from the minds of their creators. They were continuously revisited, revised, and redefined as new events and information emerged.

Grove, for example, refined his vision over five years as he transformed Intel from a broad-line maker of mostly commodity semiconductor memory products into a microprocessor company and platform leader in the computer industry. Les Vadasz, one of Grove's longtime senior executives, explained how Grove managed this strategic transition: "You can only look so far, and so you better just keep looking frequently. That's the most important element of strategy: You understand the direction you're going, but you also know what you're going to do in the next six months. Most companies will do a pretty good job many times about the direction, but then they never break it down to shorter metrics. Intel did a super job on that. When you ask why [we] succeeded, this is one of the reasons."[i]

Gates moved Microsoft in the opposite direction, broadening its product portfolio over time—but maintaining a tight focus on software. Similarly, Jobs' vision for Apple evolved continuously, from personal computers to the digital hub, and then to digital media, smartphones, software distribution, tablets, and "the cloud." But Apple under Jobs always remained tightly focused on producing simple, elegant products and services for the consumer—not for technologists or enterprise users per se.

Former IBM CEO Lou Gerstner once said, "Vision is easy. It's so easy to just point to the bleachers and say I'm going to hit one over there. What's hard is saying . . . how do I do that."[j] In other words, vision is essential but never an end in itself. Leaders must translate vision into strategy that defines the scope of an organization's activities—what it will and, perhaps even more important, what it will *not* do. The ability to update visions in response to changes in the environment, while preserving clarity and focus at the core of their strategies, was an important strength Gates, Grove, and Jobs shared.

How Moore's Law lay behind the founding of Microsoft, Intel, and Apple should also make us wonder about the future: What will be the next equivalent of Moore's Law? Will it again transform the world? The Internet as well as mobile computing and cloud-based services are technologies that, in some sense, all flowed from Moore's Law. They are also distinct innovations. But how should we think about what comes next? How frequently do we need to revise our assumptions? What are the implications for hardware and software platforms as well as digital services? Who should be in on these conversations about the future? These are questions not only for technologists and entrepreneurs, but for all organization leaders and society more broadly. ■

e "Steve Jobs introduces the 'Digital Hub Strategy' at Macworld 2001" (Jan. 9, 2001); https://www.youtube.com/watch?v=9046oXrm7f8

f Ron Johnson, interview with the authors, Oct. 10, 2013.

g Quoted in Adam Lashinsky, "How Apple Works: Inside the World's Biggest Startup," *CNNMoney* (Aug. 25, 2011); http://tech.fortune.cnn.com/2011/08/25/how-apple-works-inside-the-worlds-biggest-startup

h Fred Anderson, interview with the authors (Oct. 9, 2013).

i Les Vadasz, interview with the authors (Oct. 7, 2013).

j "In Focus: Lou Gerstner," CNN.com (July 2, 2004); http://edition.cnn.com/2004/BUSINESS/07/02/gerstner.interview/

Michael A. Cusumano (cusumano@mit.edu) is the SMR Distinguished Professor at the MIT Sloan School of Management and School of Engineering.

David B. Yoffie (dyoffie@hbs.edu) is the Max and Doris Starr Professor in the Strategy Unit at Harvard Business School and the longest-serving member of the Intel board of directors.

DOI:10.1145/2846086

Phillip G. Armour

The Business of Software
The Chaos Machine

Complexity, predictability, and modern projects.

IMAGE BY RICKY ROMERO

MODERN SOFTWARE PROJECTS can be quite complex. There are many interacting variables and many feedback loops where an action effects an action that changes the first action. If the loops provide negative feedback they will tend to correct ineffective behavior and poor results but sometimes we may attempt solutions that actually accelerate the problems. A classic case of this is when a project experiences schedule difficulties and throws additional staff at the problem, which often causes even worse slippage. In many cases, ill-informed attempts to control projects can actually result in considerable turmoil, even chaos.

The Chaos Machine

At a workshop on the management of modern technology projects to an audience of about 20 telecomm project managers, I addressed how projects and management have changed over the years. By way of a metaphor, I unveiled the "Chaos Machine."

I set the machine in motion and watched the audience's reaction. They were clearly quite surprised at how it worked because how it worked was, well, quite surprising. I stepped around the table on which the machine stood and turned to the audience to make some points about what was happening. A sudden noise behind me startled the workshop participants. "Did you see that?" One of them remarked, "… the darn thing just hopped. All by itself."

To call it a "Machine" might be a little strong since it was really just a pendulum system. Into a heavy wooden base were embedded two sturdy vertical metal supports. Across the top of the supports was a taut wire. From this wire, and tightly bonded to it, were three pendulums of equal length evenly spaced along the wire. Each pendulum was a heavy builder's plumb bob. Each of the metal supports had clips that allowed me to take one or two of the pendulums out of the system.

Simple and Predictable

The first demonstration used the simplest setup of the Chaos Machine (see Figure 1).[a] as a metaphor for the simple projects of yesteryear. Clipping the outside two pendulums to the supports left only the middle one free; I pulled the remaining pendulum a short distance perpendicular to the horizontal wire and asked the participants: "What will happen when I let this go?" The audience response was quick and confident: everyone agreed the pendulum will swing back and forth at a regular rate.

This system, like projects of long ago, is simple and very predictable. The rate of oscillation is a function of the length of the pendulum and the amplitude dependent on the initial displacement modified by normal friction. Given the initial conditions, the behavior of the system at any point in time is predictable; so much so that we can, and do, create clocks out of similar systems.

Even easy projects are not as simple as a pendulum. But projects with few interacting variables tend to be well controlled, predictable and, well, easy.

Complex and Predictable

The next step was to take out the center pendulum and put the other two back

a For a different metaphor on the same topic, see Armour, P.G. "Zeppelins and Jet Planes." *Commun. ACM 44*, 10 (Oct. 2001), 13–15.

into play (see Figure 2). I asked the audience: "If I let these two pendulums go at the same time, what will happen?" The audience was less sure, so I pulled them back and let them go.

The behavior with two pendulums is interesting. The periodicity of each pendulum is still a function of their length. But there is also an exchange of energy through the twisting of the horizontal wire and it affects the pendulums. If one plumb bob swings a lot, it twists the horizontal wire. This transfers energy to the other pendulum, which causes it to swing across an increasingly wider arc. What the audience saw was that one pendulum would slow rapidly, as if a brake were being applied, until it stopped completely while the other would speed up equally dramatically. Then the energy would be transferred back to the first pendulum and the wildly swinging pendulum would slow down while the one that had stopped would speed up until it was swinging across a wide arc and the other one would be motionless.

Conservation of momentum applies: when one pendulum was stationary the other would be at its maximum displacement. This system has two periodicities: the swinging of the pendulums and the transfer of energy across the horizontal wire. The rate of the second is a function of the twisting moment of the wire and is somewhat independent of the periodicity of the swinging pendulums. This is a more complex system but, like the simple pendulum, it is also largely predictable—from a known starting condition a future state can be computed with some effort.

When projects get more complex, our simple models no longer work well.

When projects get more complex, our simple models (think: the waterfall development cycle) no longer work well. Different and more complex project variables interact and change how the project will behave.

Complex and Unpredictable

In the workshop, we moved on to a discussion of the challenges of modern technology projects with all their interacting elements and it was time to fire up the most complex level of the Chaos Machine (see Figure 3). With all three pendulums free, I asked the audience: "What will happen now?" The participants did not have an answer.

This system is example of the "Three Body Problem." It has been the source of much thought and calculation since the time of Isaac Newton and the French mathematician Henri Poincaré was awarded a prize from the King of Sweden for his work on it. When a system has three or more interacting variables it is possible and even likely the system will be intrinsically unpredictable—even knowing everything about the system *a priori*, it may be impossible to predict its future performance.

I let the pendulums go. Sometimes two of them would stop and one would move, then the stopped pendulums would start up and the moving one slow down. Their behavior is quite arresting. I let the pendulums do their thing and turned back to the audience.

This was when the system "hopped." It hit a chaos point where the interaction of the pendulums became turbulent and some 20 pounds weight of base, supports, and pendulums hopped across the table. The pendulums crashed together and the system stopped.

More than Three Variables

This demonstration was intended to illustrate a point about modern proj-

Figure 1. The Chaos Machine with one variable.

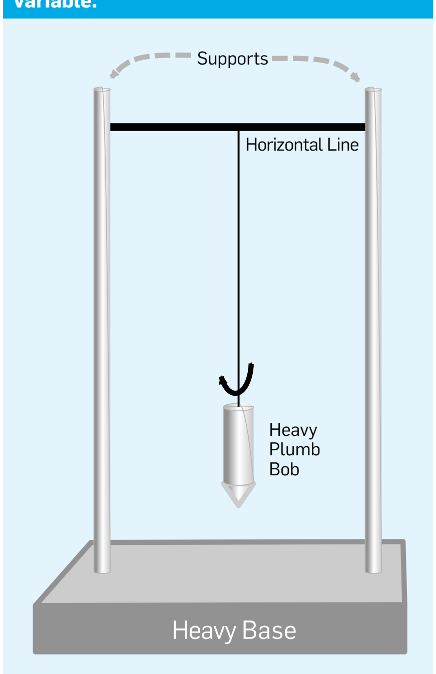


Figure 2. The Chaos Machine with two variables.

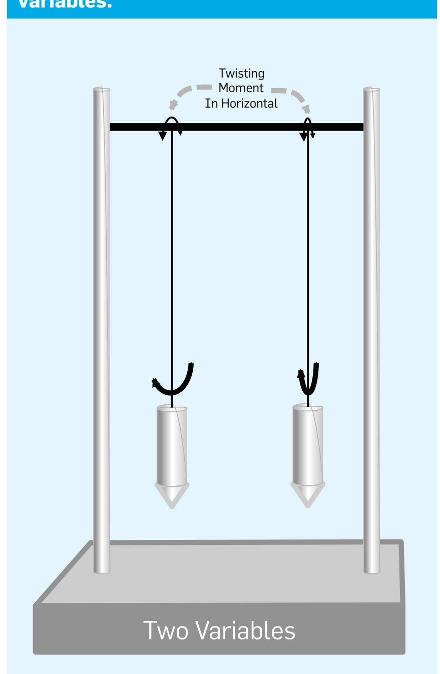
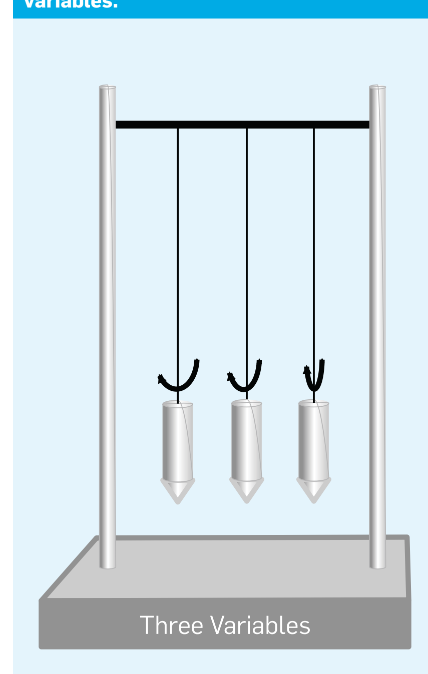


Figure 3. The Chaos Machine with three variables.



ects and how managers should view them. With a significant number of interactive variables (as in three or more), it is possible, even likely, that an unpredictable system can result; that is, the outcome will often not be predictable at the beginning of the project. It is also possible a chaotic system can develop, that at some point the interaction of parts of the project can become unstable, they can "crash together" and whole thing could grind to a halt.

Modern projects have a lot more than three interacting variables so they could behave in ways that are challenging, puzzling, and unforeseen. On the plus side and unlike a pendulum system, projects can be self-monitoring and self-modifying. So with the right measurements in place, they should be able to detect a potential lapse into chaotic behavior and take corrective action before it does.

Planners

At this workshop, I was talking to a group with a reputation for being highly planning focused. They assumed that, if a project was intensely and minutely planned, everything would naturally and inevitably proceed in an orderly and deterministic way. As project planners, they further presumed any variance from the plan must be a result of inadequate or incomplete planning; when confronted with variation in their projects they tended to try to plan their way to success. And when projects did go off the rails, their response was to stop everything and replan. Sometimes that works, but sometimes it does not and managers of today's projects must have more flexible approaches to managing changes in their projects.[b] Modern agile project management approaches have been developed to deal with these complex and unpredictable situations. But there are still plenty of by-the-book project planners out there who are surprised and sometimes baffled by the way their projects behave.

The Chaos Point Redux

I did not see the Chaos Machine crash. But it certainly surprised the audience (and me) and made an interesting and memorable point that gave us some talking points in the rest of the workshop.

I retired the Chaos Machine to my basement and sometimes, when working on something else, I would set the pendulums a'swinging and wait hoping to see them hit the chaos point again. But they never did. So perhaps this holds out some hope: it is very clear that modern projects in the business of software are relatively unpredictable compared to projects of the past and require more attention and dynamic and intelligent responses. But chaos and crashing is not inevitable.

Still, maybe I will pop down to the basement and give the Chaos Machine another shot, just in case. C

b Some resources that discuss the interaction of management and chaos include Margaret Wheatley's *Leadership and the New Science*, Berret-Koehle Publishers, 3rd edition, 2006; and Tom Peter's *Thriving on Chaos*, Harper Perennial, 1988.

Phillip G. Armour (armour@corvusintl.com) is a vice president at Applied Pathways LLC, Schaumburg, IL, and a senior consultant at Corvus International Inc., Deer Park, IL.

DOI:10.1145/2846088

Thomas Haigh and Mark Priestley

Historical Reflections

Where Code Comes From: Architectures of Automatic Control from Babbage to Algol

Considering the evolving concept of programming.

In our previous *Communications* column (September 2015) we noted that a celebrated table published by Ada Lovelace in 1843 was not a computer program, despite frequent claims to the contrary. Here we turn to a related question: where did computer code come from? Back in the 1840s nobody talked about "programming" Charles Babbage's planned engines. More importantly, nobody had yet formulated the concept of a program as a series of instructions controlling the operation of a general-purpose computer. The work of Charles Babbage and Ada Lovelace provides an important milestone on the road to this invention, but marks the beginning of the story rather than its end.

In this column we explore the rest of that story, returning briefly to the world of Lovelace and Babbage before moving on to the 1940s when their ideas were independently rediscovered, extended, and finally realized in actual machinery. Developments came thick and fast, moving in just a few years from the earliest relay computers controlled by "coded" arithmetic instructions on tape to ENIAC, the first computer to automatically carry out computations with complex structures including branches and nested loops. This was the context in which the word "programming" was initially applied to a computer, originally to describe the action of the machine's control wires, circuits, and switches when triggering the appropriate sequence of mathematical operations. Before ENIAC was even finished its creators, in collaboration with John von Neumann, had come up with a new approach in which control and arithmetic operations were both represented in a single series of coded instructions stored in an addressable memory unit. That was soon called a computer program, and although the meaning of the term has continued to evolve it has retained this basic sense of a set of instructions that direct the performance of a series of operations, enabling a computer to carry out a task without human intervention.

U.S. Army photograph of the installed EDVAC.

U.S. ARMY PHOTO

Babbage Proposes a Control Operation

Unlike Babbage's earlier proposed calculator, the Difference Engine, the Analytical Engine was what would later be called a "general-purpose" automatic computer. This flexibility meant it would have to be "ordered" or "instructed" to carry out whatever particular sequence of arithmetic operations (addition, subtraction, multiplication, or division) was needed. These orders were punched onto "operation cards" joined in a continuous chain, forming something analogous to the control roll of a player piano or the paper tapes used by later computers.[a] A separate sequence of "variable cards" told the Engine which locations in its "store" to use for the arguments and results of each operation.

Babbage recognized that punching a card for every operation needed in a complex calculation would be inefficient and inflexible, as most computations have a structure in which sequences of operations are repeated. Fully automating computation meant making explicit and mechanizing not just sequences of arithmetic operations but also the control processes needed to direct these repetitions. Babbage envisioned mechanisms to "back up" the chains of cards to repeat a sequence of operations. The rewinding operation would be triggered by special "combinatorial cards" placed among the operation cards. When Lovelace published details of her planned computation, in what has become one of history's best-known endnotes, she relied on this capability when defining two nested loops. In our previous column we observed that her table omitted the control information needed to direct the Engine through these loops and was closer to being a simulated trace than an actual program. She described the overall structure of the computation more abstractly in a symbolic expression inspired by mathematical notation.

This omission is not surprising, as Babbage's ideas about the backing-up scheme were still rather provisional. Only in 1844 did he even prepare lists of the operations that his engine would support.[1] Babbage specialist Allen Bromley described them as documenting a "programmer's interface." Along with the expected arithmetic operations, Babbage defined an operation of "Ascertaining if any Variable = 0." The card ordering this operation would specify how many cards were to be skipped if the variable defined by the current variable card was zero. That marked a crucial generalization of the notion of "operation" beyond the familiar operations of arithmetic to encompass the control operations that determine exactly which sequences of arithmetic operations are carried out.

Punched Cards and Analog Computers

The Analytical Engine was never built, or even completely and stably designed. Over the next nine decades a variety of calculating and counting machines were developed, including various kinds of punched card tabulating equipment and a number of differential analyzers. None of their designers attempted to revive Babbage and Lovelace's pursuit of a general-purpose automatic computing machine. Configuring these machines was not called "programming."

The punched card machines each tackled a specialized operation, such as tabulating cards or sorting them. Each could be wired to carry out a particular variant on this task, for example to ignore some columns on the card while calculating totals and subtotals based on others. While these machines were applied to scientific calculations from the 1930s onward most of the work we think of as executing a program was carried out by human operators, not by the machines themselves.

Analog computers, such as differential analyzers and gun-control systems, took a fundamentally different approach, representing numbers not as digits but as continuously varying quantities. Changing variables were modeled as changes in voltage or mechanical rotation in a particular unit within the machine. The computers could be configured, often with wrenches or screwdrivers, to specify particular relationships between these units, modeling the terms in an equation. Each piece of the machine performed a single task throughout the computation. These machines were not following instructions as they computed, and so the concept of a program, according to which devices carry out a sequence of different operations over time, does not apply to them.

Codes and Coding

We shall therefore jump forward from the time of Babbage directly to the early 1940s, when the first automatic digital computers were being built. Their capabilities were similar to those Babbage planned for the Analytical Engine, though they were conceived without knowledge of his designs and most used electromechanical relays rather than cogwheels to represent numbers. They included the Mark I computer built by IBM for Harvard University, a series of wartime machines built by Bell Labs in New York, and the Z3 built by Konrad Zuse in Berlin.

Like the planned Analytical Engine, these machines carried out sequences of arithmetic operations, now represented as patterns of holes punched in control tapes. These patterns were called "codes," mirroring earlier uses of "code" to describe the encoding of messages onto paper tape for machine transmission (Baudot code) or into dots and dashes for human transmission (Morse code).[b]

The designers of Mark I thought of these orders as primarily arithmetical, each specifying an operation to carry out as data was transferred from one register to another. Rather than winding the control tape back to repeat sequences, loops were implemented in the most literal sense possible: gluing the ends of the control tape together to form a physical loop. This is, we strongly suspect, the origin of the term "loop." Mark I thus used coded orders for arithmetic, but not for control structures. Loops were mapped onto the physical configuration of the tape, and transfers of control were carried out manually by humans. It took code, paper loops, and humans to carry out the functions later automated with program code alone, so specifying a problem to run on Mark I required preparation of both coded orders for the machine and detailed instructions for its human operators.

A small number of orders did perform control functions, most impor-

a Charles Babbage, "On the Mathematical Powers of the Calculating Engine," 1837 manuscript reprinted in several collections.

b For example, in the Bell Labs case "the numerical results may be translated into special codes and perforated on standard teletype tape." (B.L. Sarahan, "The Relay Interpolator: A Description of its Operation," Naval Research Laboratory Report R-3177, Sept. 25, 1947, v.) Likewise, in Mark I "the perforations in the control tape corresponding to code 21 in the Out column." (Staff of the Harvard Computation Laboratory, *A Manual of Operation for the Automatic Sequence Controlled Calculator*. Harvard University Press, Cambridge, MA, 1946, 14).

tantly one that halted the machine automatically when a loop termination condition was satisfied. At this point, a machine operator would remove the looped control tape from the tape reader and replace it with whatever sequence came next. Changing tapes manually might seem inefficient, but the machine ran so slowly that operators could plausibly keep up with them. Loop execution times on Mark I were measured in minutes rather than microseconds.

The words "program" and "programming" were not originally applied to these machines. However, by 1944 the staff of the Harvard Computing Laboratory had recognized the work of "coding" problems into sequences of operation codes as a distinct task: "… the mathematician … chooses the numerical method … such functional, value and control tapes as are required are then computed, coded and punched. Since the mathematician cannot always be present while the calculator is running, instructions must be prepared to guide the operating staff."[c]

The system of tape-driven automatic control was later extended by providing instructions to shift control between more than one tape reader. For example, a computer with four readers might use them for code sequences corresponding to inner loop, outer loop, initial setup, and the printing of results. This system was stretched to breaking point, and beyond, with the completion in 1948 of IBM's Selective Sequence Electronic Calculator. As the word "Selective" suggests, the SSEC could automatically select which of several dozen paper-tape readers to take its next instruction from. SSEC staff had to grapple with tape rolls weighing 400 pounds, used to prepare data tapes looped at high speed past multiple read heads so that values could be looked up from coded tables. A custom lift was engineered to hoist these tapes, which were so wide that a special machine was built to glue their ends together.

ENIAC and the Automation of Control

The term "programming" comes, a little indirectly, from the project to build a much faster electronic computer at the University of Pennsylvania. The unprecedented speed of ENIAC, completed in 1945, forced its designers to come up with an entirely different control system. No paper tape could possibly read operation codes rapidly enough to keep its electronic circuits busy. Neither was it practical to expect operators to change tapes every few seconds as ENIAC completed a loop or subroutine and needed to move on to the next sequence of operations.

The words "program" and "programming" were not originally applied to these machines.

People often expect the history of technology to consist of a fairly direct series of advances by which primitive old machines gradually come to look and act ever more like modern ones. ENIAC is difficult to fit into this view of history. It was the first general-purpose electronic digital computer, being reconfigured to tackle entirely different kinds of problems from weather forecasting to prime number detection. Its control mechanism provided the full range of capabilities we associate with modern computers, including conditional branches and nested loops, but used an entirely different approach.

As we explain in our new book, *ENIAC in Action: Making and Remaking the Modern Computer*,[3] ENIAC consisted of dozens of distinct units, most built to carry out specialized computational functions such as multiplication, addition and number storage, loop control, or table look-up. When one unit finished a task it generated a "program pulse" to inform the unit responsible for the next operation that it was time for it to wake up and do something. What ENIAC did next was determined by two things. The first was its wiring, as the destination of the program pulse depended on where in ENIAC the other end of the wire carrying it had been plugged. The second factor was the switch settings on the receiving unit. ENIAC resembled a RadioShack boxed electronic kit, in that configuring it for a particular job involved wiring together the units needed to build a special-purpose machine.

The ENIAC team initially called the task of configuring ENIAC to carry out a particular problem "setting up" the machine. A particular configuration was called a "set-up" and documented in a diagram showing the wiring patterns and switch settings needed on each unit. This representation is quite different from a modern program, or even from the Harvard Mark I control tape. Indeed, the ENIAC approach to sequencing operations is much more difficult for modern audiences to grasp than the coded instructions used by the relay computers.

When computer scientists look back at the computers of the 1940s it is often to argue about which of them were "Turing complete." This depends in large part on their ability to implement a conditional branch, meaning the ability to select between two possible courses of action. Deciding whether to terminate a loop is seen as a special case of conditional branching, which is indeed how the instruction sets of later computers implemented looping.

Babbage, Lovelace, and the designers of ENIAC, however, modeled the top-level structure of computations in terms of loops rather than branches. Lovelace's mathematical notation expressed a computation in summary form as nested sequences of operations repeated a certain number of times. ENIAC's designers first considered building "about 30 units which are capable of receiving program pulses on one line and transmitting them on either of two lines in accordance with pulses received on another line." If one conceptualizes ENIAC's control wires as rails and its control pulses as trains, these switches would steer the pulses onto one or another track depending on the control signals received. This is perhaps why the term "branch" was later introduced to describe this control action, as pulses literally followed one or another branch through ENIAC's networks of control wires depending on the operations of its control circuits.

However, ENIAC's designers soon rejected simple binary switches, in favor of more complex "steppers"

c Ibid, p. 50.

each able to trigger up to six different sequences of operations. ENIAC's "master programmer" unit combined enough steppers and counters to count the iterations of each sequence and control up to 10 nested loops. Routing a control signal to a special input on each stepper would terminate a loop immediately, meaning that looping mechanisms also supported simple conditional branching. It was as if the two alternative statements in an "if … then" statement were treated as loops that would be iterated at most once. Inverting the later conception of looping as a special case of conditional branching, ENIAC made looping and loop termination the fundamental behavior.

Early in the ENIAC project, before design work had progressed very far, philosopher turned engineer Arthur Burks produced a table showing how ENIAC could compute an artillery trajectory, the task for which the machine had been commissioned. Although independently developed, the structure of Burks' table strongly resembled that produced a century earlier by Lovelace. In both tables, rows represented steps in the calculation, each storage unit was given its own column, and cells showed the content of each unit at each point in the calculation.[d] As we discussed in our September 2015 column, the tables are not in themselves programs, and are best viewed as traces or walkthroughs of the machine's operation.[e] Both tables indicated a need for nested loops, but when they were produced neither target machine had a well-defined mechanism for iteration. In a sense, the tables served as functional specifications for the machine designers: devise a mechanism to generate this sequence of operations and your machine will successfully complete this computation.

This striking convergent evolution, despite the very different architectures of the two machines, shows the analysis of a problem and its reduction to a series of arithmetic operations had very little to do with the specifics of the control system that would ultimately direct those operations. Indeed, the methods used to plan computations for automatic computers often incorporated those used with earlier technologies, whether in the application of punched card machines or desk calculators to large-scale mathematical work or the analysis of printed forms and clerical procedures in the office.

Earliest Discussion of "Programming"

Our reference to ENIAC's "master programmer" in the previous section alerts you to two things. The first is the word "program" became entangled with the control of automatic computers during the ENIAC project.[f] The second is it did not mean what you expect. By the 1950s "master programmer" would read as a slightly odd job title. In 1944 it was a pair of boxes stuffed with electronics to repeatedly trigger sequences of operations by generating control pulses. In fact the words "program" and "programming" cropped up in project documents to describe many different aspects of ENIAC's control system. As well as calling its control signals "program pulses," a June 1944 progress report described two accumulator units as being "automatically programmed to receive the multiplier and multiplicand" when a program pulse triggered the multiplier unit to which they were attached. This use of "program" fits with the notion, familiar to Babbage, that an automatic computer is built to carry out defined sequences of operations. Its control mechanism must trigger the performance of the right operations in the correct order. This is very similar to the meaning of "program" in other contexts—for example, the work of a radio programmer who selects and schedules programs for broadcast, the program for a series of concerts, or the program of study followed over time by a student. The use of "programmer" as the name for a simple mechanical control unit on a washing machine reflects a similar usage—turning the dial to a particular point triggers the performance of a particular sequence of washing operations (spin, rinse, wash, and so on). Echoing this, a primary meaning of "program" on the ENIAC was to describe a single operation set up on one of its units. What were being programmed were the operations of the internal circuitry of that unit.

By late 1945, however, the ENIAC team was beginning to talk of "programming" in something much closer to its modern meaning. This reflected the emergence of an entirely new way to think about automatic control.

EDVAC and the Modern Code Paradigm

"The First Draft of a Report on the EDVAC," composed in the spring of 1945 by mathematician John von Neumann and based on his work with members of the ENIAC team, never led to a second draft, still less a published article. It nevertheless laid out the basic architecture from which almost all subsequent computers have evolved. Computers patterned after the basic structure von Neumann proposed for the EDVAC, a successor to ENIAC being designed at Penn under a government contract, are often called "stored program" computers. We have previously criticized this term as vague and irredeemably overloaded with conflicting meanings, but those words do at least have the virtue of suggesting the attractiveness of EDVAC had something to do with its control system.[5]

EDVAC, as described by von Neumann, would drop ENIAC's special-purpose units and its elaborate system of distributed control. Like Babbage's Analytical Engine and the relay computers of the 1940s, EDVAC would read and decode orders one at a time, performing the operation specified by the code. The novelty was the code integrated control and arithmetic instructions in a single, aggressively minimalistic, set of orders. EDVAC did not need the hybrid control schemes of the relay machines or the special-purpose mechanisms and programming wires and switches of ENIAC.

We have previously identified the key aspects of the EDVAC approach to automatic control as:

- The program is executed completely automatically.
- The program is written as a single sequence of instructions, known as

---

d We have exploited this similarity to produce an ENIAC set-up that performs the Bernoulli calculation as specified by Lovelace. Run on an ENIAC simulator, it does indeed generate the sequence of Bernoulli numbers.

e Allen Bromley used the term "walkthrough" to describe tables like Lovelace's in "Charles Babbage's Analytical Engine, 1838," *Annals of the History of Computing 4*, 3 (1982), 215.

f Discussed in D.A. Grier, "The ENIAC, the verb "to program" and the emergence of digital computers." *IEEE Annals of the History of Computing 18*, 1 (Jan. 1996), 51–55.

"orders" in the First Draft, which are stored in numbered memory locations along with data. These instructions control all aspects of the machine's operations. The same mechanisms are used to read code and data.

- Each instruction within the program specifies one of a set of atomic operations made available to the programmer.
- The program's instructions are usually executed in a predetermined sequence.
- However, a program can instruct the computer to depart from this ordinary sequence and jump to a different point in the program.
- The address on which an instruction acts can change during the course of the program's execution.[5]

Von Neumann's design melded facilities for arithmetic and control. It contained both types of instruction, similarly formatted. His arithmetic circuits could be used to conditionally select numbers, while his storage circuits could change destinations for jump instructions as well as overwriting numeric data. This unification of control and arithmetic operations was more important than, and facilitated, EDVAC's more celebrated innovation of storing both instructions and data in the same writable and addressable memory. The first known program written in the EDVAC style was developed by von Neumann himself, and is now on display at the American Philosophical Society in Philadelphia.[6] The first to be run on an actual computer was executed directly from a read-only memory on ENIAC in April 1948, after its conversion to the new programming mode.[4] A few months later, at the University of Manchester, a program was loaded into an experimental writable memory and executed.

In the First Draft, von Neumann followed the Mark I terminology, giving an "order code" that defined EDVAC's instruction set. This usage was extended in an influential series of reports from the computing team he set up at the Institute for Advanced Studies in 1946 to construct his own EDVAC-like computer. These reports divided the process of problem preparation into two broad phases. "Planning" was described as "a mathematical stage of preparations," but "coding" encompassed drawing flow diagrams as well as writing instructions.[g]

At Penn, however, the meaning of the verb "to program" quickly shifted from describing the action of the control circuits responsible for triggering operations at the correct time to describing the work of the humans devising such sequences. In late 1945, a report described the practices used in "planning a set-up for the ENIAC" as "programming techniques,"[2] and a letter from one of the project's leaders noted "the EDVAC will contain a large number of units capable of remembering programming instructions," to be copied from tape "before the actual program is started."[h] "Programming" was by then roughly synonymous with von Neumann's "coding," and by early

g Goldstine, H.H. and von Neumann, J. *Planning and Coding Problems for an Electronic Computing Instrument, Part II, Volume 1* (Apr. 1, 1947, section 7.9). Drawing flow diagrams was described as the "dynamic or macroscopic" stage of coding, and writing instructions as the "static or microscopic" stage.

h H. Goldstine to H. Curry, Oct. 3, 1945, in the collection "ENIAC Patent Trial Collection" in the University of Pennsylvania archives.

1947 the noun "program" was becoming firmly established to refer to coded sequences of instructions.[i]

In the 1950s, "coding" acquired a more specific meaning as the most mechanical part of programming—tasks such as looking up numerical codes corresponding to particular instructions. "Coder" endured in some organizations as a job title for the most junior programmers. In the last decade or so it has been revived as an expression of geek pride, perhaps as a reaction against the trend toward increasingly abstract job titles for software developers such as "software engineer" or "solutions architect."

New Meanings

Space does not permit us to follow the further evolution of the concepts of program and programming in any detail, so instead we will flag a few key aspects of the subsequent history. The first is the distinction, sometimes made in the late 1940s and 1950s, between a "stored program" loaded into internal memory and an "external" program wired onto plug boards or read one instruction at a time from tape.[5]

In the early 1950s, "automatic programming" systems such as assemblers complicated the concept of program. The program actually executed by a computer, a string of numerical codes, became something that could be automatically generated from a different kind of input, commonly known as "pseudocode." This introduced two levels at which a program could be viewed, and the relationship between the levels was widely understood as one of translation.[8]

As the automatic programming systems became more complex, linguistic metaphors continued to gain currency. The FORTRAN system, released by IBM in 1957, translated mathematical expressions, data structure definitions, and control structures into executable programs. FORTRAN is remembered as the first widely used "high-level programming language." Donald Knuth and Luis Trabb Prado explored the many obscure and experimental systems that led up to this milestone, concluding that Konrad Zuse's Plankalkül, a proposal for which was published in 1948, was the first public description of the concept of a programming language.[7]

The increasing need through the 1950s to run programs on machines of different types led to a search for a "universal" programming language, culminating in the publication of the Algol proposals in 1958–1960. An Algol program had no association with a particular computer and, after *Communications* standardized on the language for its "Algorithms" department, was often intended primarily to be read by humans rather than executed by machines. Nowadays usage has widened to the point where the word "program" can refer equally to the "source code" written in a high-level language and the "object code" into which it is translated for execution on a particular machine.

Conclusion

Before the 1940s nobody talked about programming computers and no computers had what we consider to have been the original and fundamental meaning of programmability: the ability to automatically execute a specified series of operations. While this sense of programming could be applied to machines able to execute a series of coded arithmetic operations but not able to automate complex control structures, the fact is the earliest references to "programming" appear in the context of the first computer able to automatically execute nested loops and conditional branches: ENIAC.

We see ENIAC's control innovations as pragmatic engineering responses to the need for mechanisms that, unlike paper tape or human intervention, could keep up with its unprecedented electronic speed of operation. Its designers relied on problem-specific wiring to route networks of "programming pulses" around the machine. In the "First Draft" design for EDVAC, von Neumann extended the coding approach of the relay computers, designing a single instruction set that could express not only sequences of arithmetic operations but also the control structures pioneered on ENIAC. The EDVAC code unified arithmetic and control, programming a single set of logical circuits. It is in this context that people began calling the coded instructions a "program," a usage that evolved from related but distinct meanings of "program" and "programming" within the ENIAC project.

This provides a rather different view of the invention of computer programming, and its relationship to logic, from the widely held assumption that computer development in the 1940s was guided directly by the theoretical work of Alan Turing. In that view of history, a metaphysical attraction to the idea of "universality" inspired a competition amongst computer builders to be the first to check a box labeled "Turing complete." Von Neumann's design for EDVAC was elegant and its generalization and simplification of ENIAC's control capabilities unquestionably reflected his grounding in mathematical logic. The usefulness of a computer able to tackle many different kinds of calculations was certainly appreciated by the creators of the first automatic computing machines. The computer builders of the 1940s and 1950s adopted EDVAC's new design paradigms because they provided an efficient way to automate real machines, running real computations to solve real problems.

i For example, in S. Lubkin, "Proposed Programming for the EDVAC" (January 1947), in box eight of the collection "Moore School of Electrical Engineering Office of the Director Records, 1931–1948" in the University of Pennsylvania archives.

References

1. Bromley, A.G. Babbage's analytical engine plans 28 and 28a—The programmer's interface. *IEEE Annals of the History of Computing 22*, 4 (2000), 5–19.
2. Eckert, J.P., Mauchly, J.W., Goldstine, H.H., and Brainerd, J.G. Description of the ENIAC and Comments on Electronic Digital Machines. AMP Report 171.2R. Distributed by the Applied Mathematics Panel, National Defense Research Committee, (Nov. 30, 1945). Moore School of Electrical Engineering, Philadelphia, PA, 1945.
3. Haigh, T., Priestley, M., and Rope, C. *ENIAC in Action: Making and Remaking the Modern Computer.* The MIT Press, Cambridge, MA, 2016.
4. Haigh, T., Priestley, M., and Rope, C. Los Alamos bets on ENIAC: Nuclear Monte Carlo simulations, 1947-1948. *IEEE Annals of the History of Computing 36*, 3 (July–Sept. 2014), 42–63.
5. Haigh, T., Priestley, M., and Rope, C. Reconsidering the stored program concept. *IEEE Annals of the History of Computing 36*, 1 (Jan.–Mar. 2014), 4–17.
6. Knuth, D.E. Von Neumann's first computer program. *ACM Computing Surveys 2*, 4 (Dec. 1970), 247–260.
7. Knuth, D. and Prado, L.T. The early development of programming languages. In *A History of Computing in the Twentieth Century*, N. Metropolis, J. Howlett, and G.-C. Rota, Eds., Academic Press, New York, 1980, 197–273.
8. Nofre, D., Priestley, M., and Alberts, G. When technology became language. *Technology and Culture 55*, 1 (Jan. 2014), 40–75.

Thomas Haigh (thaigh@computer.org) is an associate professor of information studies at the University of Wisconsin, Milwaukee, and chair of the SIGCIS group for historians of computing.

Mark Priestley (m.priestley@gmail.com) is an independent researcher into the history and philosophy of computing.

DOI:10.1145/2774217

Gio Wiederhold

Viewpoint
Unbalanced Data Leads to Obsolete Economic Advice

Few computer scientists and technological workers worry about their role in the economies of their industries and their countries.

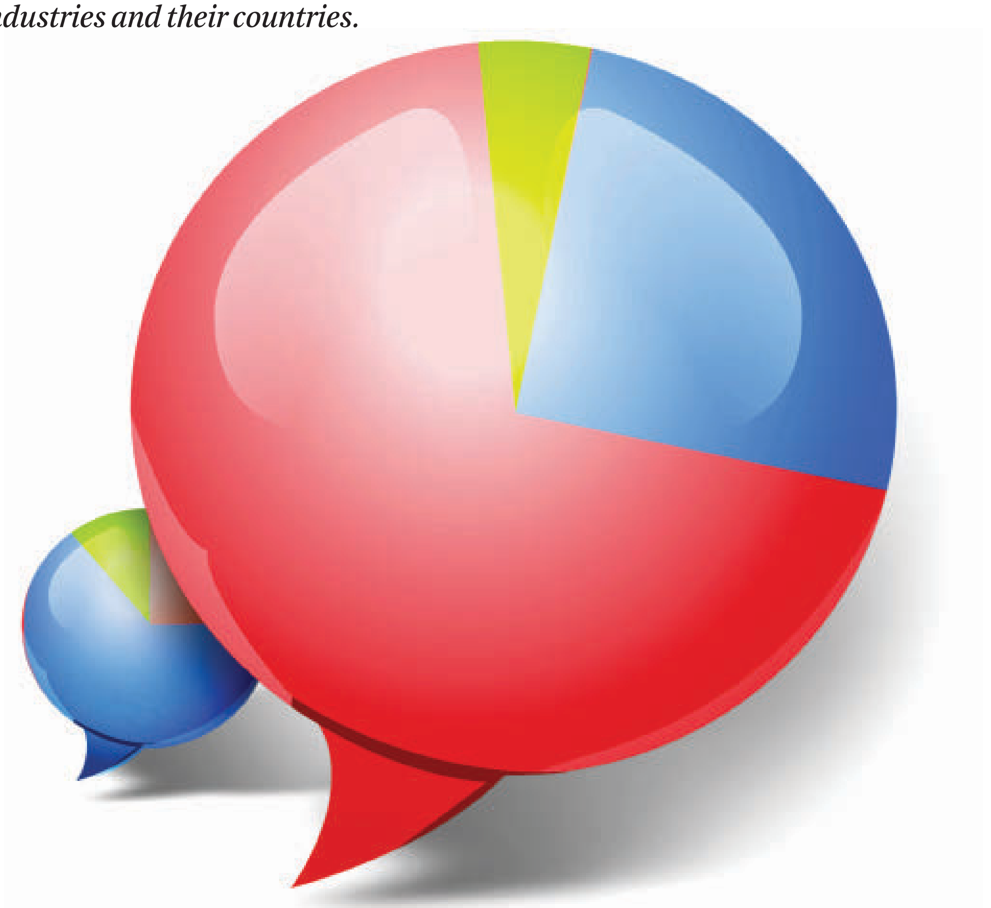


IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

THE HIGH TECHNOLOGY industry is recognized as being a major driver of the current economy. I am concerned about how poorly its role is understood in governmental decision making. Computer scientists do express concerns about disturbing trends.[9] An early study sponsored by ACM was based on opinions, rather than data.[1] Similar discussions address other high-technology engineering disciplines.[2] Analyses of relevant data should be the basis for decision making. Having "big data" raises high expectations.[7] But there are two related lacunae:

- Most analysts ignore the mechanisms that drive high technology and the intellectual origin of contents in the products being marketed.
- The data needed to measure the business interactions involving intellectual products within and among high technology enterprises is unavailable to economists.

In this Viewpoint, I expand on these two factors, which lead to misleading advice and imbalanced decision making.

Our leading economists have grown up and been educated in a time when financial capital and cheap labor were the crucial contributors to growth.[8] Building aircraft, automobiles, as well as the steel mills and machine shops that supplied them, all depended on much labor and substantial financial capital. These industries were associated with known locations, and their products were costly to ship. Geography was an important factor.

The world has changed. The post-industrial economy is based on intellectual capital, the experts, and the intellectual property (IP) they generate. The Apples, Microsofts, Googles, and the many smaller, hipper players that create an ever-larger fraction of the goods people purchase are not strapped for financial capital. The GEs, Intels, and similar enterprises that do require costly factories have moved much of the labor-intensive production of their tangible products overseas. The critical intangibles embedded in chips, phones, and computers are transmitted from their origin to far-away factories. Much research, development, testing, and prototyping, and the equally important market research and promotion activities remain in the U.S., complemented with laboratories in the EU and Asia.

Intangible products can be copied at negligible costs and shipped freely worldwide over the Internet. Containerized shipping has similarly reduced the costs of distributing the high-technology tangible products. It costs only approximately $0.50 per product to send a pallet stacked with iPads anywhere in the world. Computerized logistics minimizes inventory investments. Online payment systems allow revenues from worldwide sales to be collected anywhere, preferably in locations that do not insist on excessive reporting to their government agencies.

All these inputs to the modern economy need intellectual capital. But the prominent economists, those that have risen to the level of providing advice to governments, continue to focus on financial capital for their metrics and tools. For instance, keeping interest rates low helps primarily the traditional segments of industry, but does very little for high-technology enterprises.

However, one should not blame the economists. They depend on production and cost data derived from corporate financial results. They may also use income data from tax revenues. For tangibles and money such data is reported down to the pennies by accountants and presented in annual reports and aggregated for economists' think tanks. However, for high-tech enterprises operating globally, these "booked" values tend to be a fraction, about 20% on the average, of the market value investors assign to the corporations. Reported book values include the financial assets held outside the U.S. in tax havens as tangibles, held there for potential offshore investment by blocker statements, as being "subject to management's decision to indefinitely reinvest those earnings."[5] Keeping U.S. capital costs low discourages repatriation of those funds for investment in the U.S.[3]

Computer professionals are at the center of the storm that surrounds the industry.

However, investors in high-technology businesses value an enterprise according to future expectations, not by past and current costs. They count on future income due to the smart people and the IP they generate and exploit to make attractive products.[10] Predicting the future success of their results is always risky, but critical to understand high-technology enterprises. Avoiding the collection of suitable data because of risks and imprecision is not acceptable.

What data can be collected to drive future analyses? Amounts spent on research, development, maintenance, and marketing are available within businesses. Reporting it consistently can provide useful aggregations by industry. The maturity of an enterprise should be taken into account, since a Twitter is bound to present a different profile than a Microsoft. Venture capitalists do estimate the overall leverage of their investments and develop useful insights, but rarely share them beyond their peers. Prices of startup exits and merger prices reflect rational expert opinions. Stock market prices represent the wisdom of the investing crowd. While such data is not based on verifiable accounting data, in the aggregate it is as realistic as values for the tangibles listed on corporate books.

Economic analyses now cannot measure the value of the intellectual capital—the technological and marketing experts, the management, and IP—the factors that drive modern industry. Ignoring its contribution in decision making means the needed infrastructure—education, training, communication, as well as protection against external threats—is short-changed, since there is no documentable path of such investments to the outputs of modern industry. There are many anecdotes, and calls to allow more or less immigration of knowledge workers. But these are not placed into a broad coherent economic model.

Still, few computer scientists and technological workers worry about their role in the economies of their industries and their countries. They are willing to advocate for more education, ubiquitous Internet access, and job security. A complicating issue is that some experts advocate software should be free. That implies they expect to be supported by public funds or maybe by tax-deductible donations. Without support from professional experts little change can be expected.[6] Computer professionals are at the center of the storm that surrounds the industry. They should not just observe the effects, but try to provide data, analyses, and mechanisms so they will affect the world around them. Some modern economists will be pleased if more data becomes available.[4] **C**

References

1. Aspray, W., Mayadas, F., and Vardi, M.Y., Eds. *Globalization and Offshoring of Software; A Report of the ACM Job Migration Task Force*, ACM, 2006.
2. Charette, R.N. The STEM crisis is a myth. *IEEE Spectrum*, (Aug. 30, 2013).
3. Damodaran, A. Unlike the U.S. tax code, Apple is perfectly rational. *Financial Times* (May 7, 2013).
4. Damodaran, A. The Aging of the Tech Sector: The Pricing Divergence of Young and Old Tech Companies. Musings on Markets, http://aswathdamodaran.blogspot.com/
5. Fleischer, V. Overseas cash and the tax games multinationals play. *New York Times* (Oct. 3, 2012).
6. Gibbs, R. Leveling the Playing Field: Curbing Tax Havens and Removing Tax Incentives For Shifting Jobs Overseas. The White House (May 4, 2009).
7. Jagadish, H.V. et al. Big data and its technical challenges. *Commun. ACM 57*, 7 (July 2014), 86–94.
8. Nasar, S. *Grand Pursuit: The Story of Economic Genius.* Simon & Schuster, 2011.
9. Vardi, M. Is information technology destroying the middle class? *Commun. ACM 58*, 2 (Feb. 2015), 5.
10. Wiederhold, G. *Valuing Intellectual Capital, Multinationals and Taxhavens.* Management for Professionals, Springer Verlag, New York, Aug. 2013.

Gio Wiederhold (gio@cs.stanford.edu) is Professor (Emeritus) of Computer Science, Medicine, and Electrical Engineering at Stanford University, Stanford, CA.

Supplemental material for this Viewpoint is available at the *Communications* website; see http://cacm.acm.org

DOI:10.1145/2803170

Yoav Shoham

Viewpoint
Why Knowledge Representation Matters

A personal story: From philosophy to software.

THERE IS A big difference between the attention artificial intelligence (AI) is currently receiving and that of the 1990s. Twenty years ago, the focus was on logic-based AI, usually under the heading of knowledge representation, or KR, whereas today's focus is on machine learning and statistical algorithms. This shift has served AI well, since machine learning and stats provide effective algorithmic solutions to certain kinds of problems (such as image recognition), in a way that KR never did. However, I contend the pendulum has swung too far, and something valuable has been lost.

Knowledge representation is not a single thing. While I think an argument could be made about KR as a whole, I will be focusing on the "applied philosophy" aspect of it—the logical representation of commonsense notions, with an emphasis on clear semantical underpinnings.

I will make the case for the most part through a personal story. The story starts with a paper I published in 2009 in the *Journal of Philosophical Logic*, continues with a research project at Stanford and Duke, later with a company called Timeful, and concludes with Timeful being acquired by Google in 2015. The point of the story is there is a direct link between the original journal paper and the ultimate success of the company.

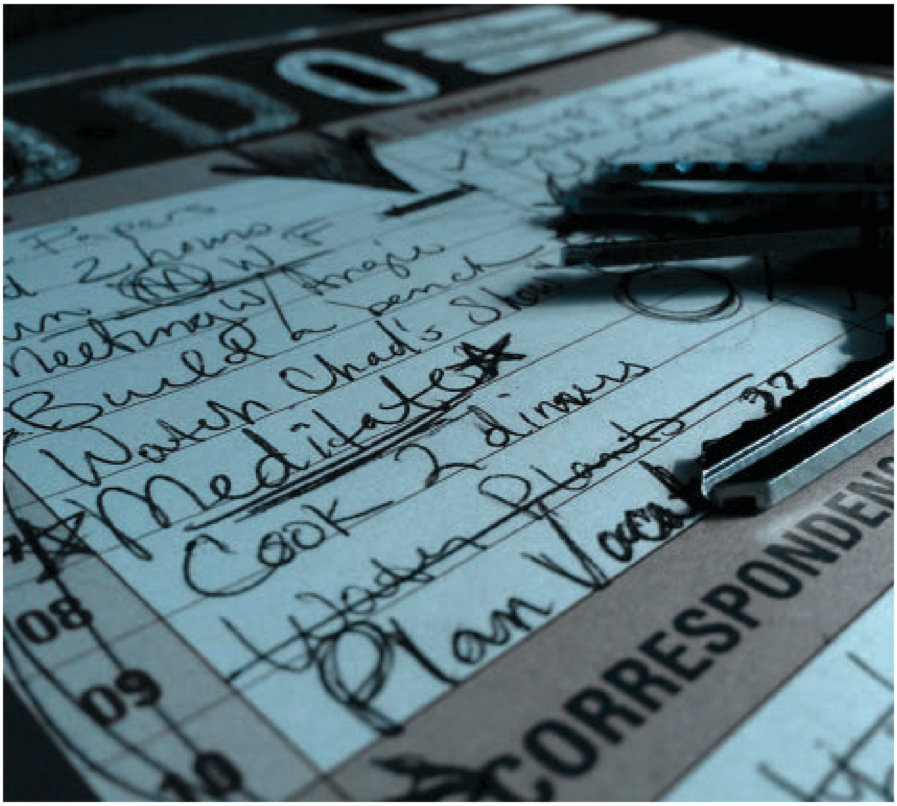
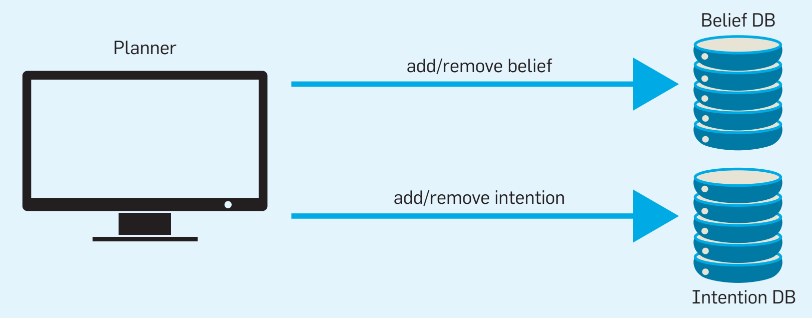


IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

The journal paper was "Logics of Intention and the Database Perspective."[6] This paper followed an important though thin strand of papers in AI on the logic of intention, spawned by Cohen and Levesque's seminal "Intention is Choice + Commitment."[4] This literature in turn was inspired by the less formal literature in philosophy on rational agency, such as Bratman's "Intentions, Plans, and Practical Reason."[3] My own paper took inspiration from the Cohen and Levesque paper but questioned its foundations, and proposed an alternative approach. Although my approach was computationally motivated (as indicated by the title),

Proposed database perspective.



the arguments were theoretical and philosophical in nature.

Following that journal paper I sought some funding to continue the research, as professors tend to do. And as funders tend to do, my would-be funder requested that I include some potential applications of this work. Then several things occurred to me. The first was I deal with intentions all the time—in my personal calendar. The second was these were intentions of a very specific kind—rigid events and meetings. And the third was that my personal calendar was not all that different from that of my late grandfather, which is odd given how the demands on people's time have changed, and how technology has advanced. This led to an obvious question: What would happen if I enhanced the calendar with richer and more flexible intention types, and the calendar had the intelligence to help deal with the resulting complexity?

To understand this point better, it is worth discussing the ideas in the journal paper a bit further. The proposed database perspective is encapsulated in the accompanying figure, which can be thought of a generalization of the AGM scheme for belief revision,[2] the latter being restricted to the "belief" part of the picture. In the AGM framework, the intelligent database is responsible not only for storing the planner's beliefs, but also ensuring their consistency. In the enriched framework there are two databases, one for beliefs and one for intentions, which are responsible for maintaining not only their individual consistency but also their mutual consistency. In the journal paper I laid out the main consistency conditions, and in a subsequent paper with Icard and Pacuit[5] we gave a logical formalization of it, which is a conservative extension of the AGM framework. It is not appropriate in this Viewpoint to go into more technical details, and indeed many of them are not relevant here. What is important to take away is the view of an intention database that performs intelligent functions on the part of the agent.

Returning to the storyline, the funder was persuaded, and we started a small project to explore these ideas. The next two years were fun but there is not much to say about them that is relevant to the story here, except: the project was soon led by a new Ph.D. student, Jacob Bank; it was also joined by my longtime friend and colleague Dan Ariely, a renowned behavioral economist; and by the beginning of 2013 we decided to start a company, which eventually came to be called Timeful. We were not so much driven by the specifics of our joint research up to that point, as by the realization of how acute the problem of time management was in society, and how ill suited current tools were to deal with it.

When Timeful 1.0 came out in July 2014, the reaction from both users and press was very favorable. Some 2,000 user email messages poured in during the first month, many of them emotional. Timeful had clearly struck a nerve, even if the product still had a way to go. Very soon the company attracted interest from major players, leading to the eventual acquisition by Google. None of this would have happened were it not for KR; here is why.

Intention Objects as the Basic Data Model

Timeful developed the concept of the Personal Time Assistant (PTA), whose role it is to help manage time, the resource that is both the scarcest and the most difficult to manage. The approach rested on three main pillars. The first was allowing the user to naturally represent in the system everything that vied for their time. The second was the application of machine learning and other algorithms to what is inherently a hard optimization problem. The third pillar was behavioral science, which meant crafting an environment that subtly helps correct for natural time-management mistakes we all make (such as procrastinating, and overestimating our future availability). Of these, it is the first pillar I want to focus on; it was the most fundamental of the three, and the one based directly on KR.

Consider all the things that vie for our time: meetings, events, errands, projects, hobbies, family, health maintenance, sports, or just time to think and recharge. They are all superficially different, and historically reside in different applications (meetings and events in the calendar, errands in a to-do list, projects in a project-management system) or simply stay in our head. But they all vie for the same resource—time—and if you are to make intelligent trade-offs, they ought to reside in the same place. And indeed, they are all intentions, albeit with different properties. Following the vision of the intelligent intention database, the first fundamental decision was to develop a data model rich enough to encompass all these intention types. The result was a data model called the intention object (IO). An IO is a feature vector that includes a textual description, temporal attributes (when it can be executed, when it should be, its duration—all specifiable at various degrees of precision), conditions for executing the intention (such as location, or tools needed), and other attribute types.

Intention objects became the foundation for the system, and everything—including the algorithmic scheduling and the behavioral nudges—hinged on them. Of course, the user was not presented with a feature vector, but rather with several pre-packaged classes of intentions. As of April 2015, there were four classes: Events (such as meetings); tasks (such as making a phone call); habits (such as jogging three times a week); and projects (such as writing a long report). But under the hood, for the system they all broke down to feature vectors.

More Product Decisions

Knowledge representation as not only the original impetus for Timeful and the inspiration for its data model. The team repeatedly found itself seeking guidance from the philosophical literature when making specific product decisions. It is difficult to fully convey this, but here are two concrete examples.

The first example has to do with the modest checkmark. Every to-do list allows you to check off tasks accomplished. Timeful had this feature too, but it bothered us that tasks had checkmarks and events did not, even though they were both IOs. It was not so much the aesthetic asymmetry, but more the underlying principle, and how that principle should be applied to other IOs, such as habits and projects. Then we went back to our roots and realized it had to do with tracking one's commitments. If there is one principle the philosophical literature agrees on it is that intention involves commitment (as reflected in the very title of the Cohen and Levesque article). When I intend to do something, it is not that I merely make a note of it; I am committed to tracking it and making it happen. When seen in this light, we realized events do not require tracking; a meeting is accomplished by being scheduled (there are exceptions, such as when the meeting has a goal that may not be achieved, but those were handled by specifying a separate task associated with the meeting). All other intention types require explicit monitoring, and so we ended up attaching checkmarks to all IOs except events.

The second example has to do with the temporal scope of an intention. Most to-do systems are "lists of shame"—things you write down but never do. We wanted to avoid that, and did it via strict time scoping. This early decision traced back to a mini-debate in the literature. In the Cohen and Levesque formalism, statements such as "I intend to read this book" are the basic concept. But in my journal paper I argued this is problematic, and it goes back to the issue of commitment. If I am committed to an intention that is not anchored in time, what exactly am I committing to, and how does it actually drive action? (If you have a teenager at home you know what I mean.) Instead, I argued, the basic construct should be statements such as "I intend to read the book from 2 P.M. to 4 P.M. on Saturday." You can then relax those by existential quantification, and say things such as "I intend to read the book for two or three hours sometime this weekend." But you are always explicit about the time scope. Timeful adopted this philosophy; the implicit contract with the user was that she should be serious about her intentions, and in return the system would help her accomplish those by placing them on her calendar and prodding her to get them done (the tagline when the app launched was "get it scheduled, get it done"). Thus every task required either a specific "do on" or "do by" date. The task then appeared on the time grid, alongside the events. (In the case of "do by," the system selected a time before the due date, which the user then could change if needed. Indeed, if an event later displaced the task, the system would move the task automatically.) The same logic applied to habits and projects, in more involved ways.

When you build a product you want it to be beautiful on the inside.

Conclusion

The story of Timeful is a happy one, and much of the credit goes to KR. Could one have arrived at the same insights without KR or philosophy? Possibly, but the fact is no one did, and I do not think that is an accident. When you build a product you want it to be beautiful on the inside. What I mean by this is that often, when you set out to design a great user experience, you either do not have the conceptual vocabulary with which to do it well, or, worse yet, you are fighting an existing conceptual framework and data model. And if the internal structure is not right, you will never have a truly beautiful user experience. Philosophy and KR encourage you to think rigorously about your conceptual architecture, and provide guidance when designing specific features.

This does not lessen the importance of machine learning or statistics. But machine learning requires a feature space, and stats require an event space. Even the most avid deep learning aficionado will not argue those will always arise ex machina, unaided by human insight (unless you work for Google, and are only interested in cats[a]).

Does this mean every philosophical conundrum and logical puzzle has a direct practical implication? Of course not. But if you are designing a car you do need wheels, so you might as well not reinvent them, especially if yours would end up not quite round.

There are reasons to be optimistic. There are signs researchers are becoming increasingly leery of the "machine learning and stats will solve everything" viewpoint, and are seeking to integrate the (fantastic) achievements of machine learning into a broader AI approach. For example, a recent AAAI symposium[1] brought together leading researchers from knowledge representation, machine learning, linguistics and neuroscience to discuss interactions among these areas. My sense is the pendulum is beginning to swing back ever so slightly, and that if we as a community encourage the trend, AI will be better for it. **C**

a Here, I am counting on the sense of humor of my Google colleagues.

References

1. AAAI Spring Symposium on KRR: Integrating Symbolic and Neural Approaches, Stanford University (Mar. 2015); https://sites.google.com/site/krr2015/
2. Alchourròn, C.E., Gärdenfors, P., and Makinson, D. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic 50* (1995), 510–530.
3. Bratman, M. *Intentions, Plans, and Practical Reason.* Harvard University Press, 1987.
4. Cohen, P. and Levesque, H. Intention is choice + commitment. *J. Artificial Intelligence 42* (1990), 213–261.
5. Icard, T., Pacuit, E., and Shoham, Y. Joint revision of belief and intention. In *Joint Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning* (KR), 2010.
6. Shoham, Y. Logics of intention and the database perspective. *J. Philosophical Logic 38* (2009), 633–647.

Yoav Shoham (shoham@stanford.edu) is Principal Scientist at Google and Professor (emeritus) at Stanford University.

DOI:10.1145/2814336

Article development led by acmqueue
queue.acm.org

BY GEORGE V. NEVILLE-NEIL

Time is an Illusion Lunchtime doubly so.

—Ford Prefect to Arthur Dent in *The Hitchhiker's Guide to the Galaxy*

ONE OF THE MORE surprising things about digital systems—and, in particular, modern computers—is how poorly they keep time. When most programs ran on a single system this was not a significant issue for the majority of software developers, but once software moved into the distributed-systems realm this inaccuracy became a significant challenge. Few programmers have read the most important article in this area, Leslie Lamport's 1978 *Communications* article "Time, Clocks, and the Ordering of Events in a Distributed System,"[2] and only a few more have come to appreciate the problems they face once they move into the world of distributed systems.

Any discussion of time should center around two different measurements: synchronization and syntonization. Synchronization, loosely defined, is how close two different clocks are to each other at any particular instant. If two clocks, or computers, claim it is 15:30 at exactly the same moment, then they are considered to be in sync. The definition of "exactly

IMAGE BY STOKKETE

the same moment" is where the first difficulty arises. Do you care about the same minute (15:30), the same second (15:30:00), millisecond, microsecond, nanosecond? The level of accuracy you wish to achieve defines, approximately, the level of difficulty in attaining it.

Syntonization is the quality of the timekeeping of an individual clock. One might assume a second is always a second, but in reality there needs to be a physical system from which to work and from which to derive time. Computers use quartz crystals as the basis of all of their internal operations. The 3GHz CPU in your laptop or server is able to run at that speed because somewhere on the motherboard is a bit of quartz that, when electricity is applied to it, vibrates at a known frequency. That frequency is then multiplied and divided to drive the various parts of the rest of the system. If crystals—and other pieces of hardware—were unaffected by their environment, then all computers would have perfect syntonization. Alas, this is one place where the real world intervenes. Crystals experience changes in heat, power, and age, and no two of them are exactly alike, but they vibrate within a tested range provided on the data sheet from the manufacturer. When a server heats up—for example, because it is running a complex workload—then the crystal will oscillate faster, and the computer's internal clock will run ahead, or skew, faster. The effect of skewing on a typical server is shown in the accompanying figure. If the air gets colder, then the crystal will oscillate slower and seconds will last longer, skewing time on the server negatively.

The level of syntonization that can be achieved with a particular crystal is also an economic problem. Cheaper crystals are less stable (that is, more prone to skew) than more expensive ones. All commodity hardware—from the most expensive server to the cheapest tablet—has a very inexpensive, low-quality crystal, which costs about 10 cents. A typical laptop or server, left without any type of external time conditioning, will drift out of sync within minutes and after a few hours may already be several minutes away from good synchronization with other systems.

To correct for poor syntonization the operating system has a set of routines to steer the clock. If the system is running too fast, then the routines will tell it to slow down; if running too slow, then it will tell it to speed up. These steering inputs must be applied gently, over a period of time, to avoid pushing the system into hysteresis, a condition whereby the clock oscillates wildly around proper syntonization but is never able actually to achieve stability.

To provide proper steering to the clock, the system needs some idea of what is too fast or too slow, and for this it must have an external source of correct, or at least better, time.

Most developers may logically think the solution is to buy better machines, but to this day you cannot buy a commercial server or laptop where the built-in crystal is any better than the cheapest ones on the market. Perhaps in time, as more companies build distributed systems, this might change, but today it is not a practical option. To achieve better syntonization you can purchase a high-quality, stable crystal on a PCI (peripheral component interconnect) add-on board. The system clock is then conditioned, across the PCI bus, from this card. At about $1,000 per card, however, this solution is practical for only a small number of servers, or for those with unlimited budgets.

Another way to condition the clock on a server is to have an external crystal oscillator distributed over a serial bus. Injecting a high-quality 10MHz signal into a server via a serial line is another way to give a set of machines good syntonization. Distributing such a signal is reasonable for a small number of machines (up to 48) but breaks down as the number of connections and length of the cables increase. Wiring an entire datacenter for such a signal is, again, prohibitively expensive and prone to its own set of problems.

I will return to the question of how to synchronize the system to an external time source later, but first I must address the question of how a program finds out what time it is on the system on which it is currently executing.

What Time Is It Now?

For the majority of software developers the concept of "now" is represented by a single system call, `gettimeofday()`, which returns to the caller the system's idea of what time it is at the moment of the call. Whether this is useful information or not depends on the quality of the time measurement required. If the system needs to be accurate to within only a second, then the `gettimeofday()` routine is adequate—and it is often good down to the millisecond, although this can vary with the quality of the timekeeping software in the operating system. The trade-off that must be made when deciding how to get the time in a program is between speed and accuracy. Asking for the time requires quite a bit of work on the part of the operating-system software to give an accurate representation back to the user program. While `gettimeofday()` will work on almost any system, the `clock_gettime()` and `clock_getres()` routines give the programmer more flexibility in working with time.

Systems with the `clock_*` routines expose several types of clocks to user programs. For example, on the FreeBSD operating system, the clocks are `CLOCK_REALTIME`, `CLOCK_MONOTONIC` `CLOCK_UPTIME`, `CLOCK_VIRTUAL`, and `CLOCK_PROF`. `CLOCK_REALTIME`, which reports time the same way a wall clock does, is the most common source of time for systems that generate log files, as these log files will be read by human users who want to know what time an event occurred. The other clock routines are more specialized and are documented in the system manual pages.

Each clock has two variants: `PRECISE`, which gets the most precise time possible but takes the longest to return to the caller; and `FAST`, which is the least accurate but the quickest to give the answer. These two variants describe one of the tensions in computer-based timekeeping quite well: the more precise you want the time to be, the longer it will take to retrieve it. Asking low-level hardware for the current time will normally return the most precise answer, but it will also take time because a protection boundary must be crossed and the machinery required to return an accurate time has its own nonzero overhead.

The `clock_gettime` and associated routines also exist on Linux but do not appear in Mac OS X release of Yosemite. Microsoft's Windows operating system has similar functionality to `gettimeofday()` but under a completely different name, `GetLocalTime()`.

Measuring a brief interval—for example, the time required for a few lines of C or C++ code to execute—can often be done with less overhead by using on-chip instructions. Modern Intel CPUs have a single instruction, `rdtsc`, which returns the on-chip Time Stamp Counter. When a program wishes to measure the time between two local, non-network events, this is the fastest and cheapest way to work with local time.

It is important to choose the right clock option for a particular application. If an interval time needs to be measured, then `rdtsc`, or a library wrapped around it, is the best solution, whereas getting the system time for use in log files probably ought to be carried out using `clock_gettime()` with a `FAST` option; or, if `clock_gettime()` is not available, then `gettimeofday()`.

Find a Better Clock

Now that we know how to get programs to ask the system properly for the time, and to measure local intervals, we can return to the problem of synchronizing systems with some form of external reference. This challenge is most often addressed by finding a better clock, periodically asking it for the time, and adjusting the system's local clock to match the external reference. One solution to finding a better source of time is to use the network, since all systems in a distributed system are, by definition, connected to the network. In the 1980s a group of researchers defined the Network Time Protocol (NTP).[3] First documented in 1985 and subsequently updated in 1988, 1992, and 2010, NTP is the most successful Internet-based network time protocol in use.

Using UDP (Unreliable Datagram Protocol), NTP is implemented as a distributed system that has higher- and lower-quality clocks forming a hierarchy in which those clocks with a lower *stratum* number are of higher quality. A stratum 0 clock is considered a *reference clock* and contains a highly stable crystal, such as one based on cesium or rubidium, or it may be synchronized with GPS, which transmits a stable and reliable time signal. Systems at stratum 1 are connected, often via a direct cable, to a stratum 0 clock; stratum 2 clocks are connected to stratum 1; and so on, with this chain of time reference extending to stratum 15. There is no defined standard for the time quality of a clock at any particular stratum, only that a lower number is expected to be of a higher quality. Not having a hard definition of quality at each layer has meant improvements in technology, whereby more accurate clocks could be deployed at the lower (better) strata, did not require the renumbering of current systems. As stratum 0 and 1 clocks improved, so did the time available at stratum 2 and above.

NTP uses a polling mechanism to query the time on a set of clocks that are used as input to an algorithm, which is then used to discipline the local clock. Most operating systems have some version of the `adjtime()` system call that allows an external program such as NTPd (Network Time Protocol daemon) either to speed up or to slow down the system clock as it deems necessary. A globally distributed protocol such as NTP must act as a good network citizen, avoiding flooding the network with requests. In part because NTP was specified early in the history of the Internet, when long, slow, and unreliable links were the norm, its default polling interval is once every 64 seconds. Because of this long default polling interval, a new clock can take tens of minutes or more to synchronize with external sources; and once some level of stabilization has been reached, it is still possible for the clock to wander as new inputs arrive only every minute or so.

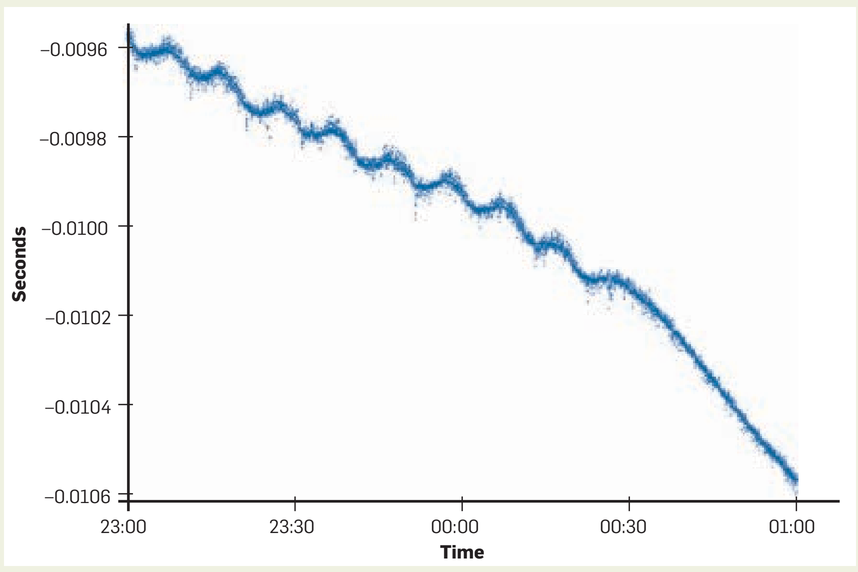
Increasing the polling frequency to the maximum of once per 16 seconds can increase the accuracy of the clock and the expense of added network load, but it is still possible to have time skewing by many milliseconds, depending on the environment and the quality of the crystal. As discussed previously, the low polling frequency of NTP usually prevents systems from achieving submillisecond synchronization.

Datacenter Time

The proliferation of datacenter-scale distributed systems has taken a seemingly obscure protocol, first defined by the IEEE in 2002, and brought it to the attention of a broader audience. Originally designed to synchronize systems in factory automation, electrical power generation, and cellular networks, the Precision Time Protocol (PTP)[1] is now used in financial services—in particular, HFT (high-frequency trading)—as well as other datacenter systems where submicrosecond synchronization is necessary.

The need for submillisecond synchronization in factory automation, electrical generation, and cellular networks is apparent when looking at the time scales involved. A robot arm moving at 30 miles per hour moves 44 feet in one second, a speed at which a collision with a human operator or a neighboring robot would be catastrophic. Electrical power networks running at 60Hz AC need to have all of their shared components synchronized; otherwise, handing off energy from one block to the next could result in a catastrophic, and fiery, failure. The alternations

PTP log graph.



in the current of two adjacent power grids need to be synchronized to well under 10 microseconds, with a typical rating at 4.6 microseconds. In cellular networks the synchronization requirements were once only necessary to handle the smooth handoff of a mobile device between towers, but now that high-speed data services are being provided via multiple base stations, the synchronization requirements among the base stations themselves are +/− 1.5 microseconds. It was for these applications that IEEE originally designed PTP.

The first decade of the 21st century saw the rise of HFT, which brought together algorithms and high-performance computing in an effort to extract large amounts of value from millions of very small financial transactions. HFT companies operate racks of machines that all look at market data in real time and then make trading decisions based on differences between the perceived value of stocks and other financial instruments. The HFT world is dominated by a need for speed, but it is not possible to do all of the work on a single large computer. To achieve scale, racks of servers are reading the same data simultaneously and making trading decisions in real time. These trading decisions require knowing the order of events—which stock moved which way before another stock. Did Mobil go up before Shell went down, or was it the other way around?

The time scales on which these financial transactions take place move first to milliseconds, then tens of microseconds, and then into the range of one microsecond—and there are applications where an even smaller range would be useful. The HFT market not only was able to apply PTP to its problems, but also its collective financial meant PTP features began to appear in expensive—though still commodity—hardware. Many network interface card and network switches now directly support the timestamping of PTP packets or include a version of the PTP daemon as an option.

PTP differs from NTP in several key ways. Having been designed for a small set of hosts, such as a factory floor or a set of cell towers, PTP is meant for use on a single network—in fact, most documentation and papers about PTP assume there are no switch or router hops between the various systems. PTP can be routed over a network, but the loss in time quality is sufficient to dissuade this use in most environments. Someone who wants a routable, Internet-based time protocol would be better off using NTP.

Syntonization is the quality of the timekeeping of an individual clock. One might assume a second is always a second, but in reality there needs to be a physical system from which to work and from which to derive time.

PTP is designed as a multicast protocol in which there is a *grandmaster*, PTP's name for the better clock, and a number of slaves that are following the grandmaster. The grandmaster sends out a `SYNC` packet once per second with the current time. The slaves periodically send a multicast `DELAY_REQUEST` packet, to which the grandmaster replies with a `DELAY_RESPONSE`. The `DELAY_REQUEST/DELAY_RESPONSE` pair allows the slave to determine the one-way delay between itself and the grandmaster. Without this measurement the slave has no way of knowing when the `SYNC` packet actually originated at the grandmaster. Taken together, these packets allow the slave to update its own clock based on the time of a better clock (the grandmaster) and to do so in the face of some amount of *jitter*, the small variations in the transit time of a packet between master and slave systems.

PTP assumes a symmetric network, where packets take the same amount of time to propagate from the grandmaster to the slave as they do from the slave to the grandmaster. If the times are asymmetric, then the offset calculated by the client may be skewed by some number of microseconds. The open source PTP daemon, PTPd, can be configured to take this asymmetry into account, but if network conditions change—for example, because one network switch was replaced with a new and different model—then the configuration will have to be changed as well. Most deployments of PTP do not use any fixed offset and simply assume the network propagation times are symmetric.

Network time protocols are particularly sensitive to jitter. If packets always required a constant amount of time to get from the master to the slave, then a simple offset could be programmed into the slave and it could use this offset, along with the time in the `SYNC` packets, to steer its clock. A simple test with the ping program will show the time for packets to transit a single network segment is subject to quite a bit of jitter that will prevent a clock from being synchronized to within 10 microseconds—and certainly to within one microsecond.

Jitter is an important consideration when PTP is being used to achieve good intersystem synchronization. Sources of jitter include network equipment, including interfaces, cards, and switches, as well as software, such as the device driver and operating system. Network switches induce jitter when competing users of the network get in the way of network timing packets. All of the literature about PTP and all of the benchmarks from hardware providers describe conditions on otherwise unused networks. Running the timing protocol on its own network is preferable, as it reduces the jitter induced by the network itself.

The current largest contributor to jitter in many deployments of PTP is in the software. When we took over the PTPd project (http://ptpd.sf.net), an open source implementation of PTP, all timestamps were recorded in user space using `gettimeofday()` after a packet had gone through several layers of software, including the device driver, and the operating system's network stack. Each layer of software induces some level of jitter, as these layers are driven by timers, and packets show up asynchronously. Our first improvement was to move the timestamping into the kernel, recording timestamps using the `SO_TIMESTAMP` socket option. While this improved synchronization quality, it was not enough, and therefore timestamping was moved to just above the network driver code in the Berkeley Packet Filter (BPF). This resulted in achieving synchronization within 10 microseconds on a system that was not otherwise busy. The proliferation of PTP within the HFT world, where 10Gb Ethernet was also prevalent, led several network card providers to add stable oscillators and timestamping hardware to their devices. At least one such vendor uses PTPd, adapted to the vendor's hardware, to achieve synchronization to within 500 nanoseconds of a grandmaster on an otherwise quiet network.

A lesser-known contributor to noise in the measurements comes from the way in which modern NICs (network interface controllers) are built. To achieve high bandwidth in a network interface it is important to batch many packets together before they are all handed, in a group, to the operating system for processing. If each packet on a 10Gb Ethernet network generated an interrupt when it arrived, that would result in 14.8 million interrupts per second, which is wasteful when moving data in bulk. Any NIC that supports speeds of 1Gbps or higher is going to have some form of AIM (adaptive interrupt moderation), in which interrupts are gathered up without the device taking any action until a certain number has arrived. Having an option such as AIM left on introduces a sinusoidal wave, whose frequency is the interrupt moderation rate (in one instance this was 8KHz), into the measurements, preventing the system from achieving a good level of synchronization.

One solution to delivering accurate time to servers, while also allowing them access to high-bandwidth connections, is to use the server's onboard 1Gbps network port for the time service, and have a separate 1Gbps or 10Gbps network port for data. Providing two network connections for each host is typical in many datacenters, as there is often an administrative network as well as a data network leading to each host. If the time data is piggybacked onto the administrative network, then the PTP packets should be placed in their own virtual LAN with a higher quality of service than the administrative data. Care also needs to be taken to avoid large data copies, such as those used to retrieve server log files, because this will introduce jitter to the time traffic.

A common practice for synchronizing time in a datacenter is to make the grandmaster a GPS-based clock with a stable oscillator. While such a clock is expensive, on the order of a few thousand dollars, each datacenter may need only one, or perhaps two for failover, which is far cheaper than placing a stable oscillator card into each server. A good grandmaster will do all of its timestamping of packets just above the physical layer, much as expensive NICs do, in order to prevent any software-based jitter from entering into the measurement.

Marching On

Submicrosecond synchronization among a large group of distributed hosts is a significant achievement, but even higher levels of synchronization are going to be necessary for future applications. The proliferation of 10Gbps Ethernet in datacenters and the prospect of 25Gbps, 40Gbps, and 100Gbps networking will require synchronization levels in the nanoseconds in order to determine whether the arrival of a packet at host A occurred before the arrival of another packet at host B. On a 10Gbps network, packets can arrive only 67 nanoseconds apart; at higher speeds this number gets considerably smaller.

While the first datacenters to apply PTP broadly are in the financial sector, many other applications are now realizing the need for higher qualities of synchronization than can be achieved with NTP. Any endeavor with a system that asks the question, "Who did what to whom and when?" and needs an answer that is correct to within less than a millisecond, will want to find a way to integrate PTP. In accordance with the IEEE requirement that all protocols be reviewed and renewed every five years, the IEEE-1588 working group is currently discussing new features that may be present in version 3 of the protocol. Once the new standard has been approved, it will remain to be seen whether the hardware and software vendors follow.

Related articles on queue.acm.org

Principles of Robust Timing over the Internet
Julien Ridoux and Darryl Veitch
http://queue.acm.org/detail.cfm?id=1773943

Passively Measuring TCP Round-trip Times
Stephen D. Strowes
http://queue.acm.org/detail.cfm?id=2539132

Securing the Network Time Protocol
Harlan Stenn
http://queue.acm.org/detail.cfm?id=2717320

References

1. IEEE Standards Association. 1588-2008—IEEE standard for a precision clock synchronization protocol for networked measurement and control systems; https://standards.ieee.org/findstds/standard/1588-2008.html.
2. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM 21*, 7 (July 1978), 558–565.
3. Mills, D.L. Network Time Protocol. RFC 958. Internet Engineering Task Force (1985); https://tools.ietf.org/html/rfc958.

George V. Neville-Neil works on networking and operating-system code. He also teaches courses on various subjects related to programming. His areas of interest are code spelunking, operating systems, and rewriting bad code.

practice

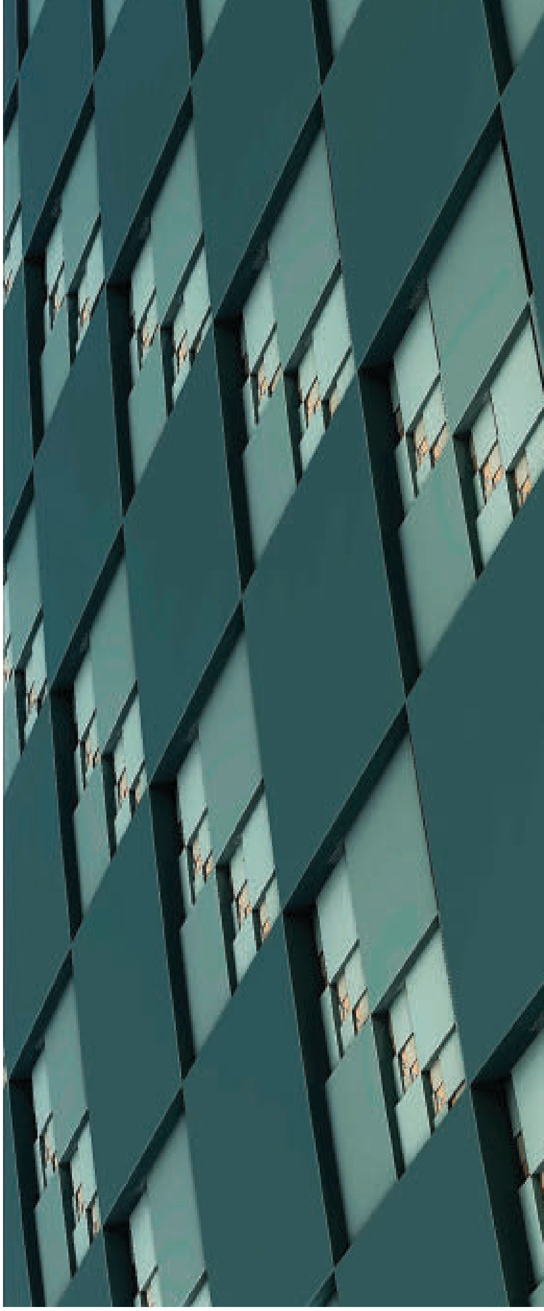
DOI:10.1145/2814342

Article development led by acmqueue
queue.acm.org

Implications of the datacenter's shifting center.

BY MIHIR NANAVATI, MALTE SCHWARZKOPF, JAKE WIRES, AND ANDREW WARFIELD

Non-Volatile Storage



FOR THE ENTIRE careers of most practicing computer scientists, a fundamental observation has consistently held true: CPUs are significantly more performant and more expensive than I/O devices. The fact that CPUs can process data at extremely high rates, while simultaneously servicing multiple I/O devices, has had a sweeping impact on the design of both hardware and software for systems of all sizes, for pretty much as long as we have been building them.

This assumption, however, is in the process of being completely invalidated.

The arrival of high-speed, non-volatile storage devices, typically referred to as storage class memories (SCM), is likely the most significant architectural change datacenter and software designers will face in the foreseeable future. SCMs are increasingly part of server systems, and they constitute a massive change: the cost of an SCM, at \$3,000–\$5,000, easily exceeds that of a many-core CPU (\$1,000–\$2,000), and the performance of an SCM (hundreds of thousands of I/O operations per second) is such that one or more entire many-core CPUs are required to saturate it.

This change has profound effects:

1. The age-old assumption that I/O is slow and computation is fast is no longer true: This invalidates decades of design decisions that are deeply embedded in today's systems.

IMAGE BY MARALOVA

2. The relative performance of layers in systems has changed by a factor of a thousand over a very short time: This requires rapid adaptation throughout the systems software stack.

3. Piles of existing enterprise datacenter infrastructure—hardware and software—are about to become useless (or, at least, very inefficient): SCMs require rethinking the compute/storage balance and architecture from the ground up.

This article reflects on four years of experience building a scalable enterprise storage system using SCMs; in particular, we discuss why traditional storage architectures fail to exploit the performance granted by SCMs, what is required to maximize utilization, and what lessons we have learned.

Ye Olde World

"Processing power is in fact so far ahead of disk latencies that prefetching has to work multiple blocks ahead to keep the processor supplied with data. ... Fortunately, modern machines have sufficient spare cycles to support more computationally demanding predictors than anyone has yet proposed."

—Papathanasiou and Scott,[10] 2005

That disks are cheap and slow, while CPUs are expensive and fast, has been drilled into developers for years. Indeed, undergraduate textbooks, such as Bryant and O'Hallaron's *Computer Systems: A Programmer's Perspective*,[3] emphasize the consequences of hierarchical memory and the importance for novice developers to understand its impact on their programs. Perhaps less pedantically, Jeff Dean's "Numbers that everyone should know"[7] emphasizes the painful latencies involved with all forms of I/O. For years, the consistent message to developers has been that good performance is guaranteed by keeping the working set of an application small enough to fit into RAM, and ideally into processor caches. If it is not that small, we are in trouble.

Indeed, while durable storage has always been slow relative to the CPU, this "I/O gap" actually widened yearly throughout the 1990s and early 2000s.[10] Processors improved at a steady pace, but the performance of mechanical drives remained unchanged, held hostage by the physics of rotational velocity and seek times. For decades, the I/O gap has been the mother of invention for a plethora of creative schemes to avoid the wasteful, processor-idling agony of blocking I/O.

Caching has always been—and still is—the most common antidote to the abysmal performance of higher-capacity, persistent storage. In current systems, caching extends across all layers: processors transparently cache the contents of RAM; operating systems cache entire disk sectors in internal buffer caches; and application-level architectures front slow, persistent back-end databases with in-memory stores such as memcached and Redis. Indeed, there is ongoing friction about where in the stack data should be cached: databases and distributed data processing systems want finer control and sometimes cache data within the user-space application. As an extreme point in the design space, RAMCloud[9] explored the possibility of keeping all of a cluster's data in DRAM and making it durable via fast recovery mechanisms.

Caching is hardly the only strategy to deal with the I/O gap. Many techniques literally trade CPU time for disk performance: compression and deduplication, for example, lead to data reduction, and pay a computational price for making faster memories seem larger. Larger memories allow applications to have larger working sets without having to reach out to spinning disks. Compression of main memory was a popular strategy for the "RAM doubling" system extensions on 1990s-era desktops.[12] It remains a common technique in both enterprise storage systems and big data environments, where tools such as Apache Parquet are used to reorganize and compress on-disk data in order to reduce the time spent waiting for I/O.

The Brave New World

"[M]ultiple sockets issuing IOs reduces the throughput of the Linux block layer to just about 125,000 IOPS even though there have been high end solid state devices on the market for several years able to achieve higher IOPS than this. The scalability of the Linux block layer is not an issue that we might encounter in the future, it is a significant problem being faced by HPC in practice today."

—Bjørling et al.,[2] 2013

The arrival of high-speed, non-volatile storage devices, typically referred to as storage class memories (SCM), is likely the most significant architectural change that datacenter and software designers will face in the foreseeable future.

Flash-based storage devices are not new: SAS and SATA SSDs have been available for at least the past decade, and have brought flash memory into computers in the same form factor as spinning disks. SCMs reflect a maturing of these flash devices into a new, first-class I/O device: SCMs move flash off the slow SAS and SATA buses historically used by disks, and onto the significantly faster PCIe bus used by more performance-sensitive devices such as network interfaces and GPUs. Further, emerging SCMs, such as non-volatile DIMMs (NVDIMMs), interface with the CPU as if they were DRAM and offer even higher levels of performance for non-volatile storage.

Current PCIe-based SCMs represent an astounding three-order-of-magnitude performance change relative to spinning disks (~100K I/O operations per second versus ~100). For computer scientists, it is rare the performance assumptions that we make about an underlying hardware component change by 1,000x or more. This change is punctuated by the fact the performance and capacity of non-volatile memories continue to outstrip CPUs in year-on-year performance improvements, closing and potentially even inverting the I/O gap.

The performance of SCMs means systems must no longer "hide" them via caching and data reduction in order to achieve high throughput. Unfortunately, however, this increased performance comes at a high price: SCMs cost 25x as much as traditional spinning disks ($1.50/GB versus $0.06/GB), with enterprise-class PCIe flash devices costing between $3,000–$5,000 each. This means the cost of the non-volatile storage can easily outweigh that of the CPUs, DRAM, and the rest of the server system they are installed in. The implication of this shift is significant: non-volatile memory is in the process of replacing the CPU as the economic center of the datacenter.

To maximize the value derived from high-cost SCMs, storage systems must consistently be able to saturate these devices. This is far from trivial: for example, moving MySQL from SATA RAID to SSDs improves performance only by a factor of 5–7[14]—significantly lower than the raw device differential. In a big data context, recent analyses of SSDs by Cloudera were similarly mixed: "we learned that SSDs offer considerable performance benefit for some workloads, and at worst do no harm."[4] Our own experience has been that efforts to saturate PCIe flash devices often require optimizations to existing storage subsystems, and then consume large amounts of CPU cycles. In addition to these cycles, full application stacks spend some (hopefully significant) amount of time actually working with the data that is being read and written. In order to keep expensive SCMs busy, significantly larger numbers of CPUs will therefore frequently be required to generate a sufficient I/O load.

All in all, despite the attractive performance of these devices, it is very challenging to effectively slot them into existing systems; instead, hardware and software need to be designed together with an aim of maximizing efficiency.

Here, we discuss some of the techniques and considerations in designing for extremely high performance and utilization in enterprise storage systems:

Balanced systems address capacity shortfalls and bottlenecks in other components that are uncovered in the presence of SCMs. For example, sufficient CPU cores must be available and the network must provide enough connectivity for data to be served out of storage at full capacity. Failing to build balanced systems wastes capital investment in expensive SCMs.

Contention-free I/O-centric scheduling is required for multiple CPUs to efficiently dispatch I/O to the same storage device, that is, to share a single SCM without serializing accesses across all the CPUs. Failing to schedule I/O correctly results in sub-par performance and low utilization of the expensive SCMs.

Horizontal scaling and placement awareness addresses resource constraints by eschewing the traditional filer-style consolidation, and instead distributes data across the cluster and proactively moves it for better load balancing. Failing to implement horizontal scaling and placement awareness results in storage systems that cannot grow.

Workload-aware storage tiering exploits the locality of accesses in most workloads to balance performance, capacity, and cost requirements. High-speed, low-capacity storage is used to cache hot data from the lower-speed tiers, with the system actively promoting and demoting data as workloads change. Failing to implement workload-aware tiering results in high-value SCM capacity being wasted on cold data.

Finally, we conclude by noting some of the challenges in datacenter and application design to be expected from the even faster non-volatile devices that will become available over the next few years.

Balancing Systems

Can we just drop SCMs into our systems instead of magnetic disks, and declare the case closed? Not really. By replacing slow disks with SCMs, we merely shift the performance bottleneck and uncover resource shortfalls elsewhere—both in hardware and in software. As a simple but illustrative example, consider an application that processes data on disk by asynchronously issuing a large number of outstanding requests (to keep the disk busy) and then uses a pool of one or more worker threads to process reads

Figure 1. Per-packet processing time with faster network adapters.

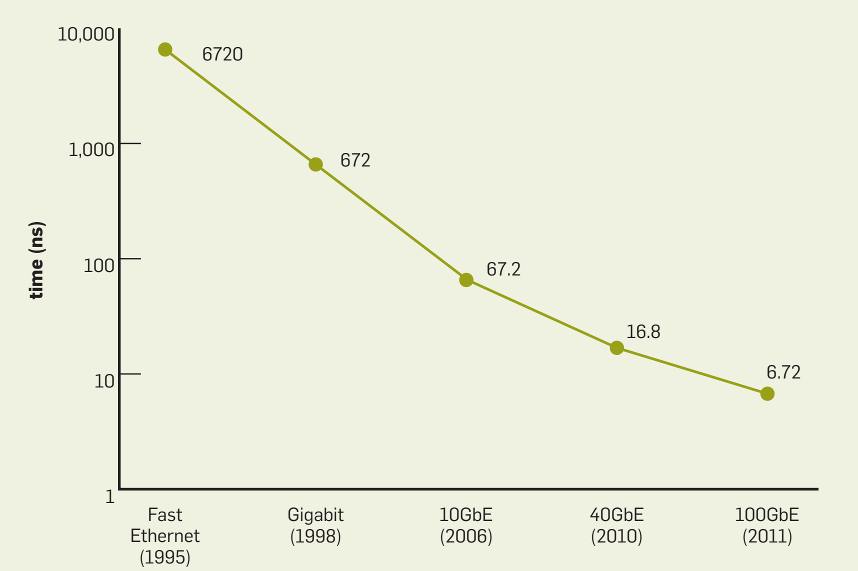
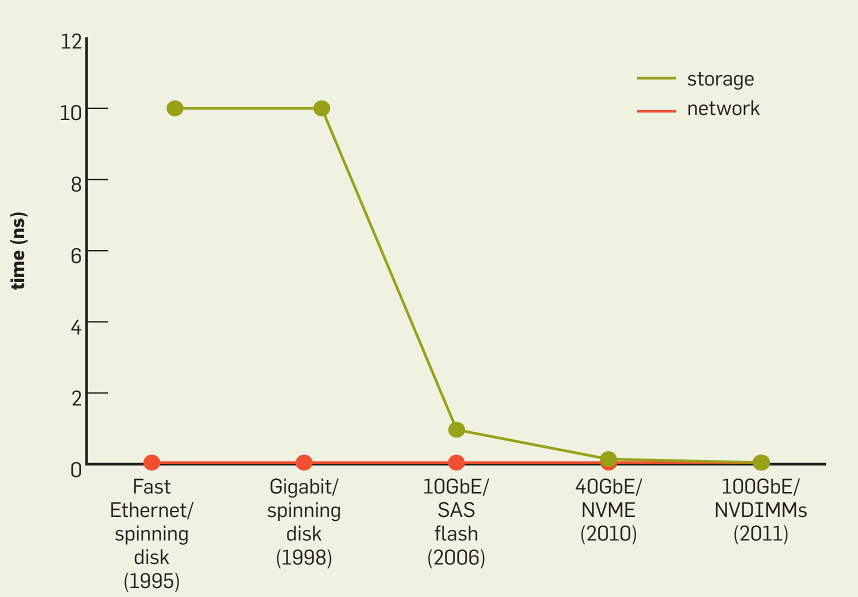


Figure 2. Progression in speed of SCMs compared to network adapters.



from disk as they complete. In a traditional system where disks are the bottleneck, requests are processed almost immediately upon completion and the (sensible) logic to keep disk request queues full may be written to keep a specified number of requests in flight at all times. With SCMs, the bottleneck can easily shift from disk to CPU: instead of waiting in queues ahead of disks, requests complete almost immediately and then wait for workers to pick them up, consuming memory until they are processed. As a result, we have seen real-world network server implementations and data analytics jobs where the concrete result of faster storage media is that significantly more RAM is required to stage data that has been read but not processed. Moving the performance bottleneck results in changes to memory demands in the system, which may, in the worst case, even lead to the host swapping data back out to disk!

Beyond the behavioral changes to existing software that result from the performance of SCMs, realizing their value requires they be kept busy. Underutilized and idle SCMs constitute waste of an expensive resource, and suggest an opportunity for consolidation of workloads. Interestingly, this is the same reasoning that was used, over a decade ago, to motivate CPU virtualization as a means of improving utilization of compute resources. Having been involved in significant system-building efforts for both CPU and now SCM virtualization, we have found achieving sustained utilization for SCMs to be an even more challenging goal than it was for CPUs. It is not simply a matter of virtualizing the SCM hardware on a server and adding more VMs or applications: We may encounter CPU or memory bottlenecks long before the SCM is saturated. Instead, saturating SCMs often requires using a dedicated machine for the SCM and spreading applications across other physical machines.

As a result, the cost and performance of storage devices dominates datacenter design. Ensuring their utilization becomes a key focus, and we found this can only be achieved by building balanced systems: systems with an appropriate number of cores and the right amount of memory to saturate exactly as many flash devices as needed for a given workload.

That balanced design with attention to storage can pay off is not a new insight: efforts like TritonSort, which won the sort benchmark in 2011 by carefully optimizing a cluster's magnetic disk throughput,[11] have done it before. However, such optimization efforts were rare in the age of magnetic disks, and are—certainly in TritonSort's case—hardware- and workload-specific. The traditional slowness of disks meant that balancing storage speed with other system components was out of the question for most workloads, so efforts focused on *masking* access to storage in order to stop it from disrupting the balance of other resources.

Good SCM utilization requires this balance far more urgently: buying too many flash devices and too few cores or too little RAM ends up wasting capital, but buying too few, sparsely spread out flash devices risks bottlenecks in accessing them—though the bottlenecks will most likely be in system resources other than the SCM itself! The right balance is, of course, still a property of the workload, which in combination with our consolidation goal makes it an incredibly challenging target to shoot for: heterogeneous workloads already make it difficult to achieve full system utilization, even before considering the storage layer. An example of this from our own work has been in the seemingly simple problem of dynamically scaling a traditional NFS server implementation to expose more physical bandwidth as additional SCMs (and accompanying CPUs and NICs) are added to the system.[5]

Contention-Free I/O-Centric Scheduling

Even if the hardware resources and the workload are perfectly balanced, the temporal dimension of resource sharing matters just as much. For a long time, interrupt-driven I/O has been the model of choice for CPU-disk interaction. This was a direct consequence of the mismatch in their speeds: for a core running at a few gigahertz, servicing an interrupt every few milliseconds is fairly easy. A single core can service tens or hundreds of disks without getting overwhelmed and missing deadlines.

This model must change drastically for low-latency ("microsecond era") devices. However, storage devices are not the only peripheral to have seen changes in speed—network devices have seen similar rapid improvements in performance from 10G to 40G and, recently, 100G. Maybe storage systems can use the same techniques to saturate devices?

Unfortunately, the answer is not a simple yes or no. The gains made by networking devices pale in comparison to the dramatic rise in speed of storage devices; for instance, Figures 1 and 2 show that in the same period of time when networks have sped up a thousandfold, storage devices have become a million times faster. Furthermore, storage stacks often have to support complex features such as compression, encryption, snapshots, and deduplication directly on the datapath, making it difficult to apply optimizations that assume independent packets without data dependencies.

One technique for reducing latency commonly adopted by network devices is to eliminate interrupt processing overhead by transitioning to polling when the system is under high load. Linux NAPI and Intel Busy Poll Sockets implement a polling mode for network adapters, which eliminates both the context switch and the cache and TLB pollution associated with interrupts. Also, busy polling cores never switch to power-saving mode, thereby saving on the cost of processor state transitions. Switching network adapters to polling mode reduces latency by around 30%, and non-volatile storage has demonstrated similar improvements.[16]

Polling comes with its own set of challenges, however. A CPU has responsibilities beyond simply servicing a device—at the very least, it must process a request and act as either a source or a sink for the data linked to it. In the case of data parallel frameworks such as Hadoop and Spark,[7,17] the CPU may also be required to perform more complicated transformations on the data. Thus, polling frequencies must be carefully chosen to ensure neither devices nor compute suffer from starvation, and scheduling strategies designed to exploit traditional I/O-heavy workloads need to be reevaluated, since these workloads are now necessarily compute-heavy as well.

At 100K IOPS for a uniform random workload, a CPU has approximately 10 microseconds to process an I/O request. Because current SCMs are often considerably faster at processing sequential or read-only workloads, this can drop to closer to 2.5 microseconds on commodity hardware. Even worse, since these requests usually originate from a remote source, network devices have to be serviced at the same rate, further reducing the available per-request processing time. To put these numbers in context, acquiring a single uncontested lock on today's systems takes approximately 20ns, while a non-blocking cache invalidation can cost up to 100ns, only 25x less than an I/O operation.

Current SCMs can easily overwhelm a single core; they need multiple cores simultaneously submitting requests to achieve saturation. While hardware multi-queue support allows parallel submission, the kernel block layer serializes access to the queues and requires significant redesign to avoid contended locks.[2] However, even with a contention-free block layer, requests to overlapping regions must be serialized to avoid data corruption.

Another key technique used by high-performance network stacks to significantly reduce latency is bypassing the kernel and directly manipulating packets within the application.[13] Furthermore, they partition the network flows across CPU cores,[1,8] allowing the core that owns a flow to perform uncontended, lock-free updates to flow TCP state.

While bypassing the kernel block layer for storage access has similar latency benefits, there is a significant difference between network and storage devices: network flows are largely independent and can be processed in parallel on multiple cores and queues, but storage requests share a common substrate and require a degree of coordination. Partitioning both the physical storage device and the storage metadata in order to give individual CPU cores exclusive access to certain data is possible, but it requires careful in-data-structure design that has not

Beyond the behavioral changes to existing software that result from performance of SCMs, realizing their value requires they be kept busy. Underutilized and idle SCMs constitute waste of an expensive resource, and suggest an opportunity for consolidation of workloads.

been required of storage and file system designers in the past. Our experience has been that networking code often involves data structures that must be designed for performance and concurrency, while file system code involves complex data dependencies that require careful reasoning for correctness. With SCMs, systems designers are suddenly faced with the need to deal with both of these problems at once.

The notion of I/O-centric scheduling recognizes that in a storage system, a primary task of the CPU is to drive I/O devices. Scheduling quotas are determined on the basis of IOPS performed, rather than CPU cycles consumed, so typical scheduling methods do not apply directly. For example, a common legacy scheduling policy is to encourage yielding when lightly loaded, in exchange for higher priority when busy and in danger of missing deadlines—a strategy that penalizes device-polling threads that are needed to drive the system at capacity. The goal of I/O-centric scheduling must be to prioritize operations that drive device saturation while maintaining fairness and limiting interference across clients.

Horizontal Scaling and Placement Awareness

Enterprise datacenter storage is frequently consolidated into a single server with many disks, colloquially called a JBOD (Just a Bunch Of Disks). JBODs typically contain 70–80 spinning disks and are controlled by a single controller or "head," and provide a high-capacity, low-performance storage server to the rest of the datacenter.

JBODs conveniently abstract storage behind this controller; a client need only send requests to the head, without requiring any knowledge of the internal architecture and placement of data. A single SCM can outperform an entire JBOD, but it provides significantly lower capacity. Could a JBOD of SCMs provide high-speed and high-capacity storage to the rest of the datacenter? How would this affect connectivity, power, and CPU utilization?

An entire disk-based JBOD requires less than 10G of network bandwidth, even when running at full capacity. In contrast, a JBOD of SCMs would require 350G–400G of network bandwidth, or approximately 10 40G network adapters. At 25W per SCM, the JBOD would draw approximately 3,000W.

Obviously, this is impractical, but even worse, it would be terribly inefficient. A single controller is simply not capable of mediating access to large numbers of SCMs simultaneously. Doing so would require processing an entire request in around 100ns—the latency of a single memory access. A centralized controller would thus leave storage hardware severely underutilized, providing a poor return on the investment in these expensive devices. A different approach is required.

Distributing accesses across cores, that is, having multiple heads, requires coordination while accessing file system metadata. Multiple network adapters within the JBOD expose multiple remote access points, requiring placement-aware clients that can direct requests to the correct network endpoint and head. At this point the JBOD resembles a distributed system, and there is little benefit to such consolidation. Instead, horizontal scaling out across machines in the cluster is preferable, as it provides additional benefits related to provisioning and load balancing.

Rather than finalizing a specification for a JBOD when first building the datacenter, scaling out allows storage servers to be added gradually in response to demand. This can lead to substantial financial savings as the incrementally added devices reap the benefits of Moore's Law. Further, since these servers are provisioned across racks, intelligent placement of data can help alleviate hotspots and their corresponding network bottlenecks, allowing for uniformly high utilization.

However, maintaining high performance across clustered machines requires much more than just reducing interrupt overheads and increasing parallelism. Access to shared state, such as file system metadata, must be carefully synchronized, and additional communication may be required to serve large files spread across multiple servers. Updates to files and their metadata must be coordinated across multiple machines to prevent corruption, and the backing data structures themselves must scale across cores with minimal contention. Shifting workload patterns often lead to poor load balancing, which can require shuffling files from one machine to another. Distributed storage systems have faced these issues for years, but the problems are much more acute under the extremely high load that an SCM-based enterprise storage system experiences.

The takeaway here is that unless the majority of data in the system is hot, it is extremely inefficient to store it all in high-speed flash devices.

Workload-Aware Storage Tiering

The capacity and performance of SCMs are orthogonal: a 4TB flash drive has about the same performance characteristics as a 1TB or 2TB drive in the same series. Workload requirements for capacity and performance are not matched to hardware capabilities, leading to underutilized disks; for example, a 10TB dataset with an expected load of 500K IOPS is half idle when all the data is stored in 1TB SCMs capable of 100K IOPS.

Besides the obvious cost inefficiency of having underutilized expensive SCMs, there are processor socket connectivity constraints for PCIe-based SCMs. A single such device requires four to eight PCIe lanes, which are shared across all the high-speed I/O devices, limiting the number of drives a single socket can support. In contrast, SATA drives, whether spinning disk or flash, do not count against the same quota.

The takeaway here is that unless the majority of data in the system is hot, it is extremely inefficient to store it all in high-speed flash devices. Many workloads, however, are not uniformly hot, but instead follow something closer to a Pareto distribution: 80% of data accesses are concentrated in 20% of the dataset.

A hybrid system with different tiers of storage media, each with different performance characteristics, is a better option for a mixture of hot and cold data. SCMs act as a cache for slower disks and are filled with hot data only. Access patterns vary across time and need to be monitored so the system can actively promote and demote data to match their current hotness level. In practice, tracking miss ratio curves for the system allows estimation of the performance impact of changing the cache size for different workloads with fairly low overheads[15] and enables fine-grained decisions about exactly where data should reside.

Tiering is an extension to caching mechanisms that already exist. System designers must account for tiers independently, rather like ccNUMA machines where local and remote memories have significantly different performance. Tiering allows systems to scale capacity and performance independently—a necessity for enterprise storage.

Despite the obvious benefits of tiering, it is fraught with complications. The difference in granularity of access at different storage tiers causes an impedance mismatch. For example, SCMs excel at random accesses, while spinning disks fare better with sequential access patterns. Maintaining a degree of contiguity in the disk tier may result in hot and cold data being "pinned" together in a particular tier.

This granularity mismatch is not unique to storage devices: MMUs and caches also operate at page and cache line granularities, so a single hot byte could pin an entire page in memory or a line in the cache. While there are no perfect solutions to this problem, the spatial locality of access patterns offers some assistance: predictable, repeated accesses allow for some degree of modeling to help identify and fix pathological workloads.

In adequately provisioned systems, simple tiering heuristics are often effective for making good use of hardware without degrading performance. However, different workloads may have differing priorities. In such cases, priority inversion and fairness become important criteria for determining layout. Tiering mechanisms must support flexible policies that prevent active but low-priority workloads from interfering with business-critical workloads. There is often a tension between such policies and the desire to maximize efficiency; balancing these concerns makes tiering a challenging problem.

The Future

PCIe SSDs are the most visible type of SCMs, and have already had a significant impact on both hardware and software design for datacenters—but they are far from the only member of that class of devices.

NVDIMMs have the performance characteristics of DRAM, while simultaneously offering persistence. A common recent approach to designing NVDIMMs has been to match the amount of DRAM on a DIMM with an equivalent amount of flash. The DRAM is then used as if it were normal memory, and the flash is left entirely alone until the system experiences a power loss. When the power is cut, a supercapacitor is used to provide enough power to flush the (volatile) contents of RAM out to flash, allowing it to be reloaded into RAM when the system is restarted. Flash-backed DRAMs are available today, and newer memory technologies such as resistive and phase-change memories have the potential to allow for larger and higher-performance nonvolatile RAM.

This emerging set of non-volatile memories has exposed software inefficiencies that were previously masked by the performance characteristics of the devices that they ran on—not just spinning disks, but even first-generation SSDs. We believe as these nefficiencies become more pronounced (and they likely will, given the continued fast pace of improvement in SCM performance), they will invite innovation in software systems design. This will have to occur at many layers of the infrastructure stack in order to be able to take advantage of fast non-volatile storage; what we see today is just the beginning!

Related articles on queue.acm.org

Don't Settle for Eventual Consistency
Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, David G. Andersen
http://queue.acm.org/detail.cfm?id=2610533

Enterprise Grid Computing
Paul Strong
http://queue.acm.org/detail.cfm?id=1080877

Why Cloud Computing Will Never Be Free
Dave Durkee
http://queue.acm.org/detail.cfm?id=1772130

References

1. Belay, A., et al. IX: A protected dataplane operating system for high throughput and low latency. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation.*
2. Bjørling, M., Axboe, J., Nellans, D. and Bonnet, P. Linux block IO: introducing multi-queue SSD access on multi-core systems. In *Proceedings of the 6th International Systems and Storage Conference*, 2013.
3. Bryant R. E. and O'Hallaron, D.R. *Computer Systems: A Programmer's Perspective, Vol. 2.* Prentice Hall, Englewood Cliffs, NJ, 2003.
4. Chen. Y. The truth about MapReduce performance on SSDs; http://radar.oreilly.com/2015/07/the-truth-about-mapreduce-performance-on-ssds.html
5. Cully, B. et al. 2014. Strata: Scalable high-performance storage on virtualized non-volatile memory. In *Proceedings of the 12th USENIX conference on File and Storage Technologies*, 2014.
6. Dean, J. Software engineering advice from building large-scale distributed systems. CS295 Lecture at Stanford University (July 2007); http://research.google.com/people/jeff/stanford-295-talk.pdf
7. Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, 2004.
8. Jeong, E.Y. et al. mTCP: A highly scalable user-level TCP stack for multicore systems. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation*, 2014.
9. Ongaro, D., Rumble, S. M., Stutsman, R., Ousterhout, J. and Rosenblum, M. Fast crash recovery in RAMCloud. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*, 2011.
10. Papathanasiou, A.E. and Scott. M.L. Aggressive prefetching: An idea whose time has come. In *Proceedings of the 10th USENIX Workshop on Hot Topics in Operating Systems*, 2005.
11. Rasmussen, A. et al. 2011. TritonSort: A balanced large-scale sorting system. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation*, 2011.
12. Rizzo, L. A very fast algorithm for RAM compression. *ACM SIGOPS Operating Systems Review 31*, 2 (1997), 36–45.
13. Rizzo, L. Netmap: A novel framework for fast packet I/O. In *Proceedings of the USENIX Annual Technical Conference*, 2012.
14. Tkachenko, V. Intel SSD 910 vs. HDD RAID in TPC-C-MySQL benchmark; https://www.percona.com/blog/2012/09/11/intel-ssd-910-vs-hdd-raid-in-tpcc-mysql-benchmark/
15. Wires, J., Ingram, S., Drudi, Z., Harvey, N. J. A., Warfield, A. Characterizing storage workloads with counter stacks. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation*, 2014.
16. Yang, J., Minturn, D.B., Hady, F. When poll is better than interrupt. In *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, 2012.
17. Zaharia, M. et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation*, 2012.

Mihir Nanavati is a developer at Coho Data, where he works on extracting maximum performance from high-speed I/O devices. He is also a Ph.D. student at the University of British Columbia. His research interests include performance and scalability of multicore systems, and security and isolation in virtualized and containerized environments.

Malte Schwarzkopf is a Ph.D. candidate at the University of Cambridge Computer Laboratory and moonlights part-time at Coho Data. His research is primarily on operating systems and scheduling for datacenters, but he has worked at many levels of the stack as part of the CamSaS initiative (http://camsas.org/).

Jake Wires is a principal software engineer at Coho Data and a doctoral candidate at the University of British Columbia. He is broadly interested in the design of storage systems and scalable data processing.

Andrew Warfield is co-founder and CTO of Coho Data and an associate professor at the University of British Columbia. He is broadly interested in software systems.

DOI:10.1145.2844112

Article development led by acmqueue
queue.acm.org

We need it, we can afford it, and the time is now.

BY PAT HELLAND

Immutability Changes Everything

THERE IS AN inexorable trend toward storing and sending immutable data. We *need immutability* to coordinate at a distance, and we *can afford immutability* as storage gets cheaper. This article offers an amuse-bouche of repeated patterns of computing that leverage immutability. Climbing up and down the compute stack really does yield a sense of déjà vu all over again.

It was not that long ago that computation was expensive, disk storage was expensive, DRAM (dynamic random access memory) was expensive, but coordination with latches was cheap. Now all these have changed using cheap computation (with many-core), cheap commodity disks, and cheap DRAM and SSDs (solid-state drives), while coordination with latches has become harder because latch latency loses lots of instruction opportunities. Keeping immutable copies of lots of data is now affordable, and one payoff is reduced coordination challenges.

Storage is increasing as the cost per terabyte of disk keeps dropping. This means a lot of data can be kept for a long time. Distribution is increasing as more and more data and work are spread across a great distance. Data within a data center seems "far away." Data within a many-core chip may seem "far away." Ambiguity is increasing when trying to coordinate with systems that are far away—more stuff has happened since you have heard the news. Can you take action with incomplete knowledge? Can you wait for enough knowledge?

Turtles all the way down.[17] As various technological areas have evolved, they have responded to these trends of increasing storage, distribution, and ambiguity by using immutable data in some very fun ways. This article explores how apps use immutability in their ongoing work, how they generate an immutable dataset for later offline analysis, how SQL can expose and process immutable snapshots, and how massively parallel big-data work relies on immutable datasets. This leads to looking at the ways in which semantically immutable dataset may be altered while remaining immutable.

Next, the article considers how updatability is layered atop the creation of new immutable files via techniques such as LSF (log-structured file system), COW (copy-on-write), and LSM (log-structured merge-tree). How do replicated and distributed file systems depend on immutability to eliminate anomalies? Hardware folks have joined the party by leveraging these tricks in SSDs and HDDs (hard-disk drives). Immutability is a key architectural concept at many layers of the stack, as shown in Figure 1.

Finally, the article looks at some of the trade-offs of using immutable data.

IMAGE BY EVA OZKOIDI

Accountants Don't Use Erasers

Many kinds of computing are *append-only*. This section looks at some of the ways this is commonly accomplished.

In append-only computing, observations are recorded forever (or for a long time). Derived results are calculated on demand (or periodically precalculated).

This is similar to a DBMS in which transaction logs record all the changes made to the database. High-speed appends are the only way to change the log. From this perspective, the contents of the database hold a caching of the latest record values in the logs. The truth is the log. The database is a cache of a subset of the log. That cached subset happens to be the latest value of each record and index value from the log.

Accounting: Observed and derived facts. Accountants don't use erasers; otherwise they may go to jail. All entries in a ledger remain in the ledger. Corrections can be made but only by making new entries in the ledger. When a company's quarterly results are published, they include small corrections to the previous quarter. Small fixes are OK. They are append-only, too.

Some entries describe *observed facts*. For example, receiving a debit or credit against a checking account is an observed fact. Some entries describe *derived facts*, meaning that based on the observations, something new can be calculated. For example, amortized capital expenses based upon a rate and a cost are derived facts. Another example is the current bank account balance with applied debits and credits.

Append-only distributed single master. Single-master computing means changes are ordered somehow. The order can come from a centralized master or some Paxos-like[11] distributed protocol providing serial ordering. Changes are semantically applied one at a time and are layered over their predecessors. New values supersede old ones. The granularity of this may be a set of records in a relational store or a new version of a document. Distributed single-master computing means there is a space of data (relational records, documents, export files, and more) that emanates from one logical location with new versions over time.

Distributed computing "back in the day." Before telephones, people used messengers—often kids walking through town to deliver the message. Alternatively, the postal service delivered the messages, which took a long time. Sometimes people used fancy forms with many layers, each a different color. They had multiple sections on the page. Each participant filled out a section (pressing hard with the pen), then tore off the back page of the form and filed it. Each participant got the data needed and added more data to the form. Earlier sections could not be updated; data could only be appended to the end.

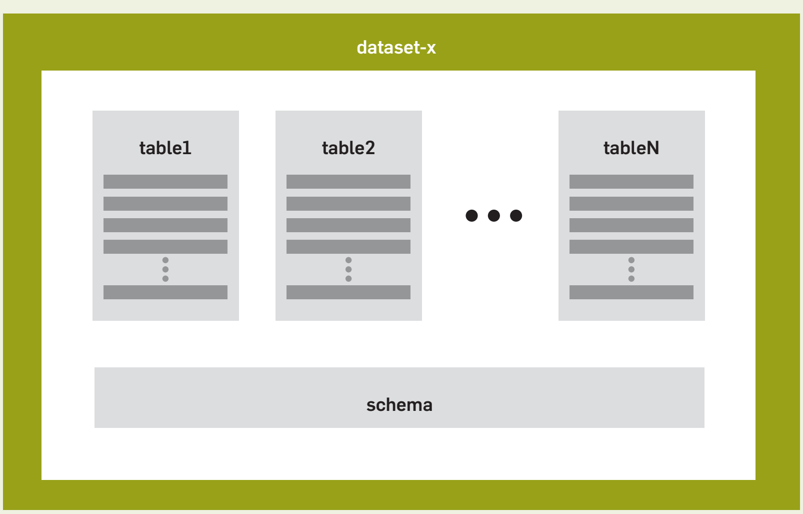
Figure 1. Immutability is a key architectural concept at many layers of the stack.

Layers	Usages of Immutable Data
Append-only apps	App over immutable data: record facts, then derive
App-generated datasets	Generate immutable data
Massively parallel big data	Read and write immutable datasets
SQL snapshots and datasets	Generate immutable data
Subjectively immutable datasets	Interpret data as immutable
LSF, LSM, and COW	Expose change over immutable files by append
Immutable files	Replication of files/blocks without update anomalies
Wear leveling on SSD	Change via COW to spread physical update blocks
Shingles on HDD	Change via COW to allow large physical rewrites

Figure 2. Characteristics of inside data and outside (immutable) data.

	Inside Data	Outside Data
Changeable	Yes!	No! Immutable
Granularity	Relational field	Document, file, or message
Representation	Typically relational	Typically semi-structured
Schema	Prescriptive	Descriptive
Identity	No identity: Data by values	Identity: URL, Msg#, Doc-ID...
Versioning	No versioning: data by value	Versions may augment identity

Figure 3. A dataset is a logical set of immutable tables and its schema.



Before computers, workflow was frequently captured in paper forms with multiple parts on the form and multiple pages (for example, "Fill out Part 3 and keep the goldenrod page from the back"). This "distributed computing" was append-only. New messages were new additions to the form—each was a version and each was immutable. You were never allowed to overwrite what had been written.

Data on the Outside vs. Data on the Inside

Surprisingly (to database old-timers), not all data is kept in relational database systems. This section (based on an earlier paper[7]) discusses some of the implications of unlocking data.

Data on the inside refers to what is kept and managed by a classic relational database system and its surrounding application code. Sometimes this is referred to as a *service*.

Data on the inside lives in a transactional world with changes applied in a serializable fashion (or something close to that).

Data on the outside is prepared as messages, files, documents, and/or Web pages. These are sent out from a service into the world. It is also possible that outside data has been created by some other mechanism than one using databases.

Data on the outside:

- *is immutable.* Once it is written, it is never changed.
- *is unlocked.* It is not locked in the database. A copy is extracted and sent outside.
- *has identity.* When sent outside, these files, documents, and messages have a unique identity (perhaps a URL).
- *may be versioned.* Updates are not updates but new versions with a new unique identifier.

Contrasting inside vs. outside. There are deep differences in the representation, meaning, and usage of inside data versus outside data. Increasingly, data is being kept as outside (immutable) data (see Figure 2).

Referencing Immutable Data

The dataset is a collection of data with a unique ID. Some datasets have structures that look like a number of tables with schema. How are these datasets referenced by a relational database, and how do relational operators span both the DBMS and dataset?

A dataset is a fixed and immutable set of tables. The schema for each table is captured in the dataset. The contents of each table are captured when the dataset is created. Since the dataset is immutable, it is created, may be consumed for reading, and then deleted. A dataset may be relational, or they may have some other representation such as a graph, a hierarchy such as JSON (JavaScript Object Notation), or any other representation (Figure 3). A dataset is a logical set of immutable tables along with its schema.

A dataset may be referenced by an RDBMS (relational DBMS). The metadata is visible to the DBMS. The data can be accessed for a read, even though it may not be updated. The dataset may be semantically present within the relational system even if it is physically stored elsewhere. Because the dataset is immutable, there is no need for locking and no worries about controlling updates.

Relational work on immutable datasets. A functional calculation takes a set of inputs and predictably creates a set of outputs. This can happen with a query against locked or snapshot data in a relational database, and it can happen on a big-data MapReduce-style system. In both cases, there is still an unchanging collection of data. With snapshots or some form of isolation, database data becomes semantically immutable for the duration of the calculation. With big-data calculations, the inputs are typically stored in GFS (Google File System) or HDFS (Hadoop Distributed File System) files.

There is no semantic obstacle to doing JOINs across data stored inside a relational database and data stored in an external dataset. Locking (or snapshot isolation) provides a version of the relational database, which may be joined. A named and frozen dataset may be joined with relational data (see Figure 4). You can meaningfully apply relational operations across data held in a DBMS and data held in an immutable dataset.

In some ways, the ability to work across immutable datasets and relational databases is surprising. An immutable dataset is defined with an identity and an optional version. Its schema, which describes the shape and form of the dataset at the time of its creation, is descriptive, whereas the schema held in the RDBMS is prescriptive.

This tailoring of the schema to meld the two connects the schema of the dataset (describing its data when written) with the schema of the RDBMS (describing its data as of the snapshot). Also, the JOINs and other relational operators must necessarily combine the contents of the dataset as interpreted as a set of relational tables. This sidesteps the notion of identity within the dataset and focuses exclusively on the tables as interpreted as a set of values held within rows and columns.

Immutability Is in the Eye of the Beholder

A consumer may see a dataset as immutable even if they change under the covers.

A dataset is semantically immutable. It has a set of tables, rows, and columns. It may also have semi-structured data (for example, JSON). It may have application-specific data in a proprietary format.

Dataset may be defined as a SELECTION, PROJECTION, or JOIN over a previously existing dataset. Semantically, all that data is now a part of the new dataset.

What is important about a dataset is it appears to be unchanging from the standpoint of the reader.

Optimizing a dataset for read patterns. Datasets are semantically immutable but can be physically changed. You can add an index or two. It is OK to denormalize tables to optimize for read access. Datasets can be partitioned and the pieces placed close to their readers. A column-oriented representation of a dataset may also make sense.

You can make a copy of a table with far fewer columns to optimize for quick access (a skinny table). The column values can be left in both the skinny table and fat table.

By watching and monitoring the read usage of a dataset, you may realize new optimizations (for example, new indices) are possible.

Immutability is the backbone of big data. Massively parallel computations are based on immutable inputs and functional calculations. MapReduce[3] and Dryad[9] both take immutable files as input. The work is cut into pieces, each with immutable input. This functional calculation (using immutable inputs) is idempotent, making it possible to fail and restart. Immutability is the backbone of big data. MapReduce performs functional computations over immutable data to create immutable outputs. Failure and restart, so essential to reliable big data, are based on the idempotent nature of functional computation over immutable inputs.

Immutability as a semantic prism. Datasets show an immutable semantic prism, even if the underlying representation is augmented or completely replaced. The King James Bible is, character for character, immutable—even when it is printed in a different font; even when digitized; even when accompanied by different pictures.

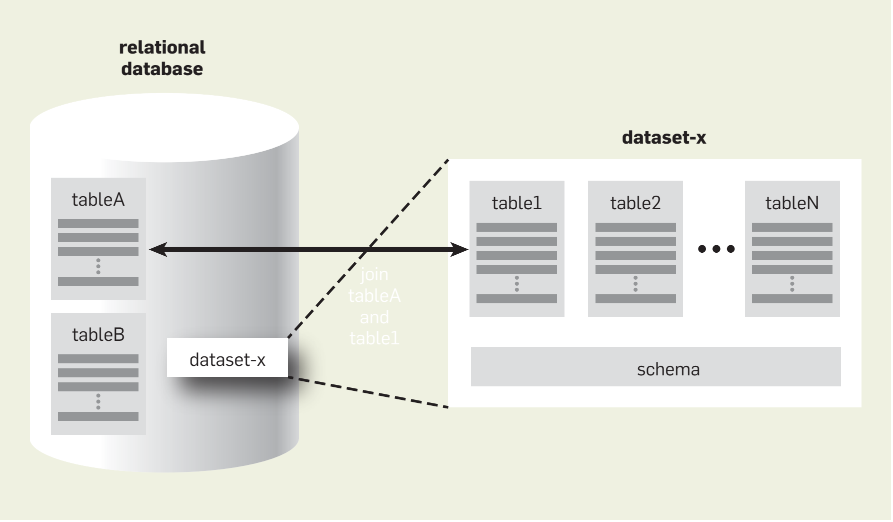
Is a dataset changed if there is a lossless transformation to a new schema representation? Can the new address field have more capacity? Can the enum values be mapped to a new underlying representation? Can the data be mapped from UTF-8 to UTF-16 encoding?

Having the right bits is not enough. You have to know how to interpret them. For example, "President Bush" had a different meaning in 1990 than in 2005. The word "napkin" is interpreted differently in the U.S. and the U.K.

Descriptive metadata when immutable. When an immutable dataset is created, the semantics of the data may not be changed. The contents may only be described as they are at the time the dataset is created.

Most programmers are used to SQL DDL (Data Definition Language) sup-

Figure 4. Immutable dataset may be joined with relational data.



porting dynamic changes in the metadata for their tables. This happens at a transaction boundary and can prescribe a new schema for the existing data. SQL DDL can be thought of as *prescriptive metadata* since it is prescribing the representation (which may change). Immutable datasets have *descriptive metadata* that explains what is there.

Of course, it is possible to create a new dataset that refers to one or more existing datasets in order to create a new representation of their data. Each new dataset has a unique ID. There is nothing wrong with having a dataset implemented by reference and not by value.

Normalization is for sissies. The goal of normalization is to eliminate update anomalies. When the data is not stored in a normalized fashion, updates might yield unpleasant results. The classic example is an imperfectly normalized table in which each employee has his or her manager's name and phone number. This makes it very difficult to update the manager's phone number since it is stored in many places. Normalization is very important in a database designed for updating.

Normalization is not necessary in an immutable dataset, however. The only reason to normalize immutable datasets may be to reduce the storage necessary for them. On the other hand, denormalized datasets may be easier and faster to process as inputs to a computation.

Versions Are Immutable, Too!

Each version is immutable. This section looks first at multiversion concurrency control; then techniques such as LSM that provide a semantic of change within a transactional space while generating immutable data that describes the state of these changes; finally, it looks at the world through the lens of COW, in which high-performance updates are implemented by writing new immutable data.

Versions and history. Versions should have immutable names. Other than the first version of something, a new version captures a replacement for or an augmentation of an earlier version. A *linear version history* is sometimes referred to as being strongly consistent: one version replaces another; there is one parent and one child; each version is immutable; each version has an identity. The alternative to linear version history is a DAG (directed acyclic graph) of version history, in which there are many parents and/or many children. This is sometimes called eventual consistency.

Multiversion concurrency control. Strongly consistent, or ACID (atomicity, consistency, isolation, durability), transactions appear as if they run in a serial order. This is sometimes called serializability.[2]

The database changes version by version. Transaction T1 is a version and later transaction T2 is a version. Transactions layer new versions of record and index changes atop earlier versions. The new versions can be captured as snapshots of the entire database (although this would not result in high performance).

Alternatively, the new version can be captured as changes to the previous version. In this way, a key-value store can be built, and a relational database can be built atop a key-value store. Records are deleted by adding tombstones. Changing the database is done by adding new records to the key-value store.

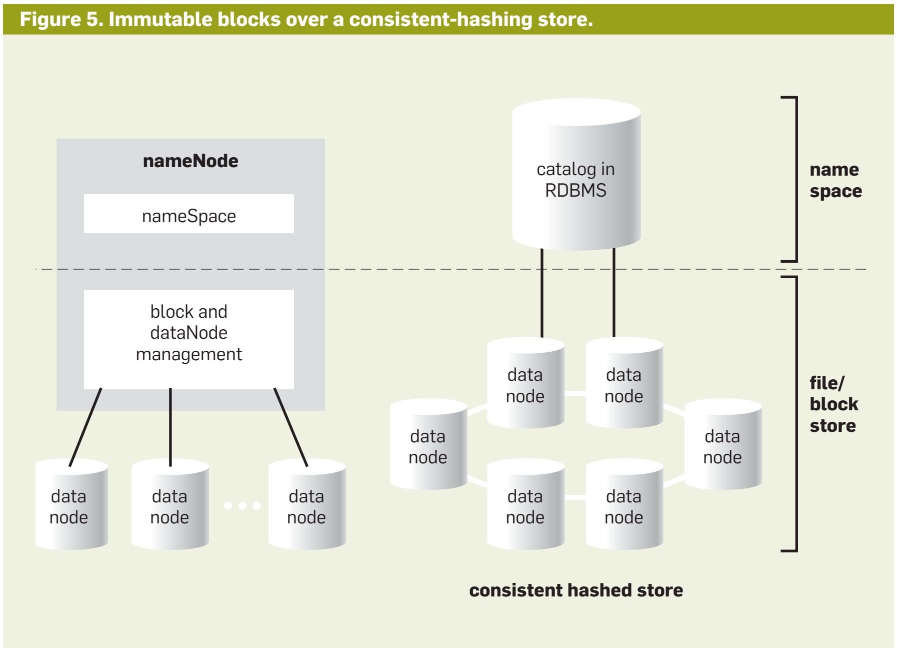
If a timestamp is added to each new version, it is possible to show the state of the database at a given point in time. This allows the user to navigate the state of the database to any older version. Ongoing work can see a stable snapshot of a version of the database.

LSM: Reorganizing immutable stuff. LSM presents a façade of change atop immutable files. With an LSM tree,[15] changes to the key-value store are accomplished by writing new versions of the affected records. These new versions are logged to an immutable file. Periodically, the new versions of the key values are sorted by key and written to an immutable file known as a Level 0 file within the LSM tree. Level 0 files are merged into a collection of Level 1 files (typically 10 Level 1 files, each containing one-tenth of the key range). Similarly, Level 1 files are merged with Level 2 files on a 10-to-1 basis. As you move down the LSM tree, each level has 10 times as many files. Reading a record typically involves searching one file per level. As the LSM files merge, new immutable files with new identities can be written.

Go ahead … have a COW! An LSM tree can create changeable data out of immutable files by performing a COW. The granularity of the copy is typically a key-value pair. For a relational database, this can be a key-value pair for each record or each index entry. The changes are copied into the log and then into the LSM tree (and copied a few more times for merges).

High-performance COW happens with logging and classic DBMS performance techniques. The new versions are captured in memory and logged for failure recovery. The identity of each log file is a unique ID, and the log files are immutable. Each new log file can record the history of its preceding log files and

Figure 5. Immutable blocks over a consistent-hashing store.



even the identity of upcoming log files. Having one of the recent log-file IDs means the entire LSM key-value store can be reconstructed.

Keeping the Stone Tablets Safe

Many file systems keep immutable files consisting of immutable blocks. This section explores at a high level the implementation of GFS and HDFS and the implications of what can be done with these files. It discusses the vagaries of files that can be renamed and considers the value of storing immutable data within a consistent hash store.

Log-structured files: Running in circles. An early example of reifying change through immutability is the log-structured file system.[16] In this wonderful invention, file-system writes are always appended to the end of a circular buffer. Occasionally, enough metadata to reconstruct the file system is added to the circular buffer. Old data must be copied forward so it is not overwritten.

Log-structured file systems have some interesting performance characteristics, both good and bad. Today they are an important technique. As technology trends continue to move in the direction of recent years, they will become even more important.

Files, blocks, and replication. GFS,[5] HDFS,[1] and others offer highly available files. Each file is a bunch of blocks (also called chunks). The file consists of a file name and a description of the blocks needed to provide a bytestream. Each block is replicated in the cluster for durability and high availability. They are typically replicated three times over different fault zones in the data center.

Each file is immutable and (typically) single-writer. The file is created, and one process can append to it. The file lives for a while and is eventually deleted. Multiwriters are difficult, and GFS had some challenges with this.[13]

Immutable files and immutable blocks empower this replication. The file system has no concept of a change to a complete file. Each block's immutability allows it to be easily replicated without any update anomalies because it does not get updated.

Widely sharing immutable files is safe. An immutable file has an identity and contents, neither of which can change. You can copy an immutable file whenever and wherever you want and share the immutable copies across users. As long as you manage reference counts (so you know when it is OK to delete it), you can use one copy of the file to share across many users. You can distribute immutable files wherever you want. With the same identity and same contents, the files are location independent.

Names and immutability ... A slippery slope. GFS and HDFS both provide immutable files. Immutable blocks (chunks) are replicated across *data nodes*. Immutable files are a sequence of blocks, each of which is identified with a GUID (globally unique identifier). The contents of a file are immutable and labeled with a GUID. The file-ID GUID always refers to exactly one file and its contents.

GFS and HDFS also provide a namespace that can be changed. The logical name of an immutable file may be changed. File names may be rebound to different contents. Users must take great care to ensure they have predictable results when changing file names. Is something really immutable when its name can change?

Immutable data and consistent hashing. Consider a strongly consistent file system in which a single master is controlling a namespace (perhaps a Posix-style namespace). Looking up a file results in a GUID that is used to find an immutable bytestream.

Now consider a store implemented with consistent hashing.[10] It is well understood that consistent hashing offers very robust rebalancing under failures and/or additional capacity. It also has somewhat chaotic placement behavior while the ring is adjusting to changes. At times, some participants have seen the changes and others have not. When reading and updating within a consistent-hashing key-value store, the read occasionally yields an older version of the value. To cope with this, the application must be designed to make the data eventually consistent.[4] This is a burden and makes application development more difficult.

When storing immutable data within a consistent-hashing ring, you cannot get stale versions of the data. Each block stored has the only version it will ever have. This provides the advantages of a self-managing and master-less file store while avoiding the anomalies and challenges of eventual consistency as seen by the application (Figure 5).

Using an eventually consistent store to hold immutable data also means log writes can have more predictable SLAs (service-level agreements) by allowing the replicas to land in less predictable locations in the cluster. In a distributed cluster, you can know *where* you are writing or you can know *when* the write will complete but not both.[8] By preallocating files from the strongly consistent catalog, log writes using the file IDs need only to touch weakly consistent servers to be able to retry getting the blocks durable in a bounded time.

Immutability and decentralized recovery. Separating the namespace from block-placement control has a number of advantages. The consistent-hashing ring can take writes and reads even when the ring is in flux.

Although the catalog is a central point for access, it does not have the same varying load a name node does when handling failures in the cluster. The larger the cluster, the more data nodes will fail, each necessitating many controlling operations to elevate the replica count back to three. While this traffic happens, operations to read and write from the cluster will experience SLA variation. Immutability allows decentralized recovery of data-node failures with more predictable SLAs.

Hardware Changes Toward Unchanging

The trend toward leveraging immutability in new designs is so pervasive it can be seen in a number of hardware areas. Here, I examine the implementation of SSDs and some new trends in hard disks.

SSDS and wear leveling. The flash chip within most SSDs is broken into physical blocks, each of which has a finite number of times it may be written before it begins to wear out and give increasingly unreliable results. Consequently, chip designers have a feature known as wear leveling[12] to mitigate this aspect of flash. Wear leveling is a form of COW and treats each version of the block as an immutable version.

Each new block or update to a block in the logical address space of the flash chip is mapped to a different physical block. Each new write (or update to a new block) is written to a different physical block in a circular fashion, evening out the writes so each physical block is written about as often as the others.

Hard disks: Getting the shingles. As hard-disk manufacturers strive to increase the areal density of the data on disk, some physical headaches have intervened. Current designs have a much larger write track than read track. Writes overlap the previous ones in a fashion evocative of laying shingles on a roof—hence the name *shingled disk systems*.[6]

In shingled disks, a large band of data is written as layered write tracks forming a shingle pattern, partially overwriting the preceding tracks. The data in the middle of the band cannot be overwritten without trashing the remaining part of the band.

To overcome this, the hardware disk controllers implement log-structured file systems within the disk controller.[14] The operating system is unaware of the use of shingles. What is written to the disk (that is, the band of data written with shingles) remains unchanged until it is discarded. The user of the disk (for example, the operating system) perceives the ability to update in place.

Immutability May Have Some Dark Sides

As immutability is leveraged in all these ways, there are trade-offs to be managed. Denormalized documents help with read performance at the expense of extra storage cost. Data is copied many times with COW. This is exacerbated when these mechanisms are layered.

Denormalization: Nimble but fat. Denormalization consumes storage as a data item is copied multiple times in a dataset. It is good in that it eliminates JOINs to put the data together, making the use of the data more efficient. Immutable data has more choices for its representation. It can be normalized for space optimization or denormalized for read usage.

Write amplification vs. read perspiration. Data may be copied many times with COW (for example, with log-structured file systems, log-structured merge systems, wear leveling in SSDs, and shingle management in HDD). This is known as *write amplification*.[18]

In many cases, there is a relationship between the amount of write amplification and the difficulty involved in reading the data being managed. For example, some LSM systems will do more or less copying as the data is reorganized and merged. If the data is aggressively merged and reorganized, then fewer places need checking to read a record. This can reduce the cost of reading at the expense of additional writing.

Conclusion

Designs are driving toward immutability, which is needed to coordinate at ever increasing distances. Given space to store data for a long time, immutability is affordable. Versioning provides a changing view, while the underlying data is expressed with new contents bound to a unique identifier.

▸ *Copy-on-write.* Many emerging systems leverage COW semantics to provide a façade of change while writing immutable files to an underlying store. In turn, the underlying store offers robustness and scalability because it is storing immutable files. For example, many key-value systems are implemented with LSM trees (for example, HBase, BigTable, and LevelDB).

▸ *Clean replication.* When data is immutable and has a unique identifier, many challenges with replication are eased. There is never a worry about finding a stale version of the data because no stale versions exist. Consequently, the replication system may be more fluid and less picky about where it allows a replica to land. There are also fewer replication bugs.

▸ *Immutable datasets.* Immutable datasets can be combined by reference with transactional database data and offer clean semantics when the dataset project relational schema and tables. Looking at the semantics projected by an immutable dataset, you can create a new version of it optimized for a different usage pattern but still projecting the same semantics. Projections, redundant copies, denormalization, indexing, and column stores are all examples of optimizing immutable data while preserving its semantics.

▸ *Parallelism and fault tolerance.* Immutability and functional computation are keys to implementing big data.

Related articles on queue.acm.org

If You Have Too Much Data, then "Good Enough" Is Good Enough
Pat Helland
http://queue.acm.org/detail.cfm?id=1988603

Enhanced Debugging with Traces
Peter Phillips
http://queue.acm.org/detail.cfm?id=1753170

Condos and Clouds
Pat Helland
http://queue.acm.org/detail.cfm?id=2398392

References

1. Apache Hadoop; http://en.wikipedia.org/wiki/Apache_Hadoop.
2. Bernstein, P., Hadzilacos, V. and Goodman, N. *Concurrency Control and Recovery in Database Systems.* Addison Wesley, 1987.
3. Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th Annual Symposium on Operating System Design and Implementation*, 2004.
4. DeCandia, G. et al. Dynamo: Amazon's highly available key-value store. In *Proceedings of the 21st Annual ACM Symposium on Operating Systems Principles*, 2007.
5. Ghemawat, S., Gobioff, H. and Leung, S. The Google File System. In *Proceedings of the 19th Annual ACM Symposium on Operating Systems Principle*, 2003.
6. Gibson, G. and Ganger, G. Principles of operation for shingled disk devices. *Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-11-107*, 2011.
7. Helland, P. Data on the outside versus data on the inside. In *Proceedings of the Conference on Innovative Database Research*, 2005.
8. Helland, P. Heisenberg was on the write track. *Abstract: Proceedings of the Conference on Innovative Database Research*, 2014.
9. Isard, M., Budiu, M., Yu, Y., Birrell, A. and Fetterly, D. Dryad: Distributed data-parallel programs from sequential building blocks. In *Proceedings of the European Conference on Computer Systems*, 2007.
10. Karger, D., Lehman, E., Leighton, T., Panigraphy, R., Levine, M. and Lewin, D. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997.
11. Lamport, L. The part-time parliament. *ACM Transactions on Computer Systems 16*, 2 (1998), 133-169.
12. Lofgren, K., Normal, R., Thelin, G. and Gupta, A. Wear-leveling techniques for flash EEPROM systems. US Patent #6850443, 2003, SanDisk, Western Digital.
13. McKusick, M. and Quinlan, S. GFS: Evolution on fast forward. *ACM Queue 7*, 7 (2009).
14. New, R. and Williams, M. Log-structured file system for disk drives with shingled writing. US Patent #7996645, 2003, Hitachi.
15. O'Neil, P., Cheng, E., Gawlick, D. and O'Neil, E. The log-structured merge-tree (LSM-tree). *Acta Informatica 33*, 4 (1996).
16. Rosenblum, M. and Ousterhout, J. The design and implementation of a log-structured file system. *ACM Transactions on Computer Systems 10*, 1 (1992), 26–52.
17. Wikipedia. Turtles all the way down; http://en.wikipedia.org/wiki/Turtles_all_the_way_down.
18. Wikipedia. Write amplification; http://en.wikipedia.org/wiki/Write_amplification.

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

18th International Conference on Human-Computer Interaction with Mobile Devices and Services

MobileHCI 2016

September 6-9, Florence, Italy

MobileHCI seeks contributions in the form of innovations, insights, or analyses related to human interaction and experiences with mobile devices and services. We seek richness and diversity in topic as well as approach, method, and viewpoint. In no particular order, this includes contributions in:

Systems & Infrastructures. The design, architecture, deployment, and evaluation of systems and infrastructures that support development of or interaction with mobile devices and services.

Devices & interaction techniques. The design, construction, usage, and evaluation of devices and techniques that create valuable new capabilities for mobile human-computer interaction.

Applications & experiences. Descriptions of the design, empirical study of interactive applications, or analysis of usage trends that leverage mobile devices and systems.

Methodologies & tools. New methods and tools designed for or applied to studying or building mobile user interfaces, applications, and mobile users.

Theories & models. Critical analysis or organizing theory with clearly motivated relevance to the design or study of mobile human-computer interaction; taxonomies of design or devices.

Visions. Well-argued and well-supported visions of the future of mobile computing or non-traditional topics that bear on mobility.

We solicit original research and technical papers not published elsewhere focusing on the following topics (but not limited to):

- Novel user interfaces and interaction techniques
- Mobile social networks and systems
- Design of location-based or context-aware services for mobile devices
- Context-aware systems
- Multimodal interaction (including audio and speech)
- User-centred design tools and methods for mobile systems
- Ethnographical and field studies with mobile technology
- Group interaction and mobility communities
- Services for mobile devices
- Design, evaluation and case studies of mobile applications
- Wearable computing, smart clothes, new devices and sensors
- Design and tools for cross-device user interfaces
- Mobile entertainment, storytelling and location based gaming
- Aesthetic interaction and experience design
- Affective computing in the mobile context
- Personal assistance with mobile devices
- Mobility and work environments
- Evaluation and usability of mobile devices and services
- Mobile accessibility

DEADLINES

FEBRUARY 12, 2016
for Full and Short Papers

FEBRUARY 26, 2016
for Workshop proposals

MAY 13, 2016
for Posters, Demos, and Tutorials

GENERAL CHAIRS

Fabio Paternò (CNR-ISTI, Italy)
Kaisa Väänänen (TUT, Finland)

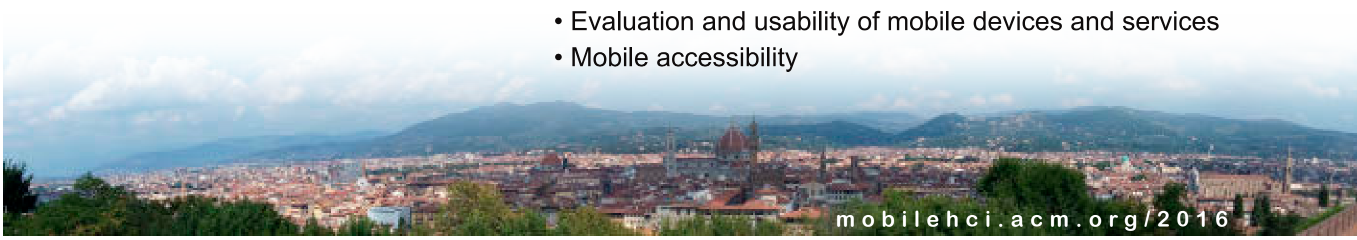
PAPER CHAIRS

Karen Church (Yahoo, USA)
Jonna Hakkila (U. Lapland, Finland)
Antonio Kruger (DFKI, Germany)
Marcos Serrano (U. Toulouse, France)

MobileHCI is sponsored by

Association for Computing Machinery

SIGCHI



DOI:10.1145/2756547

Knowing where you are in space and time promises a deeper understanding of neighbors, ecosystems, and the environment.

BY SHASHI SHEKHAR, STEVEN K. FEINER, AND WALID G. AREF

Spatial Computing

SPATIAL COMPUTING ENCOMPASSES the ideas, solutions, tools, technologies, and systems that transform our lives by creating a new understanding of locations—how we know, communicate, and visualize our relationship to locations and how we navigate through them. Pervasive GPS allows hikers in national parks, boaters on lakes, children visiting new places, and taxis (or Uber drivers or self-driving cars) and unmanned aerial vehicles to know their locations, nearby facilities, and routes to reach places of interest.[a]

Large organizations use spatial computing for site selection, asset tracking, facility management,

a Participants in the Computing Community Consortium's 2012 workshop used the term "spatial computing" as a generalization for spatial data structures,[46] spatial databases,[50] spatial data mining,[10] spatial statistics,[12] spatial cognition,[8] and other computational issues related to geographic and non-geographic spaces (such as sky catalogs, indoors, and VLSI design). Within geographic spaces, the term focuses on computational aspects of a multidisciplinary area, variously referred to as geoinformatics, geomatics, geocomputation, geoinformation science, geographical information science, and computational geography. More broadly, spatial computing refers to the study of computing in spatial, temporal, spatiotemporal spaces across both geographic and nongeographic domains.

» key insights

- **Starting with the public availability of GPS, spatial computing has enriched our lives through location-based services (such as Google Maps, Uber, geotagging, and geotargeted, including Amber, alerts).**
- **It has also advanced computer science through ideas like spatial databases (such as R-tree and OGIS simple features library), spatial statistics (such as point process theory and Kriging), and spatial data mining (such as robust hotspot detection).**
- **Future potentially transformative opportunities include ubiquitous indoor location-based services, the location-aware Internet of Physical Things, continuous global monitoring, visualization, forecast, alerts, and warnings to address societal challenges like climate change and how to provide adequate food, energy, and water.**

ILLUSTRATION BY PETER CROWTHER ASSOCIATES

navigation, and logistics. Scientists use Global Navigation Satellite Systems, or GNSS[24] (such as the global positioning system, or GPS), to track endangered species and better understand animal behavior, while farmers use these technologies to support precision agriculture to increase crop yields and reduce costs. Virtual globes[14] (such as Google Earth and NASA World Wind) help teach schoolchildren about their local neighborhoods and the world beyond (such as the Wini Seamount near Hawaii, extraterrestrial landscapes on Mars and the Moon, and the Sloan Digital Sky Survey) in an engaging and interactive way. In the wake of recent natural disasters (such as Hurricane Sandy in 2012), Google Earth has allowed millions of people to access imagery to help disaster-response-and recovery services.[26] Within days of the 2010 Haiti earthquake, post-disaster roadmaps had been created thanks to citizen volunteers submitting timely local information to the popular volunteered geographic information[13] website OpenStreetMaps.[44]

In the coming decade, spatial computing promises an array of transformative capabilities; for example, where route finding today is based on shortest travel time or distance, companies are experimenting with eco-routing, finding routes that minimize fuel consumption and greenhouse-gas emissions. Smart routing that avoids left turns saves delivery company UPS more than three million gallons of fuel annually.[20] Such savings can be multiplied many times over when eco-routing services are available for consumers, as well as fleet owners, including public transportation.

The ubiquity of mobile phones represents an opportunity for gathering information about all aspects of our world and the people in it.[17] Research has shown the potential for mobile phones with built-in motion detectors carried by everyday users to detect earthquakes seconds after they begin.[11] Navigation companies (such as Waze; https://www.waze.com/) increasingly use mobile phone records to estimate traffic levels on busy highways. There is a growing need for a cyberinfrastructure (such as the Earth Cube initiative from the National Science Foundation, http://www.nsf.gov/geo/earthcube/) to facilitate our understanding of the Earth as a complex system. Technological advances have greatly facilitated the collection of data from the field and the laboratory and simulation of Earth systems. This has resulted in exponential growth of geoscience data and the dramatic increase in our ability to accommodate diverse phenomena in models of Earth systems. Such advances may be crucial for understanding our changing planet and its physics (such as in oceans, atmosphere, and land), biology (such as plants, animals, and ecology), and society (such as climate change,[19] sustainable economic development, understanding interactions among food, energy, and water systems,[36] and connected and autonomous cars[1]).

Work in spatial computing has been extensive in recent decades, particularly in the geographic context. It is difficult to convey the breadth and depth of this large interdisciplinary body of work to the broad computing community in a magazine article. Our goal here is thus twofold: share a broad perspective on spatial computing based on discussions at the 2012 Computing Community Consortium workshop (http://cra.org/ccc/events/spatial-computing-workshop/) and start a broader discussion on the role the larger computing community can play in this interdisciplinary area. We do this by describing a few examples from the workshop without trying to either prioritize or be comprehensive; more examples are covered in the Appendix and in Shekhar et al.[52] Finally, we advocate support for the interdisciplinary field beyond the examples presented here. We include several figures to illustrate societal stories and visions from the workshop.

Transformative Accomplishments

Spatial computing initially aimed to support computational representation and analysis of maps and other geographic data. Its influence was concentrated in highly specialized disciplines (as represented by the professional organizations listed in Table 1). Since then, a number of transformative spatial computing technologies have become deeply integrated into society at large, helping answer many kinds of questions humans have always asked. Here, we briefly describe a few applications and research results of high significance and broad interest; for a deeper exploration of spatial computing, see various textbooks,[3,5,6,47,50] monographs,[45,48] encyclopedias,[16,53] and journals.[4]

Table 1. Representative spatial computing organizations.

ACM SIGSPATIAL
International Society of Photogrammetry and Remote Sensing
International Geographical Union
IEEE Geoscience and Remote Sensing Society
Institute of Navigation
Society of Photo-optics Instrumentation Engineers

Global Positioning System. Where am I on the surface of the Earth? In the 18th century, "the longitude problem"[55] was among the most challenging in science. Lacking the ability to measure their longitude, sailors in the great ages of exploration were literally lost at sea as soon as land was out of sight. Eventually, with the combined help of compasses, maps, star positions, and the chronometer (a clock that worked on moving ships), it became possible to position oneself with some level of precision, even in the middle of the ocean with no landmarks. With the 1978 launch of GPS and subsequent availability for civilian use, it is now possible to quickly and precisely locate oneself anywhere on the surface of the Earth. GPS is an example of the space-based GNSS,[24,27] which provides location and time information anywhere on Earth where there is an unobstructed line of sight to four or more navigation satellites (out of a few dozen).[39] GNSS-based accurate timekeeping facilitates everyday activities (such as clock synchronization in computer networks, including the Internet), geographic distributed sensor grids to monitor moving objects (such as missiles, planes, vehicles, and tectonic plates), and electric power distribution grids. Its localization capabilities have made possible a number of location-based services for end users (such as turn-by-turn navigation, local search, and geocoding). GNSS and related location-based services are today widely deployed and useful for commerce, science, tracking, and surveillance. Widespread proliferation of GPS systems was made possible by its low-cost very-large-scale integrated (VLSI) circuits implementations that could easily be integrated into mobile phones and tablets.

Remote sensing.[3] What fraction of the terrestrial surface is covered by forest? How has the forest cover changed in recent decades in the face of climate change, urbanization, and population growth? Traditionally, these questions were answered through manual land surveys, which are labor-intensive and thus often limited to small areas. Modern remote sensing satellites (such as MODIS; http://modis-land.gsfc.nasa.gov/, and Landsat, http://landsat.usgs.gov/) have made it possible to monitor land cover changes continuously[31] on a global scale. Moreover, specialized instruments can sense subsurface resources (such as aquifers and an underground ocean on Jupiter's largest moon Ganymede). Due to the large data volume, computing technologies are crucial in storing, querying, and analyzing remote sensing datasets. These datasets have also inspired computing innovations like Google Earth Engines.[43]

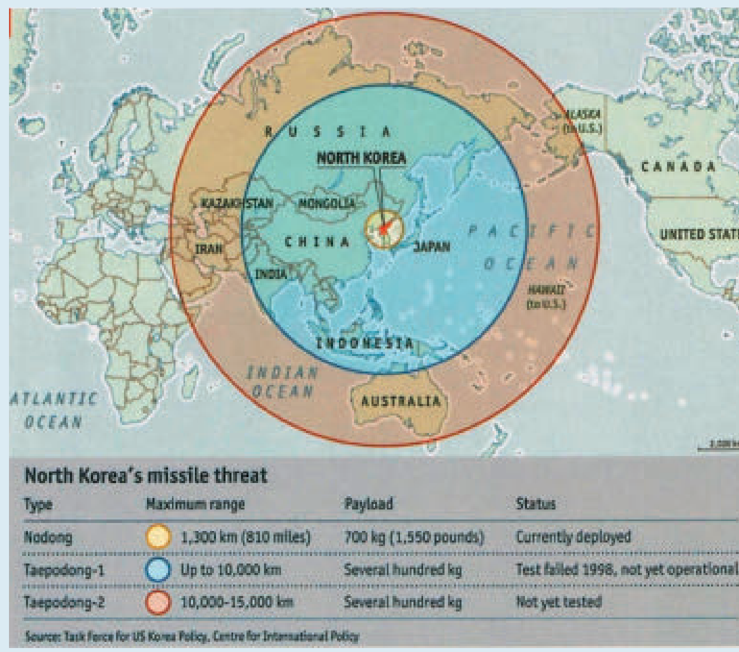
Geographic information system. Which countries can be reached by North Korea's missiles? Figure 1 is a well-known example of erroneous distance information computed on a planar map using circular distance, a mistake easily made without the help of GIS supporting spherical measurements. GIS understands a large number of map projections used by common geographic data producers and aids in fusing map data from diverse sources. As the Earth is not a perfect sphere, GIS also understands more accurate representations of the Earth, including ellipsoid representations and non-parametric representations that use land-based geodetic reference points for localization. GIS captures, stores, analyzes, manages, and visualizes spatial data;[22,53] for example, a map of the Earth is a representation of a curved surface on a plane. While map projections largely retain topological properties (except at map boundaries), retention of metric properties (such as distance and area) depends on the projection being used. GIS has a number of unique capabilities (such as cartography, geodetic data, and map layers). GIS can also join tables based on geometry to support spatial querying and statistical analysis, as explored in the next two paragraphs. GIS has greatly benefited from computing advances (such as algorithms like plane-sweep) and data structures (such as triangulated irregular networks related to map rendering and map overlay).

Spatial database management systems. Within the Sloan Digital Sky Survey, find galaxy pairs that are within 30 arc-seconds of each other. Which houses are most likely to be flooded by global warming-induced sea-level rise or cloud bursts or spring snow melt? Before development of spatial databases, such spatial queries required extensive programming and suffered from long computation times due to the mismatch between 2D spatial data and 1D datatypes (such as number) and indexes used by traditional database systems (such as B+ Tree). In addition, a naive collection of spatial data types is inadequate for multistage queries since the result of some queries (such as the union of disjoint polygons) cannot naturally be represented as a point, line, or polygon. Spatial databases[50] (such as Oracle Spatial and PostGIS) introduced spatial data types (such as OGIS simple features[38]), operations (such as inside and distance), spatial data structures (such as R-trees and Voronoi diagrams), and algorithms (such as shortest-path, nearest-neighbor, and range query) to represent and efficiently answer multistage concurrent spatial queries. The reduced programming effort resulted in more compact code and quicker response times.

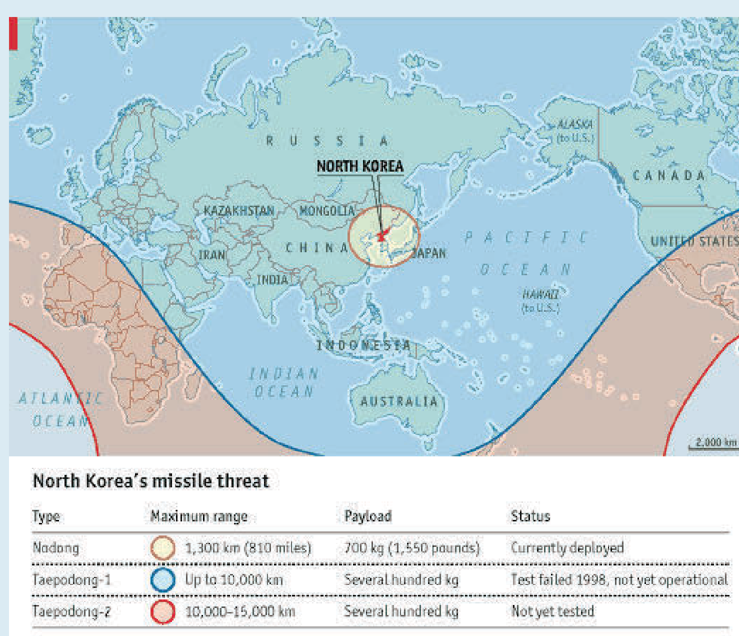
Spatial statistics. Which areas of a silicon wafer have an unusually high concentration of defects? Has there been an outbreak of disease? Where? In 1854, Dr. John Snow manually plotted Cholera locations on a street map of London to visually identify the outbreak hotspot around the Broad Street water pump (see Figure 2a). It took several days to perform this analysis for even a single disease over a small geographic area. Today, public-health agencies monitor scores of infectious diseases over very large geographic areas through spatial statistical tests (Figure 2b) designed to detect outbreaks (such as scan statistics) and hotspots, as well as distinguish these events from natural variations. Spatial statistical techniques are also routinely used in public safety (such as hotspots of crime reports), VLSI circuit design (such as defect hotspots on silicon wafers), weather forecasting (such as data assimilation), transportation (such as accident hotspots), mining (such as

Figure 1. Geographic information system. A 2003 article in *The Economist* significantly underestimated the distance North Korean missiles could travel because its map did not account for the spherical shape of the Earth; the correct version is below.[9]

(a) Flat Earth



(a) Spherical Earth



Kriging), public health (such as cancer cluster detection), and agriculture (such as designing management zones for precision agriculture and sample design for agriculture census). Spatial statistical theories (such as point processes, spatial autocorrelation, and geostatistics) address unique challenges (such as violation of independent identical distribution assumption) in applying traditional statistical models (such as linear regression, pearson correlation coefficient) to geographic data. Although spatial statistical techniques are an order of magnitude more computational and data intensive than traditional statistical techniques, the increased availability in recent decades of inexpensive high-performance computing and data technologies (such as sensors, the spatial database management system, and GIS) has facilitated wider interest in and adoption of spatial statistical methods.[12]

Where am I on the surface of the Earth?

Recent Change

In the late 20th century, most maps were produced by a small group of highly trained people in government agencies and surveying companies. Organizations (such as the U.S. Department of Defense and oil exploration companies) used highly specialized software (such as Esri ArcGIS and Oracle Spatial) for editing and analyzing geographic information. As summarized in Table 2, recent advances in spatial computing have changed this situation dramatically. Users with cellphones and access to the Internet number in the billions, meaning virtually the entire planet uses spatial technologies. Their very success has raised users' expectations of spatial computing. At the same time, users increasingly worry about the potential misuse of location data.

Billions using location-based services and updating actual maps. The proliferation of Web-based technologies, cellphones, smartwatches, consumer GPS-devices, and location-based social media facilitates widespread use of location-based services,[48] and Internet services (such as Google Earth and OpenStreetMap) bring GIS to the masses. With cellphones and consumer GPS devices, services (such as Enhanced 911 and navigation applications) are used by billions of people. Uber, Waze, Google Maps, Facebook check-in and other location-based social media are also used by more than one billion people worldwide.

Billions functioning as mapmakers, and many phenomena being observable. The source of geodata is increasingly smartphone users who may actively or even passively contribute their own geographic information. The immediate effect is wider coverage and increased numbers of surveyors for all sorts of spatial data. More phenomena are becoming observable because sensors are getting richer for 3D mapping, while broader spectrums at finer resolutions are being captured.

Multiple location-aware platforms. Spatial computing support was traditionally limited to application software layers (such as ArcGIS), Web services (such as Google Maps and MapQuest), and database management (such as SQL3/ OGIS). In the past decade, spatial computing support has emerged at several levels of the computing stack, including HTML5, social media check-ins, Internet Protocol Version 6, and open location services.

Rising expectations due to vast potential and risk. Location-based services, navigation aids, and interactive maps arguably exceed user expectations. Their intuitive basis and ease of use have earned them a solid reputation. Consumers see the potential of spatial computing for reducing greenhouse-gas emissions, strengthening cybersecurity, improving consumer confidence, and otherwise addressing many other societal problems. However, the very success of spatial computing technologies also raises red flags among users. Geoprivacy concerns must thus be addressed to avoid spooking citizens, exposing economic entities to liability, and undermining public trust.

Short-Term Opportunities

The profound changes outlined here reflect emerging avenues of research in spatial computing and give rise to a number of exciting new opportunities.

Augmented reality systems. Augmented reality enriches our perception of the real world by overlaying spatially aligned media in real time; for example, it can alter a user's view of the environment by adding computer graphics to convey past, present, or future information about a place or object, as in

Figure 3 and Figure 4. It is already used in head-up displays in aircraft cockpits and has become a popular feature in smartphone applications. As lightweight, but powerful, computer-driven eyewear becomes more commonplace, augmented reality will play a more central role in medicine, architecture, tourism, commerce, engineering, civil/urban planning, and assembly and maintenance, as well as in general day-to-day intelligence amplification. New spatial computing research challenges in this area stem from the need for new algorithms, as well as cooperation between users and the cloud, full 3D position and orientation (pose) estimation of people and devices, and registration of physical and virtual things. What natural interfaces can be adapted to leverage all human senses (such as vision, hearing, and touch) and controls (such as thumbs, fingers, hands, legs, eyes, head, and torso) to interact with augmented reality across multiple tasks? How can technology capture human bodies with full degrees of freedom and represent them in virtual space?

Spatial predictive analytics. Progress in spatial statistics[45] and spatial data mining[51] over the past decade has the potential to improve the accuracy and timeliness of predictions about the future path of hurricanes, spread of infectious diseases, and traffic congestion. Such questions have confounded classical prediction methods[32,35] due to challenges like spatial autocorrelation, nonstationarity, and edge effects. Spatial models can be invaluable when making spatiotemporal predictions about a broad range of issues, including the location of probable tumor growth in a human body or the spread of cracks in aircraft wings or highway bridges. Questions that need to be answered in this research area include: How might machine-learning techniques[30] be generalized to address spatiotemporal challenges of autocorrelation, nonstationarity, heterogeneity, and multiscale? How can frequent spatiotemporal patterns be mined despite transaction-induced distortions (such as loss or double-counting of neighborhood relationships)? What scalable and numerically robust methods are available for computing the determinants of very large sparse (but not banded) matrices in the context

Figure 2. Analysis of water pump sites and deaths from cholera in London in 1854:[54] (a) pump sites and deaths; and (b) output of spatial statistical test.

Table 2. Recent changes in spatial computing.

Late 20th Century	21st Century and Beyond
Sophisticated groups (such as the Department of Defense and oil-exploration companies) used GIS technologies.	Billions of people use location-based services and update actual maps.
Highly trained people in government agencies and surveying companies produce maps.	Billions of people are mapmakers, and many phenomena are observable.
Only specialized software (such as ArcGIS and Oracle Spatial) could edit or analyze geographic information.	More and more platforms are becoming location aware.
User expectations were modest (such as to assist in producing and distributing paper maps and their electronic counterparts).	User expectations are increasing due to vast potential and risk.

of maximum likelihood parameter estimation for spatial autoregression modeling?

Geocollaborative systems, fleets, and crowds. Spatial computing promises to take the Internet beyond cyberspace to the location-aware Internet of Everything, enabling connections among fixed structures and moving objects (such as cars, pedestrians, and bicycles), helping coordinate movement and understand patterns of mobility in cities and beyond; for example, the city of Los Angeles in April 2013 interconnected all of its 4,500 traffic signals to improve traffic flow during rush hour. Spatial computing enables smartmobs (groups of people) to come together quickly for a common cause, reducing the need for any one person to lead; drivers, smartcars, and infrastructure may cooperate in the future to reduce congestion, speed evacuation, and enhance safety. This cooperation raises the challenge of "trust" while using a group of spatial agents for computation and decision making. How might geographically distributed agents (such as smart signals and cars) cooperate in a trustworthy manner, even in the face of GPS spoofing?

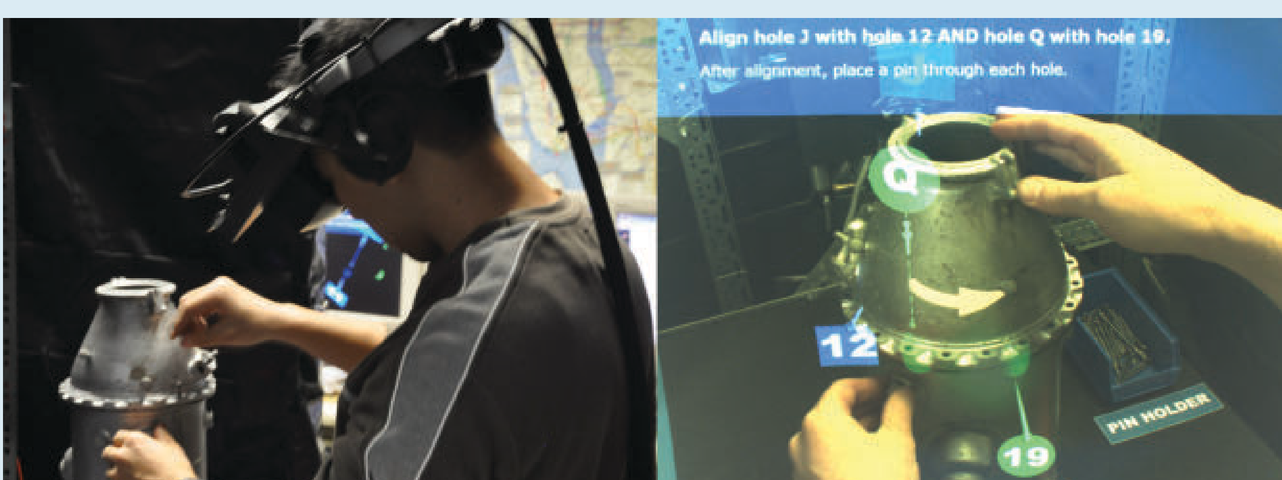
Moving spatial computing indoors, underwater, and underground. Despite worldwide availability, GPS signals are largely unavailable indoors, where we human beings spend 80% to 90% of our time.[56] Location-based services (such as route navigation) currently fill 10% to 20% of our time, but with emerging technologies like indoor localization, routing, and navigation (available in major airports and hospitals), the new expectation in the 21st century is our spatial context will be available essentially all the time, leveraging localization indoors and underground (such as mines and tunnels) through cellphone towers, Wi-Fi transmitters, and other indoor infrastructure. Indoor localization raises several new research questions, including: What scalable algorithms can create navigable maps for indoor space from CAD drawings? What about buildings where CAD drawings are not available? How can we perform reliable localization in indoor spaces where GPS signals might be attenuated or denied?

Figure 3. Augmented reality applications are becoming commonplace for smartphones.

Long-Term Research Needs

Spatial computing provides society tremendous value, but significant challenges are also emerging from those successes. Meeting them indeed requires expertise beyond the realm of spatial computing itself. First, overcoming the challenges of the public being de facto mapmakers and most phenomena being observable requires moving from the fusion of data from a few trusted sources to the synergy of data across a multitude of volunteers. Second, surmounting the challenge of

Figure 4. Experimental augmented reality assistance in aircraft-engine assembly. Henderson, S. and Feiner, S. Augmented reality in the psychomotor phase of a procedural task. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality* (2011).



equipping multiple platforms to be location aware will move spatial computing from a few platforms (such as cellphones) to almost all platforms (such as sensors, PCs, and clouds). Third, a greater understanding of human cognition is needed to ensure all members of society benefit from location-based services. And finally, spatial computing will have to address users' trust in and worry over privacy.

From fusion to synergetics. Historically, popular GIS software products (such as Esri's Arc family, PCI Geomatica, and ERDAS IMAGINE) were designed for geometric data (such as point, lines, and polygons) and raster data (such as satellite imagery). However, an ever-increasing volume of geographic data comes from volunteer citizens through check-ins, tweets, geotags, georeports from Ushahidi, and donated GPS tracks. Volunteered geographic information raises challenges related to data error, trustworthiness, and bias. The political and legal consequences of errors in spatial computing technology may be high; for example, after Hurricane Katrina in 2005, there was considerable concern in the U.S. Congress about the fact that the delays in releasing federal maps of New Orleans's most flood-prone neighborhoods had slowed rebuilding while yielding uncertainty.[15] Such political/legal complications could worsen in the future. Addressing such challenges requires a shift from traditional data-fusion ideas to a broader paradigm of data synergetics, raising yet more issues; for example, volunteers often use place names (such as Silicon Valley) and prepositions (such as near, in, at, and along) instead of numerical coordinates (such as latitude and longitude). Methods are thus needed for porting the current numerical-coordinate based data structures and algorithms to spatial data with place names and spatial prepositions. In addition, spatial and spatiotemporal computing standards are needed to more effectively use volunteered geographic information through quality-improvement processes (such as peer review and testing for recency) and documentation of quality measures (such as positional accuracy).

How do we serve societal needs (such as tracking infectious disease) while protecting individual geoprivacy?

From sensors to clouds. In the 20th century, the public face of spatial computing was represented by software (such as ArcGIS and Oracle Spatial Databases). Today, all levels of the computing stack in spatial systems are influenced by the fact that more and more platforms are location-aware due to the widespread use of smartphones and Web-based virtual globes. New infrastructure is needed to support spatial computing at lower layers of the computing stack so spatial data types and operations are appropriately allocated across hardware, assembly languages, operating system kernels, runtime systems, network stacks, database management systems, geographic information systems, and application programs. Augmented reality capabilities are needed to accommodate such devices as eyeglass displays and smartphones for automated, accurate, and scalable retrieval, recognition, and presentation of information. Sensing opportunities involve providing pervasive infrastructure for real-time centimeter-scale localization for emergency response, health management, and real-time situation awareness for water and energy distribution. Computational issues[5] raised by spatial big data mean new research opportunities for cloud computing addressing the size, variety, and update rate of spatial datasets that exceed the capacity of commonly used spatial computing technologies to learn, manage, and process data with reasonable effort.

Spatial cognition first. Spatial computing services were previously defined for only a small number of GIS-trained professionals who shared a specialized technical language not readily understood by the general public. With everyday citizens using location-based services while becoming the equivalent of mapmakers themselves, today there is an urgent need to understand the psychology of spatial cognition. Such understanding will improve the use and design of maps and other geographic information products by a large fraction of society. Further research on spatial cognitive assistance is needed to explore such ideas as landmark-based routing for individuals who cannot read maps or for navigating inside a new space (such as a building or campus) where not all areas (such as walkways) have names. Understanding group behavior in terms of participative planning (such as collaboration on landscape, bridge, and building design) or smart mobs for coordinating location movement will enhance spatial computing services for groups of people, as opposed to individuals. Context (such as who is tweeting, where they are, and physical features in the situation) should also be brought into these scenarios to investigate new opportunities for tweet interpretation for warning alerts during emergencies (such as natural disasters like Hurricane Sandy). New ways to understand our spatial abilities (such as navigation, learning spatial layouts, and reading maps) and the way different groups (such as drivers and pedestrians) think about space must be further investigated to leverage some of these opportunities: How do humans represent and learn cognitive maps? How could spatial-cognition concepts improve usability of spatial-computing services? How can we create user interfaces that bridge the gap between spatial computing "in the small," typically on indoor desktop systems with stereo displays and precise 3D tracking, and spatial computing "in the large," typically outdoors through coarse GNSS on mobile/wearable devices?

Geoprivacy. While location information (such as GPS in phones and cars) delivers great value to emergency-response personnel, consumers, and industry, streams of such data also introduce serious privacy and trustworthiness questions related to the use of geolocation and geosurveillance to monitor and control citizens, or sometimes called stalking, geo-slavery,[7] and geoprivacy;[18,34,41,42] for example, the European Union accused Google Street View (https://www.google.com/maps/streetview/) of privacy violations, leading it to suffer temporary bans in a number of countries. Striking a balance between utility and privacy remains a difficult challenge. Computer science efforts to obfuscate location information have largely yielded negative results to date. Many individuals thus hesitate to indulge in mobile commerce due to concern about privacy of their locations, trajectories, and other spatiotemporal personal information.[18] Computer scientists need to join forces with policymakers and other advocates to win consumer confidence. New legal principles must be devised to align with "fair information practices,"[42] especially those related to notice, transparency, consent, integrity, and accountability. However, such alignment also raises questions, including: What would be considered "adequate notice" for collecting spatial data? How should consent be requested? What information should be stored, and for how long? More broadly, when does localization (such as GPS tracking) lead to a privacy violation? Is reducing spatiotemporal resolution sufficient to discourage stalking and other forms of geoslavery? How do we serve societal needs (such as tracking infectious disease) while protecting individual geoprivacy?

Conclusion

Spatial computing promises an impressive array of opportunities for researchers and entrepreneurs alike in the coming decades. Successfully harnessing this potential will require significant intellectual investment and related funding of spatial computing research topics, including, but not limited to, the examples we explored here. Many spatial-computing projects today are too limited to achieve the critical mass needed for major steps forward. Benefactors must strongly consider funding larger and more adventurous efforts involving a dozen or more faculty groups across multiple universities. Some exemplary initiatives include the U.S. National Center of Geographic Information and Analysis, GEOmatics for Informed Decisions network in Canada, RGE in the Netherlands, and the Cooperative Research Centre for Spatial Information in Australia. Another barrier to progress in research has been the fact grant proposals are often reviewed by panels with few or no spatial-computing experts, sometimes resulting in a lack of champions. Funding agencies should thus consider special review panels and specialized requests for proposals.

A number of agencies have research initiatives in spatial computing,[23,25,26,28,29] including the National Cancer Institute's Spatial Uncertainty: Data, Modeling, and Communication initiative, the National Geospatial-Intelligence Agency's Academic Research Program, and the Chorochronos project[49] funded by the European Union. Given its cross-cutting reach, benefactors should aim to establish computer science leadership in this emerging area by creating a dedicated and enduring research program for spatial computing. Multiagency coordination to reduce competing projects and facilitate interdisciplinary, interagency research would benefit the entire field, as well as the agencies themselves.

Finally, spatial-computing scientists need more institutional support on their home campuses. Beyond large one-time grants, a few research universities have established GIS centers (akin to computer centers in the 1960s), as well as campuswide spatial initiatives (such as the Center for Spatial Studies at the University of California, Santa Barbara; http://spatial.ucsb.edu/, and U-Spatial at the University of Minnesota) that serve research endeavors across a range of disciplines, including climate change and public health. More research universities should follow their lead.

Spatial computing has proved itself a major economic opportunity for society, and further support for spatial computing research will ensure even more revolutionary advances to come.

Acknowledgments

We thank the participants of the Computing Community Consortium's 2012 Spatial Computing Visioning Workshop, with support from the National Science Foundation, especially the organizing committee members: Peggy Agouris, Michael F. Goodchild (member of the National Academy of Sciences), Erik Hoel, John Jensen (ASPRS, AACI), Craig A. Knoblock, Richard Langley (Institute of Navigation), Edward Mikhail (ASPRS), Ouri Wolfson, and May Yuan (former president of UCGIS). We thank the overall Computing Community Consortium, including Erwin Gianchandani, Kenneth Hines, Hank Korth, and Eric Horvitz for guidance and valuable feedback. We thank Michael Evans, Dev Oliver, Venkata Gunturi, and Reem Ali at the University of Minnesota for their help in Spatial Computing 2020 visioning activities. We also thank contributors to the Vision and Challenge paper track at the 12th International Symposium on Spatial and Temporal Databases.[40] And we thank Kim Koffolt for improving the readability of this article. Shashi Shekhar's research is supported by the National Science Foundation through grants 1029711, IIS-1320580, 0940818, and IIS-1218168, the U.S. Department of Defense through grants HM1582-08-1-0017 and HM0210-13-1-0005, and the University of Minnesota under OVPR U-Spatial. Steven K. Feiner's research is supported in part by the National Science Foundation through grant IIS-1514429. And Walid G. Aref's research is supported in part by the National Science Foundation through grant IIS 1117766. C

References

1. Ali, R.Y., Gunturi, V.M., Shekhar, S., Eldawy, A., Mokbel, M.F., Kotz, A.J., and Northrop, W.F. Future connected vehicles: Challenges and opportunities for spatiotemporal computing. In *Proceedings of the 23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, WA, Nov. 3–6). ACM Press, New York, 2015.
2. Auchincloss, A.H., Gebreab, S.Y., Mair, C., and Roux, A.V.D. A review of spatial methods in epidemiology, 2000–2010. *Annual Review of Public Health 33* (2012), 107.
3. Campbell, J.B. and Wynne, R.H. *Introduction to Remote Sensing, Fifth Edition.* The Guilford Press, New York, 2011.
4. Caron, C., Roche, S., Goyer, D., and Jaton, A. GIScience journals ranking and evaluation: An international Delphi study. *Transactions in GIS 12*, 3 (2008), 293–321.
5. De Berg, M., Van Kreveld, M., Overmars, M., and Schwarzkopf, O. *Computational Geometry.* Springer, Berlin Heidelberg, Germany, 1997.
6. Devillers, R., Jeansoulin, R., and Goodchild, M.F. *Fundamentals of Spatial Data Quality (Geographical Information Systems Series)*. Wiley-ISTE, New York, 2006.
7. Dobson, J. and Fisher, P. Geoslavery. *Technology and Society Magazine 22*, 1 (Spring 2003), 47–52.
8. Downs, R.M., Stea, D., and Boulding, K.E. *Image and Environment: Cognitive Mapping and Spatial Behavior.* Aldine Transaction, a division of Transaction Publishers, New Brunswick, NJ, and London, U.K., 2005.
9. Esri. *Understanding Geodesic Buffering*; http://www.esri.com/news/arcuser/0111/geodesic.html
10. Ester, M., Kriegel, H.-P. and Sander, J. Spatial data mining: A database approach. In *Advances in Spatial Databases, Volume 1262 of Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg, Germany, 1997, 47–66.
11. Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, K., and Krause, A. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks* (Chicago, IL, Apr. 12–14). IEEE Computer Society, Washington, D.C., 2011, 13–24.
12. Gelfand, A.E., Diggle, P., Guttorp, P., and Fuentes, M. *Handbook of Spatial Statistics. Chapman and Hall/CRC Handbooks of Modern Statistical Methods.* CRC Press, Boca Raton, FL, 2010.
13. Goodchild, M.F. Citizens as sensors: The world of volunteered geography. *GeoJournal 69*, 4 (Aug. 2007), 211–221.
14. Gore, A. The digital Earth: Understanding our planet in the 21st century. *Australian Surveyor 43*, 2 (1998), 89–91.
15. Jordan, L.J. Flood map delays spur angst for Gulf Coast, but it may be for nothing. *Djournal.com* (Mar. 26, 2006); http://goo.gl/IIzvzv
16. Kemp, K.K. *Encyclopedia of Geographic Information Science.* SAGE Publications, Thousand Oaks, CA, 2007.
17. Krause, A., Horvitz, E., Kansal, A., and Zhao, F. Toward community sensing. In *Proceedings of the Seventh International Conference on Information Processing in Sensor Networks* (St. Louis, MO, Apr. 22–24). IEEE Computer Society, Washington, D.C., 2008, 481–492.
18. Krumm, J. A survey of computational location privacy. *Personal and Ubiquitous Computing 13*, 6 (Aug. 2009), 391–399.
19. Kumar, V., Shekhar, S. and Fagmous, J., Eds. Computing & climate. *Computing in Science and Engineering 17*, 6 (Nov./Dec. 2015).
20. Lovell, J. Left-hand-turn elimination. *The New York Times* (Dec. 9, 2007).
21. Manyika, J., Institute, M.G., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C, and Byers, A.H. *Big Data: The Next Frontier for Innovation, Competition, and Productivity.* McKinsey Global Institute, 2011; http://goo.gl/BWJPqP
22. McMaster, R.B. and Usery, E.L. *A Research Agenda for Geographic Information Science.* CRC Press, Boca Raton, FL, 2004.
23. National Academies of Sciences, Engineering, and Medicine. *IT Roadmap to a Geospatial Future. Report of the Committee on Intersections Between Geospatial Information and Information Technology.* The National Academies Press, Washington, D.C., 2003.
24. National Academies of Sciences, Engineering, and Medicine. *Global Navigation Satellite Systems: Report of a Joint Workshop of the National Academy of Engineering and the Chinese Academy of Engineering.* The National Academies Press, Washington, D.C., 2012.
25. National Academies of Sciences, Engineering, and Medicine. *Priorities for GEOINT Research at the National Geospatial-Intelligence Agency.* The National Academies Press, Washington, D.C., 2006.
26. National Academies of Sciences, Engineering, and Medicine. *Successful Response Starts with a Map: Improving Geospatial Support for Disaster Management.* The National Academies Press, Washington, D.C., 2007.
27. National Academies of Sciences, Engineering, and Medicine. Figure 3.1: The precision of current geodetic applications as a function of the required time interval. In *Precise Geodetic Infrastructure: National Requirements for a Shared Resource.* The National Academies Press, Washington, D.C., 2010.
28. National Academies of Sciences, Engineering, and Medicine. *New Research Directions for the National Geospatial-Intelligence Agency.* The National Academies Press, Washington, D.C., 2010.
29. National Academies of Sciences, Engineering, and Medicine. *Understanding the Changing Planet: Strategic Directions for the Geographical Sciences.* The National Academies Press, Washington, D.C., 2010.
30. National Academies of Sciences, Engineering, and Medicine. *Frontiers in Massive Data Analysis.* The National Academies Press, Washington, D.C., 2013.
31. National Academies of Sciences, Engineering, and Medicine. *Continuity of NASA Earth Observation from Space: A Value Framework* (prepublication draft). The National Academies Press, Washington, D.C, 2015; http://www.nap.edu/read/21789/chapter/1
32. National Academies of Sciences, Engineering, and Medicine. Land change modeling approaches. In *Advancing Land Change Modeling: Opportunities and Research Requirements.* The National Academies Press, Washington, D.C., 2014.
33. National Academies of Sciences, Engineering, and Medicine: Committee on Human and Environmental Exposure Science in the 21st Century. *Exposure Science in the 21st Century: A Vision and a Strategy.* The National Academies Press, Washington, D.C., 2012.
34. National Academies of Sciences, Engineering, and Medicine: Panel on Confidentiality Issues Arising from the Integration of Remotely Sensed and Self-Identifying Data. *Putting People on the Map: Protecting Confidentiality with Linked Social-Spatial Data.* The National Academies Press, Washington, D.C., 2007.
35. National Science Foundation. *Report of the NSF Workshop on Intelligent Systems Research to Support Geosciences and the EarthCube Mission.* Arlington, VA, Mar. 2015; http://goo.gl/MyBsQY (workshop report summary http://is-geo.org/report/)
36. National Science Foundation. *An NSF Workshop to Identify Interdisciplinary Data Science Approaches and Challenges to Enhance Understanding of Interactions of Food Systems with Energy and Water Systems.* The National Institute of Food and Agriculture, Washington, D.C., Oct. 2015; http://goo.gl/GCQ2fj
37. Okolloh, O. Ushahidi, *or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Participatory Learning and Action 59*, 1 (May 2009), 65–70.
38. Open GIS Consortium. *Open GIS Simple Features for SQL*; http://www.opengis.org
39. Parkinson, B.W. and Spilker, J.J. *Global Positioning System: Theory and Applications, Volume One.* The American Institute of Aeronautics and Astronautics, Reston, VA, 1996.
40. Pfoser, D., Tao, Y., Mouratidis, K., Nascimento, M.A., Mobel, M., Shekhar, S., and Huang, Y., Eds. Challenge and Vision Track. *In the 12th International Symposium on Spatial and Temporal Databases, Volume 6849 of Lecture Notes on Computer Science* (Minneapolis, MN, Aug. 24–26). Springer, Berlin, Heidelberg, Germany, 2011; http://sstd2011.cs.umn.edu/
41. Pomfret, K. *Latitudes and Attitudes: Zooming In on Geospatial Data, Privacy, and the Law in the Digital Age.* Centre for Spatial Law and Policy, Richmond, VA, Mar. 27, 2013; http://www.spatiallaw.com/Uploads/Latitudes_and_Attitudes.pdf
42. Pomfret, K.D. *Geotargeted Alerts: Potential Impact of Privacy Concerns.* The National Academy of Sciences, Washington D.C., Feb. 21, 2013.
43. Regalado, A. New Google Earth Engine. *ScienceInsider* (Dec. 3, 2010); http://news.sciencemag.org/technology/2010/12/new-google-earth-engine
44. Richmond, R. Digital help for Haiti. *The New York Times* (Jan. 27 2010).
45. Ripley, B.D. *Spatial Statistics, Volume 575.* John Wiley & Sons, Hoboken, NJ, 2005.
46. Samet, H. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley Series in Computer Science, Reading, MA, 1989.
47. Samet, H. *Foundations of Multidimensional and Metric Data Structures.* The Morgan Kaufmann Series in Computer Graphics, San Francisco, CA, 2006.
48. Schiller, J. and Voisard, A. *Location-Based Services.* Elsevier, San Francisco, CA, 2004.
49. Sellis, T. et al., Eds. *Spatio-Temporal Databases: The CHOROCHRONOS Approach. Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, 2003.
50. Shekhar, S. and Chawla, S. *Spatial Databases: A Tour.* Prentice Hall, Upper Saddle River, NJ, 2003.
51. Shekhar, S., Evans, M.R., Kang, J.M., and Mohan, P. Identifying patterns in spatial information: A survey of methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1*, 3 (May/June 2011), 193–214.
52. Shekhar, S., Feiner, S., and Aref, W. From GPS and virtual globes to spatial computing—2020. *GeoInformatica 19*, 4 (Oct. 2015), 799–832.
53. Shekhar, S. and Xiong, H. *Encyclopedia of GIS.* Springer, New York, 2008.
54. Snow, J. *On the Mode of Communication of Cholera.* John Churchill, London, England, 1855.
55. Sobel, D. *Longitude.* Walker Books, London, U.K., 2010.
56. U.S. EPA Office of Air and Radiation. *The Inside Story: A Guide to Indoor Air Quality. Technical Report EPA 402-K-93-007.* U.S. Environmental Protection Agency, Washington, D.C.; http://goo.gl/RkZUBU
57. Uttal, D.H. and Cohen, C.A. Spatial thinking and STEM education: When, why, and how. *Psychology of Learning and Motivation 57*, 2 (2012), 147–181.

Shashi Shekhar (shekhar@cs.umn.edu) is a McKnight Distinguished University Professor in the Department of Computer Science at the University of Minnesota and a former member of the Computing Community Consortium Council.

Steven K. Feiner (feiner@cs.columbia.edu) is a professor of computer science and the director of the Computer Graphics and User Interfaces Lab at Columbia University, New York.

Walid G. Aref (aref@cs.purdue.edu) is a professor of computer science at Purdue University, West Lafayette, IN, and a past chair of ACM SIGSPATIAL (http://www.sigspatial.org).

Watch the authors discuss their work in this exclusive *Communications* video. http://cacm.acm.org/videos/spatial-computing

DOI:10.1145/2756542

Developers first need compelling incentives and committed management.

BY MELISSA LEE, ESTEVE ALMIRALL, AND JONATHAN WAREHAM

Open Data and Civic Apps:

First-Generation Failures, Second-Generation Improvements

ON HIS FIRST day in office in 2009, U.S. President Barack Obama signed the "Memorandum on Transparency and Open Government," asking government agencies to make their data open and available to the public.[4] The aim was to provide transparency in government and improve provision of services through new technologies developed on the backbone of civic open data.[5] Transparency was achieved through a public data catalog that was the most comprehensive at the time, providing such information as real-time crime feeds, school test scores, and air-quality metrics. However, as of May 2010, only one year later, few citizens had make the effort to comb through the more than 272,000 datasets they had been provided.[6]

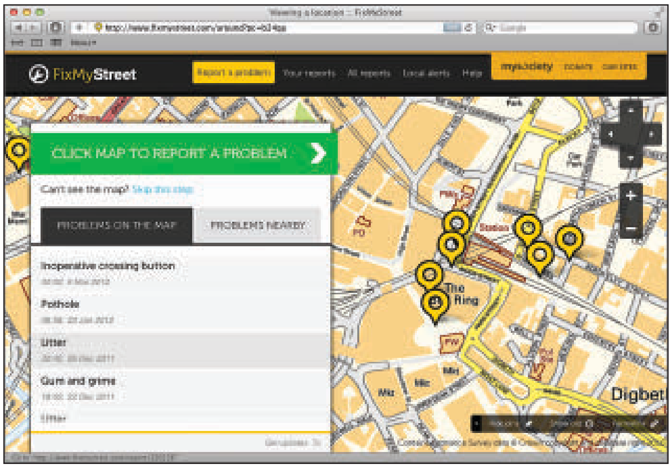
In response, leaders of the open data movement sought to engage code developers to make the information not only more digestible for greater transparency but also incorporate it into applications, services, and businesses that could better serve the public and foster economic growth.

U.S. chief information officer Vivek Kundra led the effort, enlisting the help of iStrategyLabs (https://isl.co/), a digital creative agency based in Washington, D.C. To spur interest in the data.gov repository, iStrategyLabs then launched the "Apps for Democracy" contest with cash prizes to stimulate development of civic apps. With an investment of only $50,000 provided for prize-winning solutions, 47 apps were created with an estimated value of $2.3 million based on the cost to develop them through more traditional means.[1] Moreover, the 30-day contest significantly compressed the amount of time otherwise needed to launch the government down this innovative path, estimated at two years through conventional methods. The strategy was thus deemed a success; New York and San Francisco soon followed with similar contests. Indeed, as momentum increased in the open data movement, cities, rather than the federal government, took control of publishing and promoting open data initiatives. In the following three years, these strategies were replicated in cities worldwide.

However, by 2011, much of the initial enthusiasm behind the open data

» key insights

- **A study of eight civic innovation ecosystems identified the challenges cities face in managing more diverse groups of collaborators than private firms face.**
- **First-generation initiatives lacked measurable civic benefit, as they suffered from loose governance and limited knowledge transfer.**
- **Second-generation strategies improved through stronger management and consideration of the motivations of external collaborators.**



Example map on a FixMyStreet webpage; and Street Bump mobile app on smartphone screens.

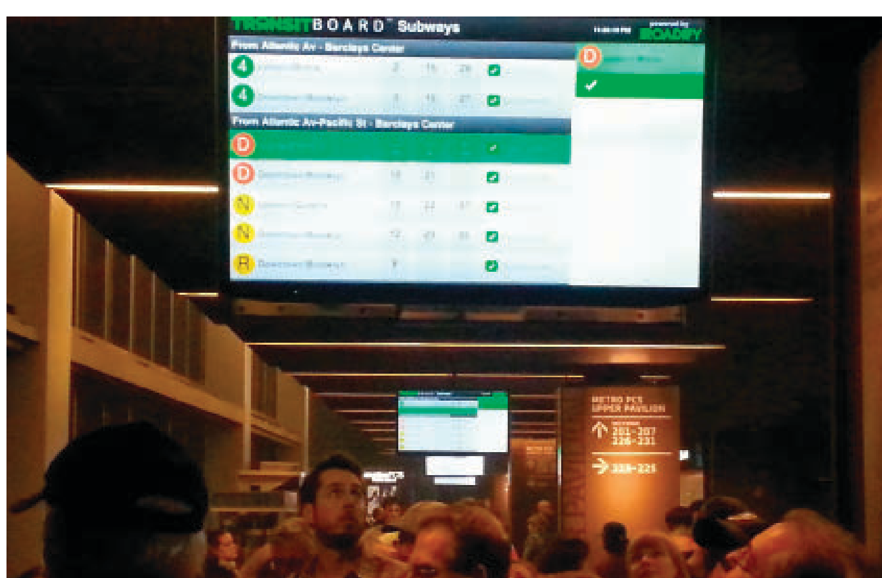
movement had waned. The adoption, impact, and value creation of apps developed through open civic data was far less than anticipated. The open data repository was accessed through downloads of more than two million datasets, though few applications based on the data were widely used, nor did they have high quality ratings;[2] for instance, none of them appeared in the top 100 overall applications in either the Apple or Android appstores. While there is a huge potential market for civic apps, these initiatives had failed to create the social or economic value that was projected.

In this article, we examine early strategies behind the open data movement. We interviewed application developers and civic organizers in eight cities in the U.S. and Europe: Amsterdam, Barcelona, Berlin, Boston, Helsinki, New York, Philadelphia, and Rome. Through the course of these interviews we tried to identify some of the reasons the initiatives failed to meet expectations. We conclude by examining more recent adaptations to the strategies that offer pathways toward greater civic benefit.

Development Contests

Bold vision, meager results. Following the apparent success of the Apps for Democracy contest in 2009, cities worldwide began hosting application contests to capitalize on their own newly open data catalogs. These contests continue to be the predominant strategy for fostering transparency and economic development provided through civic open data. However, early initiatives suffered from a lack of civic benefit, in both government and the public. Though efforts were made to open data throughout all types of government, developers tended to incorporate only a small range of it, including overuse of certain datasets. A multitude of apps targeted similar solution categories (such as transportation and mobility) with only limited use or civic benefit. Organizers began to realize neither data quality nor general interest was the cause of the meager impact. Rather, limited public knowledge of the significant operational challenges facing city governments generated a portfolio of somewhat anemic apps targeting a predominantly consumer market. Developers, with similar social demographics, were guided by personal experience or interest to develop apps centered on restaurants, parks, and public transportation. Betsy Scherzer, an organizer of New York City's Big Apps Contest, explained it like this: "I think a lot of it depends on what developers are interested in and what seems useful. For example, we get a lot of data from the Office of Management and Budget. That data does not match or lend itself easily to apps. Not too many people want a municipal budget app. Whereas the Parks Department, which has all the info on park Wi-Fi and stuff you can see, pull out your phone and use the info; those datasets get used first."

Even within the datasets that did receive attention, developers often failed to envision solutions that would greatly complement provision of municipal services. Tourism apps, for example, represented almost 12% of the apps in Amsterdam's 2013 "Apps for Amsterdam" contest (http://www.appsforamsterdam.nl/), but the utility of the solutions was anchored in mobility and consumption, not increased ser-



Publicly available Roadify display at Barclays Center arena, Brooklyn, NY.

vice provided by the city. Applications that had real impact for citizens or government were few. The app DontEat.at (http://www.donteat.at/) is an exception, demonstrating better integration of open data and civic services. DontEat.at was created as part of the New York Big Apps competition, integrating restaurant-health-inspection information provided by the New York City Department of Sanitation with restaurant location and ratings data. Upon entering an eatery, DontEat.at would recognize the locale and determine its inspection status. If that restaurant had been flagged for a sanitation- or heath-inspection violation, the app would send patrons a text message alerting them to the notice. In addition to providing a service to citizens that would greatly affect their actions, the app also affected the role of the Department of Sanitation. Previously, health inspections would go virtually unnoticed until egregious and final violations called for public notices and restaurant closure. DontEat.at reinforced even minor violations by making the public more aware of infractions. Health inspectors began to see cleanup more quickly, and without repeated visits because patrons were leaving after receiving the alerts. The app demonstrates time and cost savings civic apps can provide a city and its citizens, though few apps developed through the contests delivered such civic benefit.

Apps developed in city-sponsored contests failed to have an impact because developers often arrived with ready-made solutions. Contest organizers hoped the range of datasets would spur new and innovative apps to improve internal city processes, provide better civic services, or facilitate government-to-citizen interaction. But because the requirement for contest participation was often the simple inclusion of a city-provided open dataset, most developers submitted previously developed apps with minor adjustments to accommodate civic data. So even where numerous recycled apps exploited civic datasets, novel business innovations or improvements in the provision of civic services were rare.

So even where numerous recycled apps exploited civic datasets, novel business innovations or improvements in the provision of civic services were rare.

Failure to provide value capture. In addition to lack of impact, the open data initiatives were not managed in ways to guarantee value capture. Contest organizers did not fully understand the motivations of external participants to ensure their continued involvement, nor did they expect real savings to be accounted for in city hall. Initially, contest organizers reasoned prize money would be a strong motivator for developer participation, providing a foundation for them to jump-start and sustain development of their apps. Some contests included tens of thousands of dollars for prize winners. However, though money was never refused, most developers believed the amount was not enough to provide application support, maintenance, and sustainability over time. They were instead looking for much more. As Jonathan McKinney of Cab Corner, an app providing a cab-sharing utility, said, "Our reason for participating is to be recognized enough to get serious funding. Not $10,000 or $20,000, but someone who will give you a quarter of a million dollars or so and really get involved and bring more people in. The prize money is not a game changer. The real reward is when someone calls you out of the blue and says they have real venture capital for you. Then you can get things done."

Developers did not chase the prize money but rather participated as they would any non-city-sponsored contests—for exposure, reputation, and evaluation. Coders sought exposure to potential funders that, unlike one-time winnings, would represent a sustained source of income for those looking to start a business from their apps.

As contest organizers became more aware of developer motivations, greater effort was made to include entrepreneurs and venture capitalists on the panels of judges. They also hosted events and closing ceremonies that included potential funders, and some developers found success through this model. MyCityWay (http://www.mycityway.com/) was an app and platform developed to allow businesses to connect to their customers in real time when on the move throughout a city. MyCityWay's exposure in New York City's Big Apps Contest won its developers more than $7 million in venture capital. News of that success spread through the developer community, increasing participation by others looking for funding through contest exposure. However, MyCityWay was an exception, and, overall, developers could not expect such funding to be the norm in city-sponsored contests. This left them struggling with financial constraints that often led them to abandon their apps.

Aside from external funding, participants still hoped to capture value through contest participation—not to potential investors but to a larger citizen market for their apps. Developers hoped citizens would become aware of the civic apps through municipal websites or the city organizers to showcase participating solutions. However, such efforts fell short of expectations. Marco Cavalli, a developer in the Apps for Italy contest, said, "If only we had more exposure leading to more users who eventually paid for the premium version. We hoped to get more subscribers just to start so that we could continue with our development. But without more initial awareness through the city or other advertising, we were not able to grow."

Moreover, cities did little to advertise their new apps collections, and, not surprisingly, citizens did not flock to city websites to discover them. The usual outlets for finding apps—the Apple and Android app stores—do not feature categories highlighting city apps, making it difficult to gain awareness in the largest markets. Instead, creating awareness was left mostly to the developers, who found it difficult without additional funding. Though the market for city services remains more than enough to provide continued value to thousands of civic apps, actual adoption remains low and fails to sustain development.

Failures within government. Failures in early open data challenges also stemmed from internal issues within city governments and expectations of participating departments. The first step in these initiatives involved persuading agencies to open their data and provide it in usable formats. With strained budgets, overworked employees, and other, more critical responsibilities expected on a daily basis, releasing data was not only a chore with no tangible benefit but also subjected municipal departments to unwanted scrutiny. Employee reluctance delayed city halls in opening data repositories to the public. Most cities eventually introduced legislation to force data publication, but their departments were still slow to move.

Moreover, the managing department for most open data contests was usually the innovation, IT, or economic development department. Beyond data publication, the managing department had little interaction with more core city agencies regarding the apps challenges championed by the organizational periphery. This disconnect between city operations and open data initiatives greatly hampered their potential success.

Involvement by civic departments directly requesting specific solutions beneficial to city operations was sometimes prohibited by procurement legislation. Betsy Scherzer from New York's Department of Economic Development said, "We had a few agencies that came to us and said, 'We are from the Department of X and we would love to have the following guide made for us that does XYZ.' But that's actually a specific enough request that it would be considered something you would have to procure for, and so we're not allowed to accept them, because, if we did, it would be like procuring something for free."

Not only were agencies prohibited from requesting focused solutions, general communication between the relevant departments and developers was limited. If city departments were stifled in the development phases, their potential for adoption or support further into the app life cycle was highly unlikely. There were no instances of popular or useful apps being adopted or partially managed by a city agency. As such, civic apps suffered because the departments for which they were created failed to integrate them into the central services provided by the city (see Table 1).

First-generation failures. Because management of open data initiatives was handled outside core city departments, these agencies were not asked to make financial investment in the solutions. Likewise, accountability for the impact of the open data and the success of the resulting apps was also dispersed. Managers did not expect dramatic returns from the contests, especially in terms of savings that might directly accrue to their department. Central organizers sought to quantify the value saved by a contest with metrics measuring the comparable cost of in-house development. But, as these savings were not accounted for in any departmental budget, there were no reviews or measurements of actual benefits. Instead, the rationale provided for contests became focused outside city hall on economic development within the community, stemming from new businesses based on the apps. Not surprisingly, few sustainable businesses have materialized. The number of participants, datasets opened, and apps developed have become the metrics by which contests are evaluated. However, these numbers poorly reflect municipal savings or entrepreneurial or social value.

Second-Generation Initiatives

As open data initiatives continued to gain popularity, cities and developers began to recognize which strategies were most effective and how to improve on others. Though many of the initial efforts continue, some second-generation initiatives incorporate new mechanisms and include additional actors to increase the impact of civic open data and provide value capture

Table 1. First-generation failures.

What went wrong?
Excessive use of popular datasets
Overcrowding, with numerous similar apps in the same solution space
Apps from developers with homogeneous interests and demographics
Data published with no commensurate changes in city services
Preexisting apps tweaked for inclusion in coding contests
Prize money symbolic, insufficient for long-term sustainable app operations
Limited adoption and support by governments, with city involvement ending at data publication
Resistance to data transparency by public administrations

for those involved. These improvements represent some best practices and lessons to encourage the momentum behind the open data movement.

Increased exposure to civic needs. Early challenges often lacked impact because developers had limited experience with the full suite of civic services and instead created an abundance of solutions with what they thought would be popular consumer appeal. In order to redirect developer focus, organizers sought to educate developers about struggles in government or the plight of other citizen groups. "Hack-at-Home" is a strategy that exemplifies the improvements built into apps contests to enlighten developers about the need and potential for solutions.

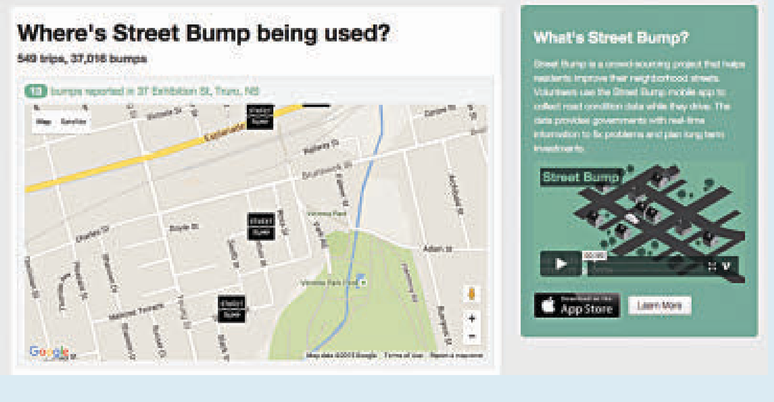
Hack-at-Home is an apps-contest model developed by DotOpen (http://dotopen.com/), an open innovation and digital media company based in Barcelona, Spain, giving developers more information about problems that could be better addressed through open data solutions by increasing the involvement of civic agencies early on. Instead of simply requesting governments' open data, DotOpen works closely with departments needing solutions to formulate the issues relevant and solvable through information and apps. The result, in addition to data repositories, is developers are presented "problem statements," or short descriptions including a "crisis statement" on the current situation or process that is failing; a "needs statement" on, generally, what utility an app would provide, without specifically detailing a developed solution; and an "impact statement" on the expected outcome and benefit the solution would provide to citizens and government alike. These 500-to-1,000-word outlines add significant impact simply by guiding developer attention to problems genuinely faced by governments. Apps developed through these challenges have, for example, increased awareness of sanitation problems while educating citizens about access to available resources and solutions.[7]

Another method for increasing the impact of open data also involves working with intermediaries to better educate developers about the situation faced in city hall.[3] This strategy, developed by nonprofit organization Code for America (http://www.codeforamerica.org/), abandons the contest model and greatly enhances the direct relationship between coders and civil servants. Code for America chooses approximately 30 developers and eight to 10 cities per year to create solutions based on open civic data. These developers must make a full-time commitment to Code for America for an 11-month period during which they relocate to San Francisco, CA. Developers engage directly with relevant city workers to better understand civic needs from their perspectives, as well as engage with citizens affected by the related problems. This model has also spread internationally through Code for Europe, Code for Africa, Code for the Caribbean, and others.

Stronger management. Second-generation open data initiatives have also increased civic benefit through better and stronger management. Where simple contest-driven strategies were disappointing in the first generation, the increased involvement of internal agencies and external partners has yielded superior results in the second wave. Boston's Office of New Urban Mechanics (NUM) is an example of an internal agency with strong management of its open data initiatives. NUM is an internal innovation department within the mayor's office, creating technological solutions that increase provision of civic services and value to government. NUM invites all actors to report their needs and suggestions for improvement, including citizens, government employees, academic institutions, nonprofit organizations, and private businesses. NUM then evaluates them based on their potential for improving civic services, filtering them on targeted areas like urban development and education. NUM follows a five-to-seven-month timeline for development of solutions, whether for a mobile app or more complete business based on the technological solution. This model of top-down management, unlike the early apps contests, has demonstrated lasting civic benefit, value capture, and solution sustainability.

One example of a NUM-developed app is Street Bump, which collects data about road conditions as users drive the roads (http://www.streetbump.org/). The city aggregates the data on real-time road deficiencies that can be fixed more quickly, reducing the cost of deploying civil servants to comb the streets for places needing repair. However, the app's success would not have been possible without NUM's involvement. Expertise was needed to develop a solution with an algorithm sophisticated enough to translate data from smartphones into physical bumps on a street (see Figure 1). NUM partnered with software company Connected Bits and design company IDEO to come up with the innovative product, an example of sophisticated collaborations that can have real impact in a city.

Figure 1. Application homepage showing cities where Street Bump is deployed and number of bumps recorded, as of November 2015.



Common platforms. The market for civic apps is virtually limitless, as civic needs are shared across municipal, regional, and national borders. However, most apps are designed for specific cities. This problem is due mainly to managers within government choosing to procure their solutions, whether developed in-house or through open innovation initiatives, as custom-tailored for their own cities. They imagine their needs are unique and want to showcase equally bespoke solutions. Yet starting from scratch takes time and resources well beyond those needed to adapt existing apps. And targeting software for a specific city decreases the potential commercial market available for the app. Small cities, in particular, lack populations large enough to support a community of civic app developers on their own, let alone justify investment in redundant functionality offered through existing software.

Application repositories, or marketplaces, provide a venue for civil servants or developers for sourcing existing solutions. The Civic Commons is such a marketplace, created to facilitate code sharing (http://theciviccommons.com/). This collection of civic apps promotes their use and reuse, providing value capture for developers as their markets increase and savings for cities as they choose to adapt, rather than create, completely new solutions. Other repositories have been developed (such as Europe Commons; http://commonsforeurope.net/) that showcase civic apps and offer best practices and case studies to provide more value capture to developers and savings to cities.

FixMyStreet is a solution hosted on Civic Commons that exemplifies the benefits of city-sharing and modularized solutions.[8] Developed by MySociety (https://www.mysociety.org/), a U.K.-based charity promoting e-democracy, FixMyStreet was originally developed to allow U.K. citizens to monitor and report street and road problems to their local councils. Recognizing its potential universal applicability, MySociety developed the solution as an easy-to-adapt platform for others. The FixMyStreet website (https://www.fixmystreet.com/) provides simple instructions for citizens looking to implement the solution locally.

FixMyStreet makes a case not only for city sharing and modularization but demonstrates the potential for real bottom-up, citizen-led impact. Unlike government-led initiatives, FixMyStreet requires few resources from city governments to be enabled. Citizens interested in hosting FixMyStreet in their locality need only the email addresses of civil servants or departments responsible for the issues on which a local citizen might report. A greatly enhanced channel of communication between cities and their constituents is thus created through citizens being able to adapt the platform or recruit others with the basic technical skills to customize the code and run the site. FixMyStreet has been used to report broken streetlamps, potholes, garbage collection, and even crime. It has been implemented in more than 15 counties.

With strained budgets, overworked employees, and other, more critical responsibilities expected on a daily basis, releasing data was not only a chore with no tangible benefit but also subjected municipal departments to unwanted scrutiny.

FixMyStreet also provides an easy-to-implement platform, with the same functionality and customization, along with training, maintenance, support, and Web and mobile app options. Average installation cost to a metropolitan area in the U.K is £15,000, with an annual maintenance and support fee averaging £2,500.[9] FixMyStreet demonstrates how an adapted, modularized solution can provide benefits at relatively modest cost. U.K. councils reported up to a 300% shift from phone calls to online reporting following integration. FixMyStreet also reaches a new demographic that would have been less likely to report through traditional channels. Further, its customization allows some cities (such as Zurich, Switzerland) to respond directly to each citizen report and track its progress through completion.

OpenStreetMap is another open data platform that crowdsources the original content rather than working from city-provided open data. Frustrated by the restrictions on proprietary map data yet inspired by the success of Wikipedia, Steve Coast, a British entrepreneur, developed OpenStreetMap in the U.K. in 2004[10] to encourage its more than 2.2 million registered users (as of August 2015) to contribute, augment, and edit geographical map data. However, OpenStreetMap's greatest value is not the output of a crowdsourced map but an open data platform from which other applications, including FixMyStreet, can source their map data. Initiatives like FIxMyStreet and OpenStreetMap show how engaged communities, and open, crowdsourced content repositories can support civic app development.

Data standardization among cities is an area in need of improvement, restricting the potential market for a given developer's app and limiting

Figure 2. Civic application development life cycle.

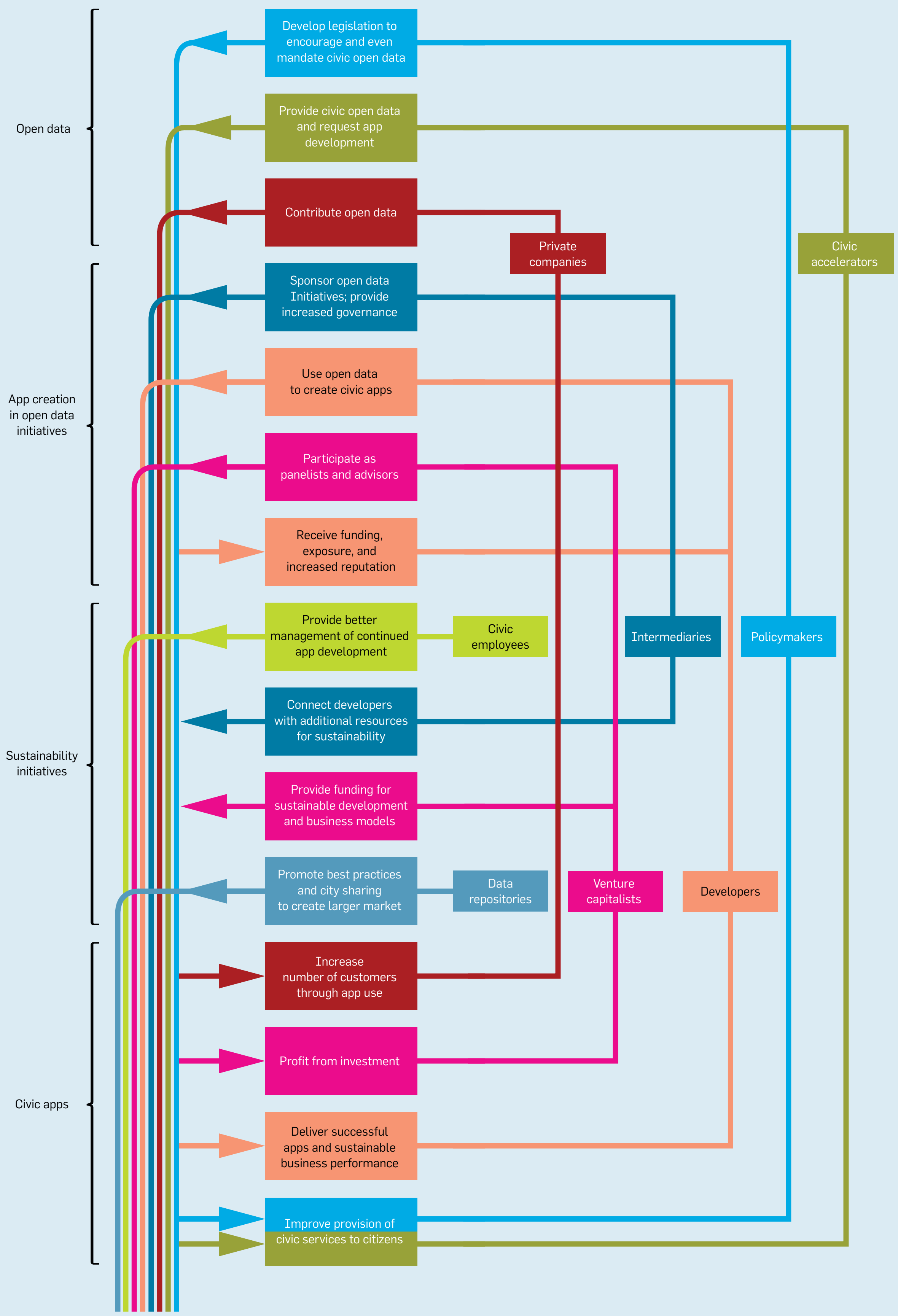


Table 2. Second-generation improvements.

What was learned?
Entrepreneurs and venture capitalists invited to judging panels to court funding opportunities
Legislation to force civic bodies to publish data in a timely manner
Problem statements published by cities to direct developer attention toward significant operational challenges
Developers embedded in city organizations for substantial time periods to better understand operations and build reciprocal engagement
Stronger management and direct coordination by city administrations
Ex-ante commitments for financial support of specific apps
Common app and crowdsourced data repositories, along with bottom-up, engaged communities
Open source coding practices and coordination of data standardization

potential value capture. Progress has been made, however, especially considering the early efforts of open data were in the form of static .pdf files published on a city's website. Civic departments have caught up and are adopting standards established by the World Wide Web Consortium (W3C; http://www.w3.org/) promoting the semantic Web and linked data, allowing not only machine-readable formats but information to be connected, queried, and shared more easily.[11] The data can then be used and collated across borders in ways envisioned by developers, independent of the original structure or intention of the data provider. However, until these standards are more universal, coders must write numerous interfaces for each city and maintain them individually.

The transportation app Roadify, for example, provides transit schedules for New York commuters (http://www.roadify.com/). Interested in increasing app adoption, the developers realized other cities would need the same information provided through local open data repositories. However, co-creator Dylan Goelz said, "The trouble is that data is provided differently in every market. Google tried to standardize the data, but there are still discrepancies. San Francisco may do something that Boston doesn't, and it makes aggregating the data difficult. We had to develop our own solutions to be able to shift and adapt, which has cost time and money."

As most city managers do not yet realize the benefit of sharing apps among cities, they also fail to understand government databases can grow beyond a city's borders (see Table 2). Data standardization requires coordination and procedural changes that are both technical and political. W3C standards greatly enhance developers' potential to more readily integrate information from multiple cities. Such efforts promote standardization and not only further sustain the solutions but also leverage network effects toward greater developer participation and user adoption (see Figure 2).

Conclusion

Momentum continues behind open data and its potential to provide cost savings to cities and better service to citizens. Early efforts focusing on application contests with low governance failed to produce the results most cities hoped for, though they did provide insights into potential fixes. Second-generation initiatives have incorporated better management and knowledge transfer to increase value capture and impact. Bottom-up initiatives, crowdsourced content, and shared open data repositories and apps also support these efforts.

The main problems involve mechanism coordination. Progress thus needs to continue toward standardization of data formats and APIs to allow effective sharing in app markets. Application discovery remains problematic, as there are no effective discovery and diffusion channels beyond the most popular 100 apps. Also needed is efficient code reuse among public organizations that would allow not only better use of taxpayers' money but leverage network effects toward incremental and cumulative innovation.

Incentive management for all actors in such heterogeneous ecosystems is certainly more complex than in traditional markets. Three main problems remain. First, market fragmentation renders standard business models based on advertising or usage fees impractical, forcing app developers to resort to reputation or signaling as alternative modes of value capture. Second, trust is needed in the stability, continuity, and availability of open data streams and APIs that are not always secure in politically turbulent municipalities. And third, the inherent tension between collaboration and competition represents a managerial challenge in these complex and diverse ecosystems.

Open data strategies in the public sector should continue to evolve, and, with continued ingenuity, greater efficacy, impact, and social value. What open data and civic-app contest designers have learned is not special to the world of government data but extendible to other spheres of distributed, collective creativity common in other software development platforms. C

References

1. Apps.Gov. *Federal Government Mobile Apps Directory*; https://www.usa.gov/mobile-apps
2. Bakici, T., Almirall, E., and Wareham, J. The role of public open innovation intermediaries in local government and the public sector. *Technology Analysis & Strategic Management 25*, 3 (2013), 311–327.
3. FixMyStreet. FixMyStreet for Councils; https://www.fixmystreet.com/council
4. FixMyStreet. *FixMyStreet Platform*; http://fixmystreet.org/
5. Freitas, A., Curry, E., Oliveira, J.G., and Riain, S.O. Querying heterogeneous datasets on the Linked Data Web: Challenges, approaches, and trends. *IEEE Internet Computing 16*, 1 (2012), 24–33.
6. Lakhani, K.R., Austin, R.D., and Yi, Y. *Data.gov.* Harvard Business School, Cambridge, MA, 2010; http://www.hbs.edu/faculty/Pages/item.aspx?num=38841
7. Open Government Initiative. *About Open Government*; https://www.whitehouse.gov/open/about
8. Opsahl, A. Washington, D.C. reloads Apps for Democracy. Government Technology (June 2, 2009); http://www.govtech.com/pcio/Washington-DC-Reloads-Apps-for-Democracy.html
9. Ramm, F., Topf, J., and Chilton, S. *OpenStreetMap: Using, and Contributing to, the Free World Map.* UIT Cambridge, Cambridge, U.K., 2010.
10. White House Status Report. *The Obama Administration's Commitment to Open Government*; https://www.whitehouse.gov/sites/default/files/opengov_report.pdf
11. World Bank Sanitation Hackathon. *Sanitation App Challenge*; http://sanitation.hackathome.com/

Melissa Lee (melissajo.lee@esade.edu) is a Ph.D. candidate in management at ESADE Business School in Ramon Llull University, Barcelona, Spain.

Esteve Almirall (esteve.almirall@esade.edu) is a professor of innovation and data sciences at ESADE Business School in Ramon Llull University, Barcelona, Spain.

Jonathan Wareham (jonathan.wareham@esade.edu) is a professor of information systems and Dean of Faculty & Research at ESADE Business & Law Schools in Ramon Llull University, Barcelona, Spain.

DOI:10.1145/2756545

Before looking to enter a cloud-based market, weigh industry characteristics and one's own stock of design capital.

BY LAZAROS GOUTAS, JULIANA SUTANTO, AND HASSAN ALDARBESTI

The Building Blocks of a Cloud Strategy: Evidence from Three SaaS Providers

CLOUD COMPUTING REFERS to an on-demand network service that allows individual users or businesses to access configurable resources. It can also be defined as an on-demand delivery model enabling the synchronized delivery of computing resources (such as applications, storage, servers, networks, and services).[2] As it stands, there are three cloud computing delivery models: software as a service (SaaS), as in Salesforce.com and Google apps, delivering applications to end users over a network; platform as a service (PaaS), as in the Google app engine and Microsoft Azure, deploying applications to a cloud; and infrastructure as a service (IaaS), as in the Amazon Elastic Compute Cloud, renting storage, processing, and network capacity to host applications. Of the three, the SaaS model has gained the most momentum, given its economically efficient foundations and ability to satisfy user preferences for the ubiquitous availability of data and applications.[1]

From the perspective of application software providers, the SaaS model offers the obvious benefit of liberating them from the traditional low-level tasks involved in setting up IT infrastructures and deploying applications to client machines.[4] Providers are thus able to scale their investment with a view to growing their businesses,[9] focusing on innovation and creating business value.[7] Accordingly, cloud computing has been associated with other benefits that arise from offering a controlled interface, a virtual business environment, increased addressability and traceability, and rapid elasticity and scalability.[5]

It is easy to understand why application software providers face increasing pressure to jump into the cloud and exchange their on-site application software for cloud-based solutions. However, the unfortunate reality is

» key insights

- **Motivated by the hype around providing application software through the cloud, many firms approach the cloud without a clear strategy, often resulting in disappointment.**
- **Deciding to offer cloud-based software must be part of a broader digital strategy informed by the industry characteristics in which a firm operates, as well as by the firm's existing stock of internal systems and processes, or its design capital.**
- **A cloud strategy must also account for customers' views of and concerns over cloud-based SaaS, especially demand for security optimization and software customization.**

IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

most promised benefits have turned out to be a triumph of hype over reality. A 2013 *Forbes* article noted many firms are following a "no-strategy" approach in moving to the cloud, leading more often to failure than success;[8] for example, Adobe's Creative Cloud product line has been impeded by the skepticism of customers not yet ready to move to subscription-based services; concerns include file recovery in the event of a subscription lapse and the need for more-tailored offerings for photography enthusiasts.[10]

As we aim to show here, there is a clear need for business managers to better understand the fundamental underpinnings of a successful cloud-based SaaS strategy, or cloud strategy, which is defined as a "set of decisions required to create and deploy a network-based, information-service-delivery strategy that results in both cost savings and organizational agility."[5] A successful cloud strategy must encompass some of the key elements distinguishing a broader digital business strategy, including a series of higher-order dimensions relating to the characteristics of a firm's respective industry and its existing stock of internal technological capabilities. Moreover, we show complementing these dimensions with certain attributes of cloud technology related to the actual application software leads to formation of distinct cloud strategies.

Components

A digital business strategy is broadly understood as the means through which a firm engages in any category of IT activity; the strategic nature of this engagement implies the "dynamic synchronization between business and IT to gain competitive advantage."[6] Recent studies have identified more specific elements of a digital business strategy, examples of which include a firm's digital strategic posture[6] and the design-based dimension of a digital business strategy.[11]

The digital strategic posture is defined as a firm's degree of engagement in a particular digital business practice relative to the industry norm.[6] The degree to which a firm chooses to diverge from or converge on the industry norm in its ongoing digital business strategy is influenced by the interaction between its current digital strategic posture and three key elements of its industry environment: turbulence, concentration, and growth.[6] These elements are defined as follows: industry turbulence is the rate at which a firm enters and exits an industry; concentration is the extent of competitive rivalry in an industry; and growth is the rate of increase in demand for the industry's output.[6] Mithas et al.[6] proposed strong industry turbulence, low industry concentration, and low industry growth influence firms to develop digital business strategies that diverge from industry norms due to intense competition and the fact such norms are less reliable guides of future success. In contrast, the same authors argued that low industry turbulence, high industry concentration, and high industry growth influence firms to develop digital business strategies that converge on industry norms, as these norms are reliable indicators of the possible success of particular strategic moves.[6]

As opposed to looking at external factors, Woodard et al.[11] proposed the design-based logic of a digital business strategy that examines a firm's internal systems and processes, or its design capital. Design capital includes the firm's option value, or the breadth of opportunities afforded by its internal systems and processes, and technical debt, or expected cost or effort to exercise the opportunities.[11] Woodard et al.[11] further proposed a digital business strategy should aim to manage the levels of option value and technical debt associated with a firm's design capital toward the ideal state of high-quality design capital characterized by high option value and low technical debt. This ideal state allows a firm to seize a range of market opportunities and respond to competitors' actions with speed and scale.

Although a digital strategy is not synonymous with a cloud strategy, insights from these higher-level frameworks arguably serve as a useful foundation for better understanding how firms approach the cloud. This conviction stems from the fact that a cloud strategy is inherently embedded in a broader digital strategy, and also from the fact that the industry environment and a firm's internal capabilities are the main determinants of a firm's competitive strategy.[6,11] We draw empirical support for this insight by analyzing recent strategic decisions to offer cloud-based application software made by three firms.

Case Analysis

All three firms are located in the same European country but operate in different industries. Firm 1 is a telecommunications provider; Firm 2 is a small engineering-simulation-software provider; and Firm 3 is a mid-size company specializing in customer relationship management (CRM) software. While it may take a while before one can conclude whether these companies' cloud strategies will ultimately succeed or not, by synthesizing the analyses of the three at the point they made their decisions, we contribute to both researcher and practitioner understanding of the different parameters firms must take into account when unfolding a cloud strategy.

Major telecommunications provider. Firm 1 is a large telecommunication provider serving both residential and business customers. The telecommunication industry is characterized by a high degree of industry turbulence, where firms frequently enter and exit from different industries (such as those in the mobile applications industry offering customers cheaper alternatives for long-distance calls), high industry concentration (generally, only a few telecommunication providers compete in a given country, three in the country where Firm 1 operates), and high industry growth (demand for improved connectivity and speed are constantly increasing). In this environment, it is not immediately clear whether a firm's digital business strategy should diverge from or converge on the telecommunications industry norm, supplementing traditional offerings (such as voice calls and data offerings) with relatively nontraditional arrangements (such as mobile payments) by collaborating with firms in other industries (such as financial services).

An assessment of Firm 1's internal systems and processes (digital capital) positions the firm in the debt-constrained design capital state, or high option value and high technical debt.[11] While its telecommunication infrastructure appears to give it plenty of options for entering the cloud business, significant investment is needed to make it IaaS-ready, as a 2012 report said, "We thoroughly discussed with [our] cloud architects ... about the IaaS. Their response was positive, but it would be very expensive ... " In such a debt-constrained state, depending on the level of its resource munificence, the firm would need to either abandon the option or reduce its debt.[11] Debt-constrained firms with access to abundant resources, as with Firm 1, can afford to reduce their debt without abandoning their strategic options.

Considering other cloud-computing delivery models besides IaaS, Firm 1 decided to invest in developing its own cloud infrastructure, as well as new business models offering various SaaS-based products to end users (in this case, business users) instead of offering IaaS-based products to application service providers. Leveraging its current position as a trusted telecommunications provider and the well-known data-protection policy of the country in which it operates, Firm 1 targets enterprises operating in high security-loss environments with highly critical SaaS security optimization. The main value proposition Firm 1 offers these enterprises is the security of their data, as the data does not leave the country (the cloud farms are located in the country where Firm 1 operates), and the security of their data processing, as the SaaS-based products will be hosted locally by Firm 1 itself.

Part of Firm 1's initiative is to collaborate with a CRM software provider to provide CRM SaaS to business customers. Note besides the existing enterprises the focal CRM provider serves, the immediate target customers of Firm 1 are mid- and large-size financial enterprises in the country that are current subscribers to its telecommunications network. These financial enterprises are working in high-security-loss environments and are likely to suffer major economic losses if the CRM system is subject to security attacks.[1]

However, the specific nature of the CRM SaaS, which supports multiple tenancy with high parameterization or customization, is technically complex and can be very expensive. According to the firm's CRM corporate collaborators, the CRM software does not support multiple tenancy, because each client (tenant) tends to require system parameterization; that is, " ... customers that buy the CRM software usually demand a customized system according to their business processes and therefore the support of multiple tenancy with high parameterization is technically very complex and expensive" (excerpt from our communication with a business development manager in Firm 1). This situation is a good example of how a firm's digital capital is intertwined with its clients' technical needs, in this case, support for multiple tenancy with high parameterization. Having access to technical resources, Firm 1 is able to find a secure cost-effective solution to the multiple tenancy challenge of providing highly customized systems by deploying multiple software instances for different tenants at a single server; the client can thus achieve software customization while still enjoying the benefits of a cloud-based service.

We found customers' requirements involving software security and customization are the two main attributes that determine a firm's decision to change its on-premise software to cloud-based SaaS.

As it turned out, Firm 1 decided to diverge from the industry norm; rather than offer the usual telecommunication infrastructure-related offerings, it aimed to offer SaaS to business customers, thus altering its position in the current industry ecosystem from being a telecommunications provider to being a SaaS provider. Its new position as SaaS provider will enable Firm 1 to enjoy multiple benefits " ... as it enables vertical selling opportunities in addition to the license fees such as iPads [rental] and voice and data subscriptions ... " (excerpt from an internal Firm 1 report). At the moment, " ... the infrastructure to host the SaaS is an ongoing work and expected to be available by early 2016 ... there is [still] a high business interest to start this SaaS project ... within the next year" (excerpt from our communication with a business development manager in Firm 1).

Engineering-simulation-software provider. Our second case is a small provider of engineering simulation software specializing in computational fluid dynamics and multiphase flow heat transfers. Its software is sold globally and used mainly by research organizations and companies in the oil and gas industry, nuclear engineering, renewable energies, microfluidics, and advanced materials science (hereafter referred to as "client companies"). These companies' use of this advanced software is limited by their access to computing power; only a few clients have the computational resources (parallel computers or clusters) required to run very demanding simulations, thus shrinking the size of the engineering simulation software market and contributing to low industry growth.

The competition for this already small market is fierce and dominated by two large companies, so is thus characterized by high industry concentration. "Unlike [the] commodity [software] market, the engineering simulation [software] market is highly oriented toward a " ... dominant design. This means the incumbents try to make their competitors obsolete by locking their customers into their software logic [and algorithms] ... " (excerpt from an internal Firm 2 report). Although the software logic and algorithm may not be the most efficient (or even the most appropriate), client companies will incur high switching costs if they change from one software provider to another due to organizational latency, training, know-how transfer, and learning curves. This is a strong indication of low industry turbulence, as it is difficult for firms from different sectors to enter and exit the industry. To sustain itself in this small yet highly competitive market, Firm 2 also offers consultancy services. In this environment of low turbulence, high concentration, and low growth, it is not immediately clear whether its digital business strategy should converge on or diverge from the engineering simulation industry norm, implying the need (for small providers) to heavily supplement their software offerings with consultancy services. However, Firm 2 recognizes providing a consultancy service is not as scalable as its software offering and thus (although it is the industry norm) may not be a sustainable strategy in the long run. As the founder of Firm 2 said in an interview with us, "[The consultancy service] is very labor intensive." Hence, the tendency is to follow a business strategy that diverges from the industry norm.

In terms of its digital capital, Firm 2 is in the low-quality design capital state, or low option value and high technical debt.[11] Such a firm, depending on its level of resource abundance and technical capability, can aim to either reduce its technical debt or create different value options.[11] By default, Firm 2 is constrained by a lack of resources but at the same time enjoys a strong relationship with academic stakeholders. As the founder of Firm 2 told us, "We are working closely with a research institute in a local university," thus enabling Firm 2 to exit a low-quality state by increasing its technical capabilities.

Since the market is small, not very adaptable, and dominated by two large companies, Firm 2 is considering (with the aid of its academic contacts) deploying its software as a cloud computing hosted service. Firm 2 also reported "Companies on the edge of starting engineering simulation activities are not willing to invest in IT infrastructure acquisition and long-term maintenance contracts. They rather [tend to] spread [their] investment over time, much like any other operation consumable. Moreover, yearly software license fees constitute a financial burden, especially when the software vendor enjoys a quasi-monopoly status. [In this context] cloud computing appears … [to be] a real alternative answer to engineering needs … " (excerpt from an internal Firm 2 report). Since the engineering data is not sensitive and the simulation process need not be performed in a highly secure environment, the cloud-hosted SaaS solution seems to be a viable way for Firm 2 to compete with the dominant players in the current market and help increase its market share.

The cloud is not only a technology that enables businesses to embrace opportunities for innovation, it also serves as a catalyst for business-model transformation.

Firm 2 realized the high switching costs associated with its product offering implies the decision as to which simulation software to adopt is in the hands of client companies' top managers, who may not be familiar with algorithms and simulation-software logics. For their engineers to use inefficient software on daily basis could thus be frustrating. With a Web-based cloud-computing hosted service, Firm 2 is able to invite these engineers to test its software during a free trial period, without having to access or use their companies' own computational resources. Firm 2's aim is to allow the engineers who are the real users of the simulation software to use it and test its efficiency and accuracy in a cloud environment, hoping they can then convince their top management to switch to Firm 2's software.

The switching cost from on-premise software to a cloud-computing hosted service is marginal. Moreover, since the cloud-based simulation software is meant to allow virtually anyone to perform highly demanding engineering simulations, with no infrastructure prerequisites, Firm 2 intends to increase its share of its target market beyond that of large companies with computational resources in place. Firm 2 is currently working with a cloud broker and a cloud infrastructure provider to implement its cloud-based solution and bring its software to the cloud. Firm 2 intends to " … use it exactly in the same way as planned: it will lock new customers by offering them trial access without software installation … " (excerpt from an internal Firm 2 report).

CRM software provider. Firm 3 is a mid-size software company specializing in CRM software, reporting, "[The CRM software] is available as fat client: it has iPhone and iPad applications and also has a Web front-end. This allows customers to flexibly use the front-end that is most suitable for their processes. [For example], sales personnel can use the iPad version to be fully mobile while being with the customer, so a call center agent can use the fat client perfectly optimized for his tasks … " (excerpt from a from an internal Firm

3 report). Here is our analysis of Firm 3's industry environment. The CRM software industry is characterized by low industry turbulence (where firms' entries and exits from different industries are less frequent), high industry concentration (few well-known CRM software providers), and high industry growth (increased demand for CRM software, especially from small- and mid-size companies). In this environment, the literature predicts Firm 3's digital business strategy should converge on the industry norm, as it is relatively easy to determine the optimal level of IT investment and its potential for success.[6] The industry norm for firms like Firm 3 that offer on-premise CRM software is to supplement this traditional offering with the SaaS version of CRM software. As our conceptual framework would predict, Firm 3 indeed views cloud computing as a new opportunity that could extend its business and thus plans to provide the SaaS version of its CRM software, alongside the on-premises version. Firm 3 says, "As the current [on-premises] CRM version already fulfils the main characteristics of online access with multiple devices, no local data storage, and scalability, the private offering can mainly be seen as a marketing enhancement … " (excerpt from an internal Firm 3 report).

Our own assessment of Firm 3's internal systems and processes (digital capital) positions Firm 3 in the debt-constraint design capital state sector (high option value and high technical debt).[11] A significant investment is thus required to produce the SaaS version of the software, and there is internal Firm 3 concern about going in this direction, as indicated by the following excerpt from a Firm 3 report: "The question of whether the CRM [software] should be offered in the cloud or not is omnipresent. This dilemma involves various factors. As [the company] is not a big software producer, this dilemma needs to be taken seriously, as [the required] financial investment can hardly be covered in case of a failure." In such a debt-constrained state, Firm 3 should either abandon the option or reduce its debt, depending on the level of its resource munificence.[11] Firm 3 recognized the significant resources needed to develop the SaaS version of its software are a strategic necessity, as it said in an internal report " … there is not only the question whether it is worth to invest into a cloud solution, but also whether it is possible for [the company] to survive in the long run without a cloud solution … " As noted earlier, the current industry norm is for CRM software providers to offer a cloud solution, "… the topic is brought up [in the company] as the customers start asking for it" (excerpt from a Firm 3 email memo).

Recognizing an aggressive move to compete with the big providers of CRM SaaS would be difficult and most likely lead to a price war, Firm 3 sought to identify a number of unique selling propositions that were difficult for its competitors to imitate before moving to the cloud. Specifically, it saw a regional advantage, a legal advantage (storing customers' data according to country-specific-laws), and a know-how advantage (involving a specific security algorithm). These unique selling propositions stem from the fact the business users who work with client data are in high-security-loss environments and face considerable economic loss if the CRM system would suffer a security attack,[1] as well as from the fact that Firm 3 has already built significant levels of personal trust with its clients.

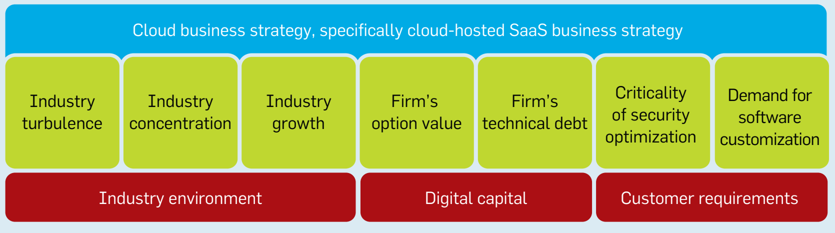
Firm 3 mainly serves customers in its home country and a customer base consisting mainly of large organizations in various industries, especially in the retail, pharmaceutical, and insurance sectors. With selling propositions in mind, Firm 3 decided against a public cloud offering, as it saw a number of constraints that would disrupt its existing business model, beyond even its in-depth, personal relationships with all its customers. Shifting to a public cloud offering would also imply a cultural change within the company that would be difficult to achieve. Firm 3 opted instead for a private cloud solution, as it would be largely compatible with its existing personalized services. This new offering might attract new customers that do not want to operate the application themselves but do want their own private application. The only constraint in this case was the lack of a data center where the application could be hosted. As running such a data center is not its core business, Firm 3 decided to outsource the task to a well-known provider that could ensure the scalability and, more important, the security demanded by its customers.

Insights

Analyzing these cases, we reached two main conclusions. First, in addition to the five building blocks related to the external environment and a firm's broader internal digital capabilities, all three firms had to account for certain requirements of their customers with respect to cloud technology before deciding how to approach them with a cloud-based SaaS offering. In particular, we found customers' requirements involving software security and customization are the two main attributes that determine a firm's decision to change its on-premise software to cloud-based SaaS. The criticality of security optimization depends on whether target users are working in high- or low-security-loss environments,[1] whereas the importance of software customization depends on the type of software being offered.

Second, we found the three firms developed three different strategies in terms of utilizing the cloud to compete in their industries, as well as their value propositions. In our empirical exam-

The seven building blocks of a cloud-hosted SaaS business strategy.



The building blocks of a cloud business strategy as reflected in three cases.

	Firm 1 Innovating Strategy	Firm 2 Disrupting Strategy	Firm 3 Optimizing Strategy
Industry turbulence	High	Low	Low
Industry concentration	High	High	High
Industry growth	High	Low	High
Option value	High	Low	High
Technical debt	High	High	High
Criticality of software security optimization	High	Low	High
Demand for software customization	High	Low	High

ples, Firm 1 can be seen as an innovator, Firm 2 as a disruptor, and Firm 3 as an optimizer. We define innovators as firms offering cloud-based application software to create new revenue streams by moving into an adjacent ecosystem or marketplace. In the course of this extension and transformation, innovators often have a chance to combine elements of the value propositions and value chains that were previously unrelated, and so increase their competitive advantage.[3] The cloud is not only a technology that enables businesses to embrace opportunities for innovation,[3] it also serves as a catalyst for business-model transformation.

Unlike innovator strategies, companies classified as having disruptive strategies share the perception that cloud-based application software offerings can generate completely different value chains. We define disruptors as firms that either radically reformulate customer value propositions or generate new customer needs in their current ecosystems. Disruptors have the potential to capture inimitable competitive advantage by creating disruptive mechanisms in existing markets or industries. These firms typically provide customers with what they either were unaware of or did not realize they needed. While businesses using this model face greater risk, they also tend to gain higher rewards. Cloud computing enables the radical transformation of existing markets or industries by enabling businesses to be more agile and adopt technology-integrated business strategies in place of technology strategies based on business strategies.[3]

We define optimizers as firms drawing advantage from cloud computing to improve their existing customer value propositions within their existing ecosystems. Optimizers can expand their value propositions by offering enhanced products and services, improved customer experience, and/or more extensive channel-delivery options;[3] they also tend to be more risk-averse than innovators and disruptors. By supporting fast experimental implementation of new application software offerings without need for substantial upfront costs, cloud computing drives improvement across an optimizer's value propositions and value chains.

Important to note is that adopted strategies are contingent on the configuration of the focal firms with respect to the initial five building blocks, as well as the remaining two blocks related to their target clients' requirements of the application software through the cloud. The figure here outlines the seven building blocks that largely determine whether a firm should use cloud computing to innovate, disrupt, or optimize its business model. The first five are derived from the two frameworks of digital business strategy we discussed earlier, namely those of digital strategic posture[6] and design-based logic of digital business strategy.[11] The remaining two—the criticality of security optimization and demand for software customization—emerge from the cases and refer to clients' requirements with respect to a cloud-based SaaS offering. These two attributes of the software appear to complement the broader categories of a firm's industry environments and digital capital.

The first three building blocks, which we categorize as the characteristics of the industry in which a firm operates, include the degree of turbulence, concentration, and growth rate of the industry.[6] It has been established that high (low) industry turbulence, low (high) industry concentration, and low (high) industry growth would influence a firm to develop a digital business strategy that diverges from (converges on) industry norms.[6] In our three empirical examples, we observed two scenarios. In one, Firm 3 operates in an industry with low turbulence, high industry concentration, and high industry growth, and so has designed a digital business strategy that converges to the industry norm to optimize its existing software offering by adding a SaaS version of the software. Firm 3 imitates what happens in its industry without much innovation or attempt to disrupt industry norms. In the other, Firm 1 operates in an industry distinguished by high turbulence, high concentration, and high growth, whereas Firm 2 is in an industry characterized by low turbulence, high concentration, and low growth. For these two firms, not immediately clear is whether they should diverge from or converge on their respective industry norms with regard to developing cloud-related strategies.

The next two building blocks—option value and technical debt—refer to a firm's design capital (in terms of internal systems and processes).[11] While Firm 1 is in the debt-constrained design capital state (high option value and high technical debt), Firm 2 is in the low-quality design capital state (low option value and high technical debt),[11] but its close relationships with local academic institutions and researchers enables it to escape a low-quality state by accessing the technical capabilities it needs from these institutions. Leveraging its current position as a trusted telecommunication provider and the well-regarded data-protection policy of the country in which it operates, Firm 1 decided to invest considerable resources to innovate and alter its position in its industry ecosystem—from telecommunications infrastructure provider to SaaS provider specializing in servicing business users in high-security-loss environments—meaning Firm 1 is diverging from its industry norm.

Unlike Firm 1, which also diverges

from its industry's norm, Firm 2 hopes it will disrupt the operations of the current industry players by offering SaaS for engineering simulation instead of innovating by establishing a new revenue stream through a cloud-based offering. It considers this a radical solution that will allow it to disrupt the current top-down nature of its clients' decision making about engineering-simulation software, and thus break through the dominant market players. By offering a Web-based cloud-computing hosted service, Firm 2 can invite its potential customers' engineers—the ultimate users of its software—to test it during a free trial period without them needing to access or use their own companies' computational resources; it hopes these engineers will then convince top management in their organizations to adopt its software.

There are two main reasons for this apparent difference in the strategies of Firm 1 and Firm 2. First, engineering-related data for simulation purposes does not need security optimization; neither the storage nor the processing of the data has to be fully secured. This makes it easy for Firm 2 to disrupt the market by inviting the engineers who they hope will be their software's ultimate users to test it by uploading engineering-related data to the cloud, avoiding having to access or use their organizations' own computational resources to test the software. It would be difficult to disrupt the current market through a free trial of a Web-based cloud-computing-hosted service if demand for data-security optimization was high. And second, the software offered by Firm 2 need not cater to users' demand for software customization. Engineering-simulation software has a specific logic and algorithms that are relevant for each engineer using it. A less-critical security optimization and low demand for software customization ultimately drive Firm 2's attempt to grow its market share by adopting a disruptive cloud strategy.

Even though the attributes of the industry environment seem to be the main driver of Firm 3's cloud strategy, our analysis of its clients' demand for security optimization and customization complete its detailed plan. Considering the high potential risk to customers' data and the high demand for software customization, Firm 3 decided to offer a private cloud-based application software solution. Unlike Firm 1, Firm 3 is in a state of debt-constrained design capital (high option value and high technical debt). However, Firm 3 cannot invest as much in infrastructure and know-how as Firm 1. Investing aggressively to acquire necessary infrastructure and know-how, Firm 1 is able to deploy multiple software operations for different tenants on a single server, enabling it to offer customization while still being able to pursue the cost benefits of multiple tenancy. With only limited resources to invest, Firm 3 continues to seek to collaborate with a well-known cloud infrastructure provider to offer its customization services through a private cloud-based system, thus optimizing rather than innovating in terms of its cloud offering.

The table here outlines the seven building blocks of a cloud strategy and the resultant strategies adopted by the three firms in our study. Note there can be other combinations of the seven building blocks than those we cover, nor is it our aim to present all possible combinations; for instance, we anticipate having distinct structural characteristics (such as a start-up firm without an existing customer base, as in our empirical examples) would most likely lead to distinct configurations in our scheme. In this respect, any future study that would examine how startup companies strategize their cloud-based SaaS offerings according to our framework would complement our findings.

Conclusion

Our main aim here is to assist researchers and practitioners alike in utilizing the building blocks identified as essential ingredients for analyzing a firm's cloud strategy. We thus provided an overarching framework consisting of seven building blocks encompassing the characteristics of the firm's industry environment, internal digital capabilities, and target clients' requirements for the particular cloud-based offering. These blocks ultimately determine how firms embark on a cloud-based SaaS strategy.

While significant challenges exist for firms offering application software through the cloud, we find it important to start right and better understand the main building blocks of the endeavor. Innovation, optimization, and disruptive strategies represent possible ways for firms to leverage the cloud to advance their value propositions. It is important to highlight we do not advocate any one approach as superior to the others; rather, the strategies identified here should be viewed as a viable aftermath of a firm's industry characteristics (turbulence, concentration, and growth), stock of digital capital (option value and technical debt), and clients' requirements for the cloud-based SaaS offering (criticality of security optimization and software customization). C

References

1. August, T., Niculescu, M.F., and Shin, H. Cloud implications on software network structure and security risks. *Information Systems Research 25*, 3 (Sept. 2014), 489–510.
2. Bento, A. and Bento, R. Cloud computing: A new phase in information technology management. *Journal of Information Technology Management 22*, 1 (Mar. 2011), 39–46.
3. Berman, S., Kesterson-Townes, L., Marshall, A., and Srivathsa, R. *The power of cloud: Driving business model innovation.* IBM Institute for Business Value, Feb. 2012; http://www-935.ibm.com/services/us/gbs/thoughtleadership/ibv-power-of-cloud.html
4. Geetha, M., Kanagamathanmohan, K., and Paul, C. Ranking the cloud services based on QoS parameters. *International Journal For Technological Research In Engineering 1*, 5 (Jan. 2014), 272–276.
5. Iyer, B. and Henderson, J. Preparing for the future: Understanding the seven capabilities of cloud computing. *MIS Quarterly Executive 9*, 2 (June 2010), 117–131.
6. Mithas, S., Tafti, A., and Mitchell, W. How a firm's competitive environment and digital strategic posture influence digital business strategy. *MIS Quarterly 37*, 2 (June 2013), 511–536.
7. Padhy, R. and Patra, M. Managing IT operations in a cloud-driven enterprise: Case studies. *American Journal of Cloud Computing 1*, 1 (Aug. 2013), 1–18.
8. Peters, B. The worst and best enterprise cloud strategies. *Forbes* (May 21, 2013); http://www.forbes.com/sites/bradpeters/2013/05/21/the-worst-and-best-enterprise-cloud-strategies
9. Shayan, E., Ghotb, F., and Suganda, R. Characterizing diffusion patterns of cloud computing. *International Journal for Management Science and Technology 1*, 9 (Nov. 2013), 2–15.
10. Vance, A. IBM faces a crisis in the cloud. *Bloomberg Businessweek* (Nov. 15, 2013); http://www.businessweek.com/articles/2013-11-15/ibm-faces-a-crisis-in-the-cloud
11. Woodard, C.J., Ramasubbu, N., Tschang, F., and Sambamurthy, V. Design capital and design moves: The logic of digital business strategy. *MIS Quarterly 23*, 2 (June 2012), 537–564.

Lazaros Goutas (L.Goutas@lboro.ac.uk) is a lecturer in information systems at the School of Business and Economics, Loughborough University, Loughborough, U.K., and a research fellow at the Global Center for Digital Business Transformation, IMD Business School, Lausanne, Switzerland.

Juliana Sutanto (j.sutanto@lancaster.ac.uk) is a professor of information systems in the Department of Management Science at Lancaster University, Lancaster, U.K.

Hassan Aldarbesti (ahassan@ethz.ch) is a Ph.D. candidate in the Department of Management, Technology, and Economics at ETH Zurich, Zurich, Switzerland.

DOI:10.1145/2742544

Algorithmic solutions to tough computational problems are making an impressive comeback.

BY IOANNIS KOUTIS AND RYAN WILLIAMS

Algebraic Fingerprints for Faster Algorithms

IT WAS A major surprise when, in 2010, Andreas Björklund discovered what many previously thought impossible: a significantly improved algorithm for the famous *Hamiltonian path* problem, also known as Hamiltonicity.[4] Hamiltonicity asks if a given graph contains a path that goes through each vertex exactly once, as illustrated in Figure 1.

Hamiltonicity was one of the first problems shown to be NP-complete by Karp.[20] The only known algorithms for NP-complete problems require time scaling *exponentially* with the size of the input. It is believed they cannot be solved much faster in general, although the possibility has not been ruled out.[17] Given an undirected graph on n vertices, Björklund's algorithm can find a Hamiltonian path or report that no such path exists in $O^*(1.657^n)$ time.[a] The algorithm still runs in exponential time but it is much faster than the $O^*(2^n)$ running time of the previously fastest algorithm, known since the 1960s.[3,19]

Hamiltonicity is a prominent algorithmic problem. Many researchers before Björklund have tried their hand at it, without success. But some of the tools Björklund used did not become available until 2009 thanks to progress in the k-path problem, a related problem in the context of *parameterized algorithms.*

A parameterized race and a conjecture. The k-path problem is a natural parameterized analogue of Hamiltonicity. The goal now is to find in the given graph a path of length k for some specified value of k, rather than a path of length n. It could be argued that, from a practical standpoint, the k-path problem is better motivated relative to Hamiltonicity. Algorithms designed specifically for the k-path problem have been actually used to detect signaling pathways in large protein interaction networks.[29]

Most NP-complete problems have similar parameterizations, where the parameter k measures the *solution size* to the given instance. This is natural because in practice, the solution length to an NP-hard problem is often short relative to the length of the overall instance. Downey and Fellows[12] observed that some parameterized problems appear to be "easier" than others. To capture that,

» key insights

- **There has recently been impressive progress—after nearly 50 years of stagnation—in algorithms that find exact solutions for certain hard computational problems, including the famous Hamiltonian path problem.**
- **This progress is due to a few core ideas that have found several applications. A unifying theme is algebra: we "transform" the given problem into a more general algebraic format, then solve the corresponding algebraic problem that arises.**
- **This article walks the reader through some of these exciting developments and the underlying ideas. It also puts them in context with the discover process that led to them, highlighting the role of parameterization.**

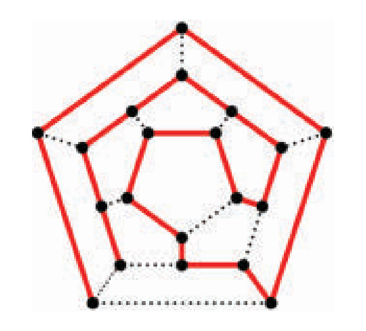
a $O^*(f(k))$ means a function smaller than $p(n) \cdot f(k)$ for some polynomial $p(n)$.

they defined the class of *fixed parameter tractable* (FPT) problems. The definition of FPT is rather simple. The running time of a parameterized algorithm should clearly depend on the input size n, and the parameter k. But the dependence should be somewhat special: a problem is defined to be FPT if it can be solved by an algorithm in $O^*(f(k))$ time.

The FPT notion has proven to be extremely insightful and has led to a more detailed understanding of problem hardness. It is believed that not all NP-hard problems are FPT; a prominent example is the k-clique problem, where the goal is to detect a k-subset of the network nodes that are all pairwise connected (see Figure 2). The best known algorithm for k-clique is not much better than exhaustive search, which takes time $O(n^k)$. However, experience shows that when a problem is shown to be FPT, algorithmic progress has just begun as it is often the case that the function $f(k)$ in the running time can be improved.

The k-path problem is a prime example. The problem was shown to be FPT by Monien[25] who described an algorithm running in $O^*(k!)$ time. This was later improved to $O^*((2e)^k)$ by Alon et al.[2] and to $O^*(4^k)$ by Chen et al.[10] Each of these steps represents the introduction of a significant new idea. Similarly, for many FPT problems there are now *races* for faster parameterized algorithms,[27] which have enriched the field of algorithm design with new techniques.

Figure 1. A Hamiltonian path.



However, with each improvement, one is faced with the question: Is further progress possible? Known algorithms for the "parent" NP-complete problem can be a valuable guide to answering this question. This is because their running time sets a clear target for the parameterized algorithm. In the case of Hamiltonicity, we know of an algorithm that runs in $O^*(2^n)$ time. It is then reasonable to conjecture that there is an algorithm for k-path that runs in $O^*(2^k)$ time, "matching" the Hamiltonian path algorithms for the extreme value of the parameter.

Roadmap. The development of the algorithmic techniques we are about to discuss was largely driven by the k-path conjecture. But for the sake of simplicity we will illustrate them in the context of another important NP-complete problem known as the 3D-matching problem, which has also played a central role. In fact, the Hamiltonicity algorithm was preceded by another record-breaking algorithm for the 3D-matching problem, also due to Björklund.[5]

In this article, our first step toward faster algorithms is what we call an *algebraization* of the combinatorial problem, that is, capturing the problem as a question about the existence of certain monomials in a polynomial. We discuss one algebraization of the 3D-matching problem later. Given the polynomial, the original combinatorial question becomes now an issue of *algebraic detection* of monomials; that is, extracting information from the polynomial by assigning values into its variables. Looking for assignments that fit our purpose will lead us to calculations modulo 2, over an algebra in which sums of pairs cancel out. This in turn will present us with a problem of *unwanted cancellations* of monomials. We will solve it with the method of *fingerprints* that augments the 3D-matching polynomial in order to ensure no unwanted cancellations occur. The k-path conjecture was answered in the positive via a generalization of algebraic detection with fingerprints, which also yielded many other faster parameterized algorithms. These ideas are reviewed later, as is Björklund's key twist that helped unlock the faster Hamiltonicity algorithm. While still employing in a relaxed way the method of fingerprints, Björklund treated computation modulo 2 not just as a nuisance, but also as a resource, by presenting algebraizations where many *unwanted monomials cancel out*, because they come in pairs. In order to derive these sharper algebraizations that *exploit cancellations*, Björklund tapped into the power of algebraic combinatorics that study graphs via linear algebra. Faster algorithms for several other parameterized problems followed.

Algebraization

Consider the following fictional airline scheduling problem: on any given day there is a set X of designated captains, a set Y of designated second officers, and a set Z of destinations. Each captain and second officer declare a number of preferred destinations. The airline would like to accommodate as many preferences as possible; but at the same time, they would like to match on the same flight a pilot and second officer who have flown together before. This creates a number of triples: if captain x has previously flown with second officer y and they would both like to fly to destination z, they define a triple $\{x, y, z\}$. Of course, any given captain, second officer or destination can appear in many triples. But to maximize utility, the airline would like to select a big matching, that is, a subset of triples that are pairwise disjoint; based on it they can then schedule the personnel. The NP-complete problem asks for a matching of maximum size, while the parameterized problem asks for a k-matching with at least k triples, which we also call a k-3D matching. An example of the problem is shown in Figure 3.

A polynomial for 3D-matching. It was probably understood by many before it was made explicit in Koutis[21] that we can view the k-3D-matching problem through an algebraic lens. Let us explain how, by means of the example in Figure 3.

First, we view as a variable each element of the sets X, Y, Z. That is we have the variables $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$ and $Z = \{z_1, z_2, z_3, z_4\}$. Then for each triple we construct a monomial, by taking the product of the corresponding variables. In our example of Figure 3, we have the monomials.

$$\{x_1y_2z_2,\ x_2y_1z_1,\ x_3y_2z_3,\ x_3y_3z_4\}.$$

We also define the *instance polynomial* P_1 to be the sum of these monomials. Finally, we set $P_k = P_1^k$; we will call P_k the kth *encoding polynomial.* For instance, for the 2nd encoding polynomial, we have

$$P_2 = (x_1y_2z_2 + x_2y_1z_1 + x_3y_2z_3 + x_3y_3z_4)^2.$$

To see the motivation behind these definitions, consider the expansion of P_2 into a sum-product form.

$$\begin{aligned} P_2 = {} & (x_1y_2z_2)^2 + (x_2y_1z_1)^2 \\ & + (x_3y_2z_3)^2 + (x_3y_3z_4)^2 \\ & + 2x_1x_2y_1y_2z_1z_2 + 2x_1x_3y_2^2z_2z_3 \\ & + 2x_1x_3y_2y_3z_2z_4 + 2x_2x_3y_1y_2z_1z_3 \\ & + 2x_2x_3y_1y_3z_1z_4 + 2x_3^2y_2y_3z_3z_4. \end{aligned} \tag{2.1}$$

The monomials in the sum-product expansion fall into two basic classes. In the solution part of the expansion, monomials that correspond to solutions of the problem are linear; the product does not contain squares or other high powers of variables. As an example, the monomial $x_2x_3y_1y_3z_1z_4$ in Equation (2.1) is linear, and corresponds to selecting the second and fourth triples. In the complementary non-solution part, each nonlinear monomial corresponds to a "non-solution," that is, a choice of

Figure 2. A 3-path and a 3-clique. Finding a *k*-clique appears to be much more difficult than finding a *k*-path.

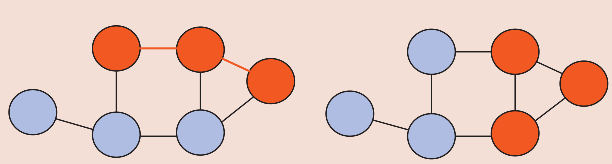
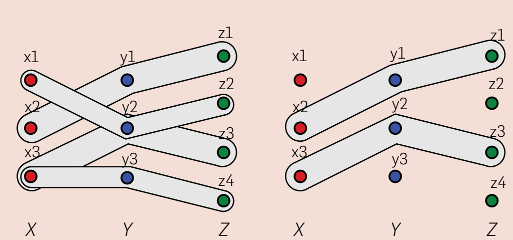


Figure 3. A set of triples and a matching. [Source: Wikipedia]



non-disjoint triples.

This construction can be generalized to any instance of the problem; the only difference would be that if we want to check for a k-matching, we would have to look at the sum-product expansion of the polynomial P_k. To summarize:

Observation. *An instance of the 3D-matching problem contains a matching of size k if and only if the sum-product expansion of the kth encoding polynomial P_k contains a linear monomial.*

Therefore, to solve the 3D matching problem, it would suffice to expand the polynomial P_k into a sum of products and check if the sum contains a linear monomial. However, an $O^*(2^N)$ *dynamic programming* algorithm is known, for $N = |X| + |Y| + |Z|$. On the other hand, the number of possible monomials in N variables is much larger than $O^*(2^N)$. Thus, the simple idea of expanding P_k into a sum of products does not give a good algorithm.

The dynamic programming algebra. Dynamic programming often involves inductive definitions that are tedious to state; the analogue in our algebraic setup is perhaps more intuitive. It can be naturally viewed as a "truncated" expansion of the polynomial P_k into a sum-product form. More concretely, imagine expanding P_k slowly, one multiplication at a time. We observe that monomials containing squared variables can be thrown away as soon as they are formed, because they do not affect the presence of linear monomials in the full sum-product expansion of P_k. There are 2^N linear monomials in N variables. This implies the "truncated" expansion can be carried out in $O^*(2^N)$ time. If there is at least one monomial left in the final truncated sum-product form of P_k, we can conclude the given instance contains a 3D matching of size k. Otherwise, it does not.

More formally, we are computing the sum-product expansion of P_k in an extended *dynamic programming algebra of polynomials* that has all the usual rules for multiplication and addition of polynomials, plus the additional rule that *all squared variables evaluate to zero.* Notice this implies that all nonlinear monomials also evaluate to zero. We can thus recast in algebraic terms our observation regarding linear monomials of P_k:

> *An instance of the 3D-matching problem contains a matching of size k if and only if the kth encoding polynomial P_k is not identical to zero in the dynamic programming algebra.*

In general we can have $N = 3k$, since all elements of $X \cup Y \cup Z$ could participate in a 3D matching. However, if we are merely looking for a 3D matching containing only k triples, then we might set as our target an $O^*(2^{3k})$ time parameterized algorithm.

> **Testing Polynomials**
> Testing whether a given arithmetic expression equals the unique polynomial whose coefficients are all zeros will be a recurring theme in the sequel.

Parameterizing via assignments. The goal here is to describe a first parameterized algorithm for the 3D-matching problem, which will demonstrate the use of assignments of the variables of the polynomial in order to extract information from it.

The problem with "parameterizing" the $O^*(2^N)$ algorithm lies clearly in the number of variables N. Trying to deal with this problem Alon, Yuster and Zwick[2] came up with a method known as color coding. Their solution can be understood as an assignment as follows. Going back to the earlier example, suppose we are interested in finding a matching of size 2. Then, as discussed, all linear monomials in the sum-product expansion of P_2 have degree six; they are products of six distinct variables. Alon et al. proposed the introduction of a new set W containing exactly six variables (more generally $3k$ variables for a k-matching). So, let

$$W = \{w_1, w_2, w_3, w_4, w_5, w_6\}.$$

We then perform a random assignment of the variables in W into the variables in X, Y, Z. We can also think of this assignment as a random coloring of the N variables with k colors. Let us consider one such assignment:

$$\begin{aligned} x_1 &\leftarrow w_1, & x_2 &\leftarrow w_4, \\ x_3 &\leftarrow w_1, & y_1 &\leftarrow w_3, \\ y_2 &\leftarrow w_2, & y_3 &\leftarrow w_5, \\ z_1 &\leftarrow w_6, & z_2 &\leftarrow w_6, \\ z_3 &\leftarrow w_1, & z_4 &\leftarrow w_2. \end{aligned}$$

By just substituting in Equation (2.1), $P_2(X, Y, Z)$ becomes now a polynomial in the variables W:

$$\begin{aligned} P_2(W) = {} & (w_1w_2w_6)^2 + (w_4w_3w_6)^2 \\ & + (w_1w_2w_1)^2 + (w_1w_5w_2)^2 \\ & + 2w_1w_2w_3w_4w_6^2 + 2w_1^3w_2^2w_6 \\ & + 2w_1^2w_2^2w_5w_6 + 2w_1^2w_2w_3w_4w_6 \\ & + 2w_1w_2w_3w_4w_5 + 2w_1^3w_2^2w_5. \end{aligned}$$

It can be seen that monomials in $P_k(X, Y, Z)$ get mapped to monomials in $P_k(W)$. If a monomial in $P_k(X, Y, Z)$ is not linear, the same holds for the corresponding monomial in $P_k(W)$. In contrast, a linear monomial of $P_k(X, Y, Z)$ may or may not survive as a linear monomial $P_k(W)$. In our example all but one linear monomials are mapped to a linear monomial in $P_2(W)$. So detecting a linear monomial in $P_k(W)$ allows us to infer its presence in $P_k(X, Y, Z)$ as well, and consequently the existence of a k-matching in our original problem. The inverse may be not true, as it may be the case that no linear monomial survives the assignment.

We can evaluate $P_k(W)$ in the dynamic programming algebra as we did with $P_k(X, Y, Z)$. Because now W contains only $3k$ variables, the evaluation takes $O^*(2^{3k})$ time. However, the probability that any given linear monomial survives through a random color coding assignment is very small; it can be calculated to be around e^{-3k} where $e \simeq 2.72$. This means one has to try $O(e^{3k})$ random assignments in order to have the monomial survive with an acceptable constant probability, independent from k. So, the overall running time is $O^*((2e)^{3k})$. This is still a factor of e^{3k} away from our $O^*(2^{3k})$ target.

> **Color Coding**
> Color coding has proven to be an extremely useful tool in the design of parameterized algorithms. It has been applied on a diverse list of parameterized problems, including the k-path problem for which it yields an algorithm that runs in $O^*((2e)^k)$ time.

Algebraic Detection

The methods mentioned previously for detecting a linear monomial are essentially *combinatorial*, but we presented them in algebraic guise. Can we find an even faster algorithm if we use a genuinely algebraic method?

A matrix assignment. Earlier, we considered the idea of an assignment into the variables of the encoding polynomial P and its subsequent evaluation in the dynamic programming algebra. The dynamic programming algebra and the color-coding assignment are only one possibility. There are numerous possible algebras and assignments. Matrices in particular seem to offer rich possibilities. An attractive feature of matrix algebra is that the square of a matrix can be 0, offering a genuinely algebraic way of "implementing" the dynamic programming algebra, and specifically its rule that squares of variables should be zero. As a simple example:

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

This leads us to the following list of requirements for a set of random matrices to be used for detecting linear monomials of degree k, that is, products of k distinct variables.

(i) To simplify reasoning about monomials, the matrices must be pairwise commutative, that is, the order in which we multiply them should not affect the value of the product.
(ii) The square of each matrix must be equal to zero. Along with commutativity, this implies that nonlinear monomials will evaluate to zero as well.
(iii) Linear monomials of degree k must "survive" the assignment, that is, evaluate to a non-zero matrix, with constant probability, say at least 1/4. It would be convenient if this non-zero matrix contains only ones in its diagonal.
(iv) Since we are looking for speed, the matrices must be as small as possible, in order to support fast evaluation. Ideally, matrix operations should take time $O^*(2^k)$ or less.

In Koutis[22] an efficient randomized construction of such matrices was given, showing it is possible to satisfy all these requirements if we are willing to relax our notion of zero:

By "zero," we now mean zero modulo 2

We will refer to this as the *The Mod-2 Matrix Fact.*

We would not discuss the technical details of how to prove the Mod-2 Matrix Fact, since all we need here is to understand its algorithmic power and consequences.

Comparing with the color coding method, we see that evaluation of the polynomial takes the same time, $O^*(2^k)$ for detecting a linear monomial with k variables. The gain is in the probability that a linear monomial survives the assignment; from the exponentially small probability e^{-k}, we went to a constant probability of 1/4. This appears to suggest we got rid of the undesired e^k factor and have reached our target of detecting degree-k linear monomials in $O^*(2^k)$ time and therefore a size k 3D matching in $O^*(2^{3k})$ time.

Unwanted cancellations. However, a more careful look reveals we are not done yet due to our relaxed notion of what is zero. It is clear the polynomial $P(X)$ evaluates to zero if it does not contain linear monomials. However, the Mod-2 Matrix Fact does not guarantee $P(X)$ evaluates to non-zero, even if some of its linear monomials evaluate to non-zero. The most significant source of this problem is not in the evaluation, but rather in the polynomial P itself: by taking a look at the sum-product expansion in Equation (2.1) we see that the solution part of P that consists of the linear monomials is already equal to 0 *mod* 2, just because the monomials are multiplied by 2.

The method of fingerprints. We tackle this problem of multiple copies of linear monomials by ensuring each monomial is unique, through the use of a set of "fingerprint" variables $A = \{a_1, a_2, \ldots,\}$. We illustrate the idea on the k-3D-matching polynomial P_2 which we will now define as follows:

$$\begin{aligned} P_2 = &(a_1x_1y_2z_2 + a_2x_2y_1z_1 \\ &+ a_3x_3y_2z_3 + a_4x_3y_3z_4) \\ &* (a_5x_1y_2z_2 + a_6x_2y_1z_1 \\ &+ a_7x_3y_2z_3 + a_8x_3y_3z_4). \end{aligned}$$

All we are doing here is multiplying each occurrence of a triple in P_2 with a "fresh" variable from the set A. Consider then what happens to the two copies of the monomial $x_2x_3y_1y_3z_1z_4$; they are now replaced by two separate linear monomials: $a_2a_8x_2x_3y_1y_3z_1z_4$ and $a_6a_4x_2x_3y_1y_3z_1z_4$.

It appears the introduction of the auxiliary variables comes at the cost of getting linear monomials of higher degree, which would require larger matrices in order to survive through an evaluation, resulting in a slower algorithm.

However, an alternative idea is to use a different assignment for the A variables. For example, we can assign random $\{0, 1\}$ values in the variables of A. In essence we wish to create an odd number of copies for the linear monomials, so that the solution part of the encoding polynomial is not trivially zero modulo 2. In Koutis[22] it was shown this idea is enough for linear monomial detection to go through *in the 3D-matching case*. That is, by randomly assigning matrices to the variables X, Y, Z, and $\{0, 1\}$ values to the fingerprint variables, the polynomial will evaluate to non-zero with constant probability if its sum-product expansion contains a linear monomial and always to zero otherwise.

Target reached. We have designed an $O^*(2^{3k})$ time algorithm for the k-3D-matching problem.

A general algebraic framework. We have come awfully close to a fast parameterized algorithm for a problem much more general than the k-3D-matching problem: that is the detection of degree-k linear monomials in the sum-product expansion of arbitrary polynomials $P(X)$ that are given to us in some concise, "unexpanded" representation. The importance of this problem was established in Koutis[22,24] where it was observed that a fast algorithm for it implies faster algorithms for several parameterized problems, including the k-path problem. So, what do we need to do in order to generalize the method?

All steps described here carry over to arbitrary polynomials, including generating unique linear monomials with an appropriate pre-processing of the polynomial and a placement of fingerprint variables in it. However, we are still stuck with the multiple copies problem because the random $\{0, 1\}$ assignment to the fingerprint variables is not guaranteed to work for any polynomial. We will thus need to generalize to some other type of assignment that works in arbitrary situations, while still being compact and easy to handle computationally.

Here is the solution proposed in Williams.[32] Imagine evaluating $P(X, A)$ in two stages. First we assign random matrices $\bar{X}$ from the Mod-2 Matrix Fact into the X variables and compute $P(\bar{X}, A)$, leaving the A variables unevaluated. The result is a matrix whose diagonal is a polynomial $Q(A)$. The Mod-2 Matrix Fact implies that $Q(A)$ is zero modulo 2 if $P(X)$ does not contain a linear monomial. If $P(X)$ does contain a linear monomial then $Q(A)$ is non-zero with probability at least 1/4. This consideration crystallizes the problem: we wish to be able to test whether $Q(A)$ is identical to zero modulo 2, and we would like to do so by evaluating $Q(A)$ on some "compact" assignment to its variables. This problem, however, has been studied and solved earlier. It is known as *identity testing*.

A solution to identity testing, known as Schwartz–Zippel Lemma,[26] is simple and intuitive. The reason we picked the values $\{0, 1\}$ in our assignment for the 3D-matching polynomial is that with them we can perform multiplication and addition that respects modulo 2 arithmetic. But there are other algebraic "fields" that allow us the same kind of operations, with the bonus fact they are larger, in the sense they contain more values. If instead of $\{0, 1\}$ we pick a field consisting of $O(k)$ values the number of possible assignments to the fingerprint variables by far outnumbers the number of possible roots of $Q(A)$, that is, the number of assignments that make it evaluate to zero. Hence a random assignment from this larger field will result with high probability in $Q(A)$ evaluating to a non-zero value in the field. This allows us to claim the following result.

Fast linear monomial detection. The problem of detecting a degree-k square-free monomial in an arbitrary polynomial $P(X)$ can be solved in $O^*(2^k)$ time.

Linear Monomial Detection

The parameterized linear monomial detection problem along with the algorithm for its solution provide a general framework for the solution of parameterized problems. Most parameterized algorithms that rely on color coding can be accelerated by a factor of $O^*(e^k)$, by simply applying linear monomial detection, as we saw in the matching problem. In fact the acceleration is precisely e^k in most of these cases, including the k-path problem. This yields an $O^*(2^k)$ time algorithm for the k-path problem.

Breaking Barriers

A faster parameterized algorithm for linear monomial detection would have a tremendous impact, as it would imply faster algorithms not only for many parameterized problems, but also for the corresponding non-parameterized NP-hard problems. Thus, an intriguing and natural question is whether the problem can be solved faster by evaluating the input polynomial over a more exotic algebra supporting faster operations. Unfortunately, the question has been answered in the negative; it is impossible to find a better algebra.[24]

This algebraic barrier suggests new techniques are required for further progress in the linear detection problem, if progress is possible at all. However, one can be more optimistic about specific problems. Linear monomial detection is very general and completely agnostic to combinatorial properties of the underlying problem, so taking advantage of specific problem properties may sometimes get around the algebraic barrier. On the other hand, attempting to make progress on well-studied NP-complete problems, such as the Hamiltonian path problem or the 3D-matching problem, one faces a perhaps more significant psychological barrier: a lack of progress in nearly 50 years.

In brilliant work, Andreas Björklund broke the psychological barrier with an $O^*(2^{N/3})$ time algorithm for the "exact" X3D-matching problem[5] and an astonishing $O^*(1.657^n)$ time algorithm for the Hamiltonicity.[4] These running times were later almost matched by parameterized algorithms.[6] The reader can find in Fomin et al.[14] an excellent exposition of the algorithm for the Hamiltonian path problem for bipartite graphs.

Central to Björklund's work are sharper algebraic tools that draw from the large pool of algebraic combinatorics (for example, Royle[28]) to produce lower degree polynomials. More crucially, Björklund proposed the idea of relaxing the method of fingerprints in order to exploit cancellations of pairs of non-solution monomials. Here, we will see how these ideas got applied in the case of the X3D-matching problem.

Counting weighted matchings Mod 2. Consider a restriction of our 3D-matching problem to a 2D-matching problem in which we are given pairs consisting of a captain and a second officer, rather than triples. We will assume the two sets X and Y are of equal size n.

The problem has a graph representation as shown in Figure 4. A solution of n edges/pairs that covers all vertices of the graph is called a *perfect matching.*

The perfect matching problem can be solved in polynomial time. But *counting* the number of perfect matchings is known to be a hard problem; it is as hard as counting the number of solutions for any NP-complete problem.[30] In fact, it is also hard to compute the number of perfect matchings modulo any prime p, except when $p = 2$. If $p = 2$ the number of perfect matchings modulo 2 is equal to the determinant of the incidence matrix of the bipartite graph.[28] This key fact is going to play a significant role, because the matrix determinant can be computed with a polynomial number of arithmetic operations.

To define the incidence matrix A, we arbitrarily number the nodes in X with numbers from 1 to n, and do the same with the nodes in Y, as shown in Figure 4. The entry $A(i, j)$ is 1 if node $i \in X$ is connected to node $j \in Y$, and 0 otherwise.

Given this definition, if N_2 denotes the number of perfect matchings, we have what we call the *Matching Lemma:*

$$N_2 \text{ } mod \text{ } 2 = Determinant\,(A) \text{ } mod \text{ } 2.$$

We now consider an extension of the above lemma that will be particularly useful. The perfect matching counting problem has a natural generalization to edge-labeled multi-graphs, where (i) we allow multiple "parallel" edges between any two given nodes and (ii) each edge e is labeled with a unique monomial l_e.

The signature of a matching μ is defined to be the monomial formed by taking the product of the labels of the edges in the matching μ. For example, the signature of the matching in Figure 4 is $l_{1,4}l_{2,1}l_{3,2}l_{4,3}$. More generally,

$$sig(\mu) \triangleq \prod_{e \in \mu} l_e.$$

Given a labeling of the edges we can extend N_2 to a solution polynomial $N_2(L)$, which is simply the sum of the signatures of all perfect matchings, such as if M is the set of perfect matchings in the graph, then

$$N_2(L) \triangleq \sum_{\mu \in M} sig(\mu).$$

Finally, we can extend the incidence matrix A to a matrix A_L over L in a natural way: if nodes i and j are connected by one or more edges, we let $A_L(i,j)$ be the *sum* of their labels. We have the following Generalized Matching Lemma:

$$N_2(L) \bmod 2 = Determinant(A_L) \bmod 2. \quad (4.2)$$

Monomial detection for X3D-matching. We now return to the X3D-matching problem, which is the 3D-matching problem augmented with the constraint that $|X| = |Y| = |Z| = n$. In this case we want to decide if there is a solution that consists of exactly $n = N/3$ triples, that is, a solution that covers all elements of X, Y, Z.

The algorithm involves a transformation of the X3D matching problem into a labeled matching instance. The sets X and Y define the two parts of the bipartite graph. In order to determine the labels, we use two sets of variables, Z and U. Each destination corresponds to a distinct variable in Z and each triple corresponds to a distinct variable in U.

To see now how we form the labeled bipartite graph we give an example that illustrates the edges between two given vertices of the graph, along with their labels. Suppose the X3D-matching instance contains the triples

$$\{W, K, \text{SJU}\}, \{W, K, \text{SFO}\}, \{W, K, \text{PIT}\}, \quad (4.3)$$

and no other triples with $\{W, K\}$ as a captain and second officer. Also suppose we have associated with these triples the variables u_1, u_2, u_3, respectively. Then, nodes W and K will be connected by three edges, labeled respectively with the monomials u_1z_{SJU}, u_2z_{SFO}, and u_3z_{PIT}.

Consider now the polynomial $N_2(Z, U) \bmod 2$. Because each edge of the bipartite graph corresponds to a triple, any monomial of $N_2(Z, U) \bmod 2$ corresponds to the selection of n triples, because it is the signature of a matching. By definition of $N_2(Z, U) \bmod 2$ these n triples cover exactly and without overlaps the sets X and Y. In the case there is an overlap in destinations, the monomial contains a squared variable, corresponding to that destination. On the other hand if a set of triples forms a 3D matching, the monomial is linear with respect to the variables in Z because no destination appears twice. So, assigning the random matrices from the Mod-2 Matrix Fact to the variables in Z we zero-out the nonlinear monomials. In addition, we observe the coefficient of each monomial in $N_2(L) \bmod 2$ is 1, because the U variables specify each set of n triples. Thus the U variables are our fingerprint variables and we will assign to them values from an appropriate algebraic field of size $O(n)$. With these assignments, the outcome of the evaluation will be non-zero with a good probability if and only if $N_2(L)$ contains a monomial corresponding to a solution of the instance. Hence we can detect an exact 3D matching if there exists one. The running time of the algorithm is $O^*(2^n)$ because the degree of $N_2(Z, U)$ in terms of the Z variables is n.

Determinants and cycles. The monomials that are canceled when the determinant is computed modulo 2 as in Equation (4.2) correspond to cycle covers. A cycle is a path plus one edge that closes the loop. A cycle cover is a set of cycles in the graph such that each vertex participates in exactly one cycle. In particular, a Hamiltonian cycle, that is, a cycle containing all the vertices, is a cycle cover. A perfect matching is also a cycle cover albeit consisting only of "degenerate" cycles, that is, edges. Cycle covers that contain at least one non-degenerate cycle come in pairs basically because each such cycle can be traversed in two possible ways: clockwise and counterclockwise. Björklund's Hamiltonicity algorithm, which actually targets Hamiltonian cycles rather than paths, breaks this symmetry by introducing direction to some edges in the graph. His algorithm still relies crucially on the remaining symmetries and cancellations. It remains open whether Hamiltonicity has a faster than $O^*(2^n)$ time algorithm for general directed graphs.

Exploiting Cancellations

The algebraization for X3D matching is relaxed by design, as not all monomials in the determinant polynomial have unique fingerprints. Instead, many monomials that correspond to graph structures other than matchings happen to come in pairs, and computing the determinant modulo 2 exploits that. The idea of treating modulo 2 arithmetic as a resource rather than as a nuisance was also used for Hamiltonicity and later works.

More Recent Advances

Since its appearance, the general framework of algebraic fingerprints with modulo computations has been used in the design of faster algorithms for several parameterized problems. Examples include: finding subgraphs that are more complicated than paths,[16,31] finding functional motifs in biological networks,[8,18,23] finding the shortest cycle through specified nodes in a given graph,[7] and the repetition-free longest common subsequence problem for strings, with applications in computational biology.[9] Among the many examples, we highlight remarkable progress in algorithms for NP-complete problems on graphs, parameterized by the so-called *treewidth* of the input graph.

Several NP-complete problems on graphs are tractable if the graph is a tree, that is, if it contains no cycles. The NP-completeness of graph problems is usually proved via the construction of very intricate graphs that are arguably artificial comparing to graphs arising in practical situations. In particular, real-world graphs often have a more tree-like structure.

The notion of treewidth offers a way to quantify how much a given graph "deviates" from a being tree. An example is given in Figure 5. The vertices of the graph are arranged in a tree structure. Each tree node lists copies of some of the graph vertices. Each edge of the graph connects two vertices that are listed together at some tree node. In addition, any given graph vertex can appear only in contiguous tree nodes. The treewidth *tw* of a graph is defined as the maximum number of graph vertices hosted in a tree node; in our example $tw = 3$. As a more general example, the natural class of *planar* graphs, that is, graphs that can be drawn on the plane without crossings has treewidth $O(\sqrt{n})$, where *n* is the number of vertices.

Cygan et al.[11] gave faster algorithms for many graph problems parameterized by treewidth. Previous algorithms had running time of the form $O^\star(tw^{tw})$, while the new algorithms have dramatically improved running times of the form $O^\star(c^{tw})$ for small constants *c*; for example the Hamiltonian path cycle can be now solved in $O^\star(4^{tw})$ time.

Open Problems

The algorithms in this article are all randomized; on any given run they have a small probability of reporting that an instance does not have a solution, when the opposite is true. But running them $O(\log(1/p))$ times will make the overall probability of failure smaller than *p*, for any $p>0$. The fastest known *deterministic* algorithm for the *k*-path problem requires $O^\star(2.85^k)$ time.[15] Finding a deterministic algorithm that solves the problem in $O^\star(2^k)$ time remains an open problem. The way things stand, it would seem to require a deterministic version of the Schwartz–Zippel Lemma: a deterministic polynomial-time algorithm for testing whether a polynomial given as an arithmetic circuit is identically zero.

The method of algebraic fingerprints can be adapted to solve the weighted *k*-path problem; here the goal is to find a *k*-path of minimum total cost, assuming a cost is attached to every edge of the graph. The running time is $O^\star(2^k W)$ where *W* is the largest cost associated with an edge in the graph. On the other hand, dynamic programming can handle the problem in $O^\star(2^n \log W)$ time, when $k = n$. It is reasonable to conjecture that *k*-path has an $O^\star(2^k \log W)$ algorithm. It should be noted though that, unlike the algebraic fingerprints method that can be implemented in memory of size polynomial in *n*, color coding for weighted *k*-path requires $O^\star(2^k)$ memory, which can be a very limiting factor in practice.

Dynamic programming can also be used to *count* the number of Hamiltonian paths. Counting the number of *k*-paths exactly is probably not FPT,[13] but color coding can be used to count *k*-paths *approximately*, in $O^\star((2e)^k)$ time.[1] This stands as the fastest known algorithm for this problem, but again it is reasonable to conjecture there is an $O^\star(2^k)$ algorithm for approximate counting.

Solving any of these open questions may require fresh ideas that will start a new cycle of discovery.

Acknowledgments

I. Koutis is supported by NSF CAREER award CCF-1149048. R. Williams is supported by a David Morgenthaler II Faculty Fellowship, NSF CCF-1212372, a Sloan Fellowship, and a Microsoft Research Fellowship. The final phase of this work occurred at the Simons Institute for the Theory of Computing. C

References

1. Alon, N., Dao, P., Hajirasouliha, I., Hormozdiari, F., Sahinalp, S.C. Biomolecular network motif counting and discovery by color coding. In *Proceedings 16th International Conference on Intelligent Systems for Molecular Biology (ISMB)* (Toronto, Canada, July 19–23, 2008), 241–249.
2. Alon, N., Yuster, R., Zwick, U. Color coding. *J. ACM 42*, 4 (1995), 844–856.
3. Bellman, R. Dynamic programming treatment of the travelling salesman problem. *J. ACM 9* (1962) 61–63.
4. Björklund, A. Determinant sums for undirected hamiltonicity. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010* (Las Vegas, Nevada, USA, October 23–26, 2010). IEEE Computer Society, 173–182.
5. Björklund, A. Exact covers via determinants. In *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010*, J. Marion and T. Schwentick, eds. Volume 5 of *LIPIcs* (Nancy, France, March 4–6, 2010). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 95–106.
6. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M. Narrow sieves for parameterized paths and packings. *CoRR*, abs/1007.1161, 2010.
7. Björklund, A., Husfeld, T., Taslaman, N. Shortest cycle through specified elements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012*, Y. Rabani, ed. (Kyoto, Japan, January 17–19, 2012). SIAM, 1747–1753.
8. Björklund, A., Kaski, P., Kowalik, L. Probably optimal graph motifs. In *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013*, N. Portier and T. Wilke, eds. Volume 20 of *LIPIcs* (Kiel, Germany, February 27–March 2, 2013). Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 20–31.
9. Blin, G., Bonizzoni, P., Dondi, R., Sikora, F. On the parameterized complexity of the repetition free longest common subsequence problem. *Inform. Process. Lett. 112*, 7 (2012), 272–276.
10. Chen, J., Lu, S., Sze, S.-H., Zhang, F. Improved algorithms for path, matching, and packing problems. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2007), 298–307.
11. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O. Solving connectivity problems parameterized by treewidth in single exponential time. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, R. Ostrovsky, ed. (Palm Springs, CA, USA, October 22–25, 2011), 150–159.
12. Downey, R.G. and Fellows, M.R. *Parameterized Complexity*. Texts in Computer Science. Springer, 1999.
13. Flum, J., Grohe, M. The parameterized complexity of counting problems. *SIAM J. Comput. 33*, 4 (2004), 892–922.
14. Fomin, F.V., Kaski, P. Exact exponential algorithms. *Commun. ACM 56*, 3 (Mar. 2013), 80–88.
15. Fomin, F.V., Lokshtanov, D., Panolan, F., Saurabh, S. Representative sets of product families. In *Proceedings of the Algorithms – ESA 2014 – 22th Annual European Symposium, Wroclaw, Poland, September 8–10, 2014*, A.S. Schulz and D. Wagner, eds. Volume 8737 of *Lecture Notes in Computer Science* (2014). Springer, 443–454.
16. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S., Rao, B.V.R. Faster algorithms for finding and counting subgraphs. *J. Comput. Syst. Sci. 78*, 3 (2012), 698–706.
17. Fortnow, L. The status of the P versus NP problem. *Commun. ACM 52*, 9 (2009), 78–86.
18. Guillemot, S., Sikora, F. Finding and counting vertex-colored subtrees. *Algorithmica 65*, 4 (2013), 828–844.
19. Held, M., Karp, R. A dynamic programming approach to sequencing problems. *J. Soc. Indust. Appl. Math. 10* (1962) 196–210.
20. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R.E. Miller and J.W. Thatcher, eds. (1972). Plenum Press, 85–103.
21. Koutis, I. A faster parameterized algorithm for set packing. *Inform. Process. Lett. 94*, 1 (2005), 4–7.
22. Koutis, I. Faster algebraic algorithms for path and packing problems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming Colloquium, ICALP 2008, Part I : Tack A: Algorithms, Automata, Complexity, and Games*, L. Aceto, I. Damgård, L.A. Goldberg, M.M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, eds. Volume 5125 of *Lecture Notes in Computer Science* (Reykjavik, Iceland, July 7–11, 2008). Springer-Verlag, 575–586.
23. Koutis, I. Constrained multilinear detection for faster functional motif discovery. *Inf. Process. Lett. 112*, 22 (2012), 889–892.
24. Koutis, I., Williams, R. Limits and applications of group algebras for parameterized problems. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP '09 (Berlin, Heidelberg, 2009). Springer-Verlag, 653–664.
25. Monien, B. How to find long paths efficiently. *Ann. Dis. Math. 25* (1985), 239–254.
26. Motwani, R., Raghavan, P. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.
27. Rosamond, F., Fellows, M. Table of FPT Races. http://fpt.wikidot.com/fpt-races/.
28. Royle, G., Godsil, C. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer Verlag, 1997.
29. Scott, J., Ideker, T., Karp, R.M., Sharan, R. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol. 13*, 2 (2006), 133–144.
30. Valiant, L.G. The complexity of enumeration and reliability problems. *SIAM J. Comput.* 8, 3 (1979), 410–421.
31. Williams, R. Finding paths of length *k* in $O^*(2^k)$ time. *Inf. Process. Lett. 109* (February 2009), 315–318.
32. Williams, V.V., Wang, J.R., Williams, R.R., Yu, H. Finding four-node subgraphs in triangle time. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*, P. Indyk, ed. (San Diego, CA, USA, January 4–6, 2015). SIAM, 1671–1680.

Ioannis Koutis (i.koutis@gmail.com) is an assistant professor at the University of Puerto Rico, Rio Piedras.

Ryan Williams (rrw@cs.stanford.edu) is an assistant professor at Stanford University, Stanford, CA.

Watch the authors discuss their work in this exclusive *Communications* video. http://cacm.acm.org/videos/algebraic-fingerprints-for-faster-algorithms

research highlights

DOI:10.1145/2845910

To view the accompanying paper, visit doi.acm.org/10.1145/2845648 rh

Technical Perspective
High-Performance Virtualization: Are We Done?

By Steve Hand

WHOLE SYSTEM VIRTUALIZATION allows us to package software systems inside self-contained virtual machines, each running its own operating system along with a collection of applications and services. The use of "virtual hardware" allows agile software management of machine deployment, and the provision of strong resource isolation between co-located virtual machines enables modern cloud computing platforms. However, these benefits do not come entirely for free: the additional level of indirection can lead to increased runtime overheads and reduced performance.

The past decade has seen huge advances in tackling this problem. CPU virtualization overheads due to dynamic binary rewriting and memory virtualization overheads due to shadow page tables were reduced by a combination of clever algorithms and hardware assistance. A key remaining problem has been I/O virtualization, and most notably tackling the challenges introduced by high-speed networks running at 10Gb/s or 40Gb/s. The following paper shows how to enable a virtual machine to attain "bare metal" performance from high-speed network interface cards (NICs).

Their starting point is to use direct device assignment (sometimes referred to as "PCI pass-through"). The idea here is to dedicate a NIC to a virtual machine, and allow it to access the device registers directly. This means the device driver running in the virtual machine will be able to program DMA transfers to send and receive packets, just like it would if running on a real machine. Configuring an IOMMU to disallow transfers to or from non-owned physical memory ensures security. Getting the virtualization layer out of the way for data transfer is a big win, and goes a long way to reducing the performance overheads. But it turns out there is another, rather surprising, problem: interrupt processing.

The ELI idea is a neat one, and is applicable to more than just network interface cards. With some tweaks, the approach should be able to scale to multiple cores, devices, and virtual machines.

As device speeds increase, the number of packets per second increases too, leading to a high rate of device interrupts. There are various techniques to ameliorate high-interrupt load, such as using larger packets or batching interrupt generation, but these come with potentially undesirable side effects, and even then only partly mitigate the problem. For example, one of the experiments in the paper shows that even with adaptive interrupt batching, a NIC can easily generate many tens of thousands of interrupts per second. This is already a problem for regular machines but it is even worse for virtual machines because as it turns out they incur far higher interrupt processing costs.

As the authors explain, the standard practice for handling a device interrupt that occurs while running in a virtual machine requires two "exits": context-switches from the guest-operating mode into the host-operating mode. The overhead from these exits (and their matched re-entries into guest-operating mode) can lead to performance degradation of 30%–50% for network intensive workloads. This observation leads directly to the authors' idea: Exitless Interrupts (ELI).

Their solution comes in two parts. The first technique is to configure the system to deliver *all* interrupts directly to the virtual machine running in guest-operating mode, but arrange for those not intended for direct consumption to immediately exit back to the host. The second technique is to allow the guest-operating system running in the virtual machine to directly acknowledge to the hardware it has handled the interrupt. There is a lot of cleverness required to make this work safely and transparently—read the paper for more details! But the key take-away is these two techniques, in combination, allow a virtual machine to attain close to 100% of bare-metal performance when using a 10Gb/s NIC.

The ELI idea is a neat one, and is applicable to more than just network interface cards. With some tweaks and additional features, the general approach should be able to scale to multiple cores, multiple devices, and multiple virtual machines. And if so, perhaps we can finally say that virtualization performance is a solved problem … for now.

Steve Hand (sthand@google.com) is senior staff engineer at Google, Mountain View, CA.

DOI:10.1145/2845648

Bare-Metal Performance for Virtual Machines with Exitless Interrupts

By Nadav Amit, Abel Gordon, Nadav Har'El, Muli Ben-Yehuda, Alex Landau, Assaf Schuster, and Dan Tsafrir

Abstract

Direct device assignment enhances the performance of guest virtual machines by allowing them to communicate with I/O devices without host involvement. But even with device assignment, guests are still unable to approach bare-metal performance, because the host intercepts all interrupts, including those generated by assigned devices to signal to guests the completion of their I/O requests. The host involvement induces multiple unwarranted guest/host context switches, which significantly hamper the performance of I/O intensive workloads. To solve this problem, we present ExitLess Interrupts (ELI), a software-only approach for handling interrupts within guest virtual machines *directly* and *securely*. By removing the host from the interrupt handling path, ELI improves the throughput and latency of unmodified, untrusted guests by 1.3×–1.6×, allowing them to reach 97–100% of bare-metal performance even for the most demanding I/O-intensive workloads.

1. INTRODUCTION

I/O activity is a dominant factor in the performance of virtualized environments,[17, 25] motivating *direct device assignment* where the host assigns physical I/O devices directly to guest virtual machines. Examples of such devices include disk controllers, network cards, and GPUs. Direct device assignment provides superior performance than alternative I/O virtualization approaches, because it almost entirely removes the host from the guest's I/O path. Without direct device assignment, I/O-intensive workloads might suffer unacceptable performance degradation.[17, 19, 25] Still, on x86 CPUs (the most popular platform for virtualization), direct assignment *alone* does not allow I/O-intensive workloads to approach bare-metal (nonvirtual) performance[6, 9, 16, 25]; by our measurements, such workloads achieve only 60–65% of bare-metal performance. We find that nearly the *entire* performance difference is induced by interrupts of assigned devices.

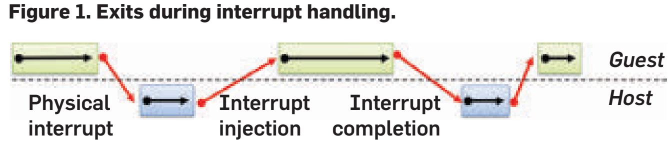
I/O devices generate interrupts to notify the CPU of I/O operations' completion. In virtualized settings, each device interrupt triggers a costly *exit*,[1, 6] causing the guest to be suspended and the host to be resumed, regardless of whether or not the device is assigned. The host first signals to the hardware the completion of the physical interrupt as mandated by the x86 specification. It then injects a corresponding (virtual) interrupt to the guest and resumes the guest's execution. The guest in turn handles the virtual interrupt and, like the host, signals completion, believing that it directly interacts with the hardware. This action triggers yet another exit, prompting the host to emulate the completion of the virtual interrupt and to resume the guest again. The chain of events for handling interrupts is illustrated in Figure 1.

The guest/host context switches caused by interrupts induce a tolerable overhead for non-I/O-intensive workloads, a fact that allowed some previous virtualization studies to claim they achieved bare-metal performance.[5, 14] But our measurements indicate that this overhead quickly ceases to be tolerable, adversely affecting guests that require throughput of as little as 50 Mbps. Notably, previous studies improved virtual I/O by relaxing protection[13, 14] or by modifying guests,[5] whereas we focus on the most challenging virtualization scenario of untrusted and unmodified guests.

Many previous studies identified interrupts as a major source of overhead,[6, 15] and many proposed techniques to reduce it, both in bare-metal settings[10, 21, 23, 26] and in virtualized settings[3, 9, 16, 25]. In principle, it is possible to tune devices and their drivers to generate fewer interrupts, thereby reducing the related overhead. But doing so in practice is far from trivial[22] and can adversely affect both latency and throughput.

Our approach rests on the observation that the high interrupt rates experienced by a core running an I/O-intensive guest are mostly generated by devices assigned to the guest. Indeed, we measure rates of over 150K physical interrupts per second, even while employing standard techniques to reduce the number of interrupts, such as *interrupt coalescing*[3, 21, 26] and *hybrid polling*.[10, 23] As noted, the resulting guest/host context switches are nearly exclusively responsible for the inferior performance relative to bare metal. To eliminate these switches, we propose ExitLess Interrupts (ELI), a software-only approach for handling physical interrupts directly within the guest in a secure manner.

Figure 1. Exits during interrupt handling.



A full version of this paper is available in *Proceedings of ACM Architectural Support for Programming Languages (ASPLOS) 2012*.

With ELI, physical interrupts are delivered directly to guests, allowing them to process their devices' interrupts without host involvement; ELI makes sure that each guest forwards all other interrupts to the host. With x86 hardware, interrupts are delivered using a software-controlled table of pointers to functions, such that the hardware invokes the kth function whenever an interrupt of type k fires. Instead of utilizing the guest's table, ELI maintains, manipulates, and protects a "shadow table," such that entries associated with assigned devices point to the guest's code, whereas the other entries are set to trigger an exit to the host.

We experimentally evaluate ELI with micro- and macro-benchmarks. Our baseline configuration employs standard techniques to reduce (coalesce) the number of interrupts, demonstrating ELI's benefit beyond the state-of-the-art. We show that ELI reduces CPU overheads that limit the attainable throughput, and thereby it improves the throughput and latency of guests by 1.3×–1.6×. Notably, whereas I/O-intensive guests were so far limited to 60–65% of bare-metal throughput, with ELI they reach performance that is within 97–100% of the optimum. Consequently, ELI makes it possible to, for example, consolidate traditional data-center workloads that nowadays remain nonvirtualized due to unacceptable performance loss.

2. MOTIVATION AND RELATED WORK

For the past several decades, interrupts have been the main method by which hardware devices can send asynchronous events to the operating system.[7] The main advantage of using interrupts to receive notifications from devices over polling them is that the processor is free to perform other tasks while waiting for an interrupt. This advantage applies when interrupts happen relatively infrequently, as was the case until high performance storage and network adapters came into existence. With these devices, the CPU can be overwhelmed with interrupts, leaving no time to execute code other than the interrupt handler.[18] When the operating system is run in a guest, interrupts have a higher cost, since every interrupt causes multiple exits.[1,6] ELI eliminates most of these exits and their associated overhead.

In the remainder of this section we introduce the existing approaches to reduce the overheads induced by interrupts, and we highlight the novelty of ELI in comparison to these approaches. We subdivide the approaches into two categories.

2.1. Generic interrupt handling approaches

We now survey approaches that apply equally to bare metal and virtualized environments.

Polling disables interrupts entirely and polls the device for new events at regular intervals. The benefit is that handling device events becomes synchronous, allowing the operating system to decide when to poll and thus limit the number of handler invocations. The drawbacks are added latency, increased power consumption (since the processor cannot enter an idle state), and wasted cycles when no events are pending. If polling is done on a different core, latency is improved, but a core is wasted.

A **hybrid** approach for reducing interrupt-handling overhead is to switch dynamically between using interrupts and polling.[10,18] Linux uses this approach by default through the NAPI mechanism.[23] Switching between interrupts and polling does not always work well in practice, partly due to the complexity of predicting the number of interrupts a device will issue in the future.

Another approach is **interrupt coalescing**,[3,21,26] in which the OS programs the device to send one interrupt in a time interval or one interrupt per several events, as opposed to one interrupt per event. As with the hybrid approaches, coalescing delays interrupts and hence might increase latency[15] and burst TCP traffic.[26] Deciding on the right model and parameters for coalescing is particularly complex when the workload runs within a guest.[9] Getting it right for a wide variety of workloads is hard if not impossible.[3,22] Unlike coalescing, ELI does not reduce the number of interrupts; instead it streamlines the handling of interrupts targeted at virtual machines. Coalescing and ELI are therefore complementary, as we show in Section 5.4: coalescing reduces the number of interrupts, and ELI reduces their cost.

All evaluations in Section 5 were performed with the default Linux configuration, which combines the hybrid approach (via NAPI) and coalescing.

2.2. Virtualization-specific approaches

Using an emulated or paravirtual[5] device provides much flexibility on the host side, but its performance is much lower than that of device assignment, not to mention bare metal. Liu[16] shows that device assignment of SR-IOV devices can achieve throughput close to bare metal at the cost of as much as 2× higher CPU utilization. He also demonstrates that interrupts have a great impact on performance and are a major expense for both the transmit and receive paths.

There are software techniques[2] to reduce the number of exits by finding blocks of exiting instructions and exiting only once for the whole block. These techniques can increase the efficiency of running a virtual machine when the main reason for the overhead is in the guest code. When the reason is in external interrupts, such as for I/O intensive workloads with SR-IOV, such techniques do not alleviate the overhead.

Dong et al.[9] discuss a framework for implementing SR-IOV support in the Xen hypervisor. Their results show that SR-IOV can achieve line rate with a 10Gbps network interface controller (NIC). However, the CPU utilization is 148% of bare metal. In addition, this result is achieved using adaptive interrupt coalescing, which increases I/O latency.

Several studies attempted to reduce the aforementioned extra overhead of interrupts in virtual environments. vIC[3] discusses a method for interrupt coalescing in virtual storage devices and shows an improvement of up to 5% in a macro-benchmark. Their method uses the number of "commands in flight" to decide how many to coalesce. Therefore, as the authors say, this approach cannot be used for network devices due to the lack of information on commands (or packets) in flight. Dong et al.[8] use virtual interrupt coalescing via polling in the guest and receive side scaling to reduce network overhead in a paravirtual environment. Polling has its drawbacks, as discussed above, and ELI improves the more performance-oriented device assignment environment.

NoHype[13] argues that modern hypervisors are prone to attacks by their guests. In the NoHype model, the hypervisor is a thin layer that starts, stops, and performs other administrative actions on guests, but is not otherwise involved. Guests use assigned devices and interrupts are delivered directly to guests. No details of the implementation or performance results are provided. Instead, the authors focus on describing the security and other benefits of the model.

3. X86 INTERRUPT HANDLING

To put ELI's design in context, we begin with a short overview of how interrupt handling works on x86 today.

3.1. Interrupts in bare-metal environments

x86 processors use interrupts and exceptions to notify system software about incoming events. Interrupts are asynchronous events generated by external entities such as I/O devices; exceptions are synchronous events—such as page faults—caused by the code being executed. In both cases, the currently executing code is interrupted and execution jumps to a pre-specified interrupt or exception handler.

x86 operating systems specify handlers for each interrupt and exception using an architected in-memory table, the Interrupt Descriptor Table (IDT). This table contains up to 256 entries, each entry containing a pointer to a handler. Each architecturally-defined exception or interrupt has a numeric identifier—an exception number or interrupt *vector*—which is used as an index to the table. The operating systems can use one IDT for all of the cores or a separate IDT per core. The operating system notifies the processor where each core's IDT is located in memory by writing the IDT's virtual memory address into the Interrupt Descriptor Table Register (IDTR). Since the IDTR holds the virtual (not physical) address of the IDT, the OS must always keep the corresponding address mapped in the active set of page tables. In addition to the table's location in memory, the IDTR holds the table's size.

When an external I/O device raises an interrupt, the processor reads the current value of the IDTR to find the IDT. Then, using the interrupt vector as an index to the IDT, the CPU obtains the virtual address of the corresponding handler and invokes it. Further interrupts may or may not be blocked while an interrupt handler runs.

System software needs to perform operations such as enabling and disabling interrupts, signaling the completion of interrupt handlers, configuring the timer interrupt, and sending interprocessor interrupts (IPIs). Software performs these operations through the Local Advanced Programmable Interrupt Controller (LAPIC) interface. The LAPIC has multiple registers used to configure, deliver, and signal completion of interrupts. Signaling the completion of interrupts, which is of particular importance to ELI, is done by writing to the end-of-interrupt (EOI) LAPIC register. The newest LAPIC interface, x2APIC,[11] exposes its registers using model specific registers (MSRs), which are accessed through "read MSR" and "write MSR" instructions. Previous LAPIC interfaces exposed the registers only in a predefined memory area which is accessed through regular load and store instructions.

3.2. Interrupts in virtual environments

x86 hardware virtualization[11] provides two modes of operation, *guest mode* and *host mode*. The host, running in host mode, uses guest mode to create new contexts for running guest virtual machines. Once the processor starts running a guest, execution continues in guest mode until some sensitive event forces an exit back to host mode. The host handles any necessary events and then resumes the execution of the guest, causing an entry into guest mode. These exits and entries are the primary cause of virtualization overhead,[1, 6, 19] which is particularly pronounced in I/O intensive workloads.[16, 20, 24] It comes from the processor cycles spent switching between contexts, the time spent in host mode to handle the exit, and the resulting cache pollution.

This work focuses on running unmodified and untrusted operating systems. On the one hand, unmodified guests are not aware they run in a virtual machine, and they expect to control the IDT exactly as they do on bare metal. On the other hand, the host cannot easily give untrusted and unmodified guests control of each core's IDT. This is because having full control over the physical IDT implies total control of the core. Therefore, x86 hardware virtualization extensions use a different IDT for each mode. Guest mode execution on each core is controlled by the guest IDT and host mode execution is controlled by the host IDT. An I/O device can raise a physical interrupt when the CPU is executing either in host mode or in guest mode. If the interrupt arrives while the CPU is in guest mode, the CPU forces an exit and delivers the interrupt to the host through the host IDT.

Guests receive virtual interrupts, which are not necessarily related to physical interrupts. The host may decide to inject the guest with a virtual interrupt because the host received a corresponding physical interrupt, or the host may decide to inject the guest with a virtual interrupt manufactured by the host. The host injects virtual interrupts through the guest IDT. When the processor enters guest mode after an injection, the guest receives and handles the virtual interrupt.

During interrupt handling, the guest will access its LAPIC. Just like the IDT, full access to a core's physical LAPIC implies total control of the core, so the host cannot easily give untrusted guests access to the physical LAPIC. For guests using the first LAPIC generation, the processor forces an exit when the guest accesses the LAPIC memory area. For guests using x2APIC, the host traps LAPIC accesses according to an MSR bitmap, which specifies the sensitive MSRs that cannot be accessed directly by the guest. When the guest accesses sensitive MSRs, execution exits back to the host. In general, x2APIC registers are considered sensitive MSRs.

3.3. Interrupts from assigned devices

The key to virtualization performance is for the CPU to spend most of its time in guest mode, running the guest, and not in the host, handling guest exits. I/O device emulation and paravirtualized drivers[5] incur significant overhead for I/O intensive workloads running in guests.[6, 16] The overhead is incurred by the host's involvement in its guests' I/O paths for programmed I/O (PIO), memory-mapped I/O (MMIO), direct memory access (DMA), and interrupts.

Direct device assignment is the best performing approach for I/O virtualization[9, 16] because it removes some of the host's involvement in the I/O path. With device assignment, guests are granted direct access to assigned devices. Guest I/O operations bypass the host and are communicated directly to devices. As noted, device DMA also bypasses the host; devices perform DMA accesses to and from guest memory directly. Interrupts generated by assigned devices, however, still require host intervention.

In theory, when the host assigns a device to a guest, it should also assign the physical interrupts generated by the device to that guest. Unfortunately, current x86 virtualization only supports two modes: either all physical interrupts on a core are delivered to the currently running guest, or all physical interrupts in guest mode cause an exit and are delivered to the host. An untrusted guest may handle its own interrupts, but it must not be allowed to handle the interrupts of the host and the other guests. Consequently, before ELI, the host had no choice but to configure the processor to force an exit when *any* physical interrupt arrives in guest mode. The host then inspected the interrupt and decided whether to handle it by itself or inject it to the associated guest.

Figure 1 describes the interrupt handling flow with baseline device assignment. Each physical interrupt from the guest's assigned device forces at least two exits from guest to host: when the interrupt arrives and when the guest signals completion of the interrupt handling. As we show in Section 5, interrupt-related exits are the foremost contributors to virtualization overhead for I/O intensive workloads.

4. ELI: DESIGN AND IMPLEMENTATION

ELI enables unmodified and untrusted guests to handle interrupts directly and securely. ELI does not require any guest modifications, and thus should work with any operating system. It does not rely on any device-specific features, and thus should work with any assigned device.

4.1. Exitless interrupt delivery

ELI's design was guided by the observation that *nearly* all physical interrupts arriving at a given core are targeted at the guest running on that core. This is due to several reasons. First, in high-performance deployments, guests usually have their own physical CPU cores (or else they would waste too much time context switching); second, high-performance deployments use device assignment with SR-IOV devices; and third, interrupt rates are usually proportional to execution time. The longer each guest runs, the more interrupts it receives from its assigned devices. Following this observation, ELI makes use of available hardware support to deliver *all* physical interrupts on a given core to the guest running on it, since most of them should be handled by that guest anyway, and forces the (unmodified) guest to reflect back to the host all those interrupts which should be handled by the host.

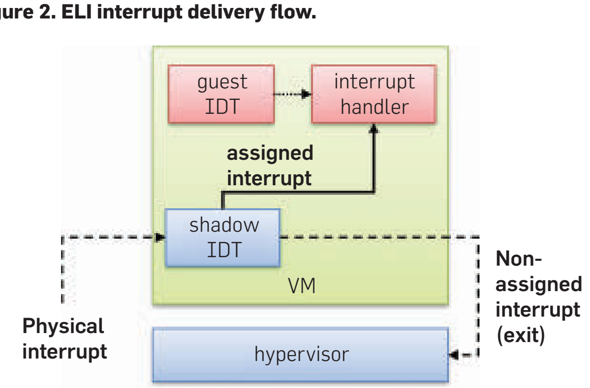
The guest OS continues to prepare and maintain its own IDT. Instead of running the guest with this IDT, ELI runs the guest in guest mode with a different IDT prepared by the host. We call this second guest IDT the *shadow* IDT. Just like shadow page tables can be used to virtualize the guest MMU,[1, 5] IDT shadowing can be used to virtualize interrupt delivery. This mechanism, which is depicted in Figure 2 and described below, requires no guest cooperation.

By shadowing the guest's IDT, the host has explicit control over the interrupt handlers invoked by the CPU on interrupt delivery. The host can configure the shadow IDT to deliver assigned interrupts directly to the guest's interrupt handler or force an exit for nonassigned interrupts. The simplest method to cause an exit is to force the CPU to generate an exception, because exceptions can be selectively trapped by the host and can be easily generated if the host intentionally misconfigures the shadow IDT. For our implementation, we decided to force exits primarily by generating not-present (NP) exceptions. Each IDT entry has a present bit. Before invoking an entry to deliver an interrupt, the processor checks whether that entry is present (has the present bit set). Interrupts delivered to NP entries raise a NP exception. ELI configures the shadow IDT as follows: for exceptions and physical interrupts belonging to devices assigned to the guest, the shadow IDT entries are copied from the guest's original IDT and marked as present. Every other entry in the shadow IDT should be handled by the host and is therefore marked as not present to force a NP exception when the processor tries to invoke the handler. Additionally, the host configures the processor to force an exit from guest mode to host mode whenever a NP exception occurs.

Any physical interrupt reflected to the host appears in the host as a NP exception and must be converted back to the original interrupt vector. The host inspects the cause for this exception. If the exit was actually caused by a physical interrupt, the host raises a software interrupt with the same vector as the physical interrupt, which causes the processor to invoke the appropriate IDT entry. If the exit was not caused by a physical interrupt, then it is a true guest NP exception and should be handled by the guest. In this case, the host injects the exception back into the guest. True NP exceptions are rare in normal execution.

The host sometimes also needs to inject into the guest virtual interrupts raised by devices that are emulated by the host (e.g., the keyboard). These interrupt vectors will have their entries in the shadow IDT marked NP. To deliver such virtual interrupts through the guest IDT handler, ELI enters a special *injection mode* by configuring the

Figure 2. ELI interrupt delivery flow.



processor to cause an exit on any physical interrupt and running the guest with the original guest IDT. ELI then injects the virtual interrupt into the guest for handling, similarly to how it is usually done (Figure 1). After the guest signals completion of the injected virtual interrupt, ELI leaves injection mode by reconfiguring the processor to let the guest handle physical interrupts directly and resuming the guest with the shadow IDT. As we later show in Section 5, the number of injected virtual interrupts is orders of magnitude smaller than the number of physical interrupts generated by the assigned device. Thus, the number of exits due to physical interrupts while running in injection mode is negligible.

Even when all the interrupts require exits, ELI is not slower than baseline device assignment. The number of exits never increases and cost per exit remains the same. Common OS rarely modify the IDT content after system initialization. Entering and leaving injection mode requires only two memory writes, one to change the IDT pointer and the other to change the CPU execution mode.

4.2. Placing the shadow IDT

There are several requirements on where in guest memory to place the shadow IDT. First, it should be hidden from the guest, that is, placed in memory not normally accessed by the guest. Second, it must be placed in a guest physical page that is always mapped in the guest's kernel address space. This is an x86 architectural requirement, since the IDTR expects a virtual address. Third, since the guest is unmodified and untrusted, the host cannot rely on any guest cooperation for placing the shadow IDT. ELI satisfies all three requirements by placing the shadow IDT in an extra page of a device's PCI Base Address Register (BAR).

PCI devices which expose their registers to system software as memory do so through BAR registers. BARs specify the location and sizes of device registers in physical memory. Linux and Windows drivers will map the full size of their devices' PCI BARs into the kernel's address space, but they will only access specific locations in the mapped BAR that are known to correspond to device registers. Placing the shadow IDT in an additional memory page tacked onto the end of a device's BAR causes the guest to (1) map it into its address space, (2) keep it mapped, and (3) not access it during normal operation. All of this happens as part of normal guest operation and does not require any guest awareness or cooperation. To detect runtime changes to the guest IDT, the host also write-protects the shadow IDT page.

4.3. Configuring guest and host vectors

Neither the host nor the guest have absolute control over precisely when an assigned device interrupt fires. Since the host and the guest may run at different times on the core receiving the interrupt, both must be ready to handle the same interrupt. (The host handles the interrupt by injecting it into the guest.) Interrupt vectors also control that interrupt's priority relatively to other interrupts. Therefore, ELI makes sure that for each device interrupt, the respective guest and host interrupt handlers are assigned to the same vector.

4.4. Exitless interrupt completion

Although ELI IDT shadowing delivers hardware interrupts to the guest without host intervention, signaling interrupt completion still forces an exit to host mode. This exit is caused by the guest signaling the completion of an interrupt. As explained in Section 3.2, guests signal completion by writing to the EOI LAPIC register. This register is exposed to the guest either as part of the LAPIC area (older LAPIC interface) or as an x2APIC MSR (the new LAPIC interface). With the old interface, every LAPIC access causes an exit, whereas with the new one, the host can decide on a per-x2APIC-register basis which register accesses cause exits.

Before ELI, the host configured the CPU's MSR bitmap to force an exit when the guest accessed the EOI MSR. ELI exposes the x2APIC EOI register directly to the guest by configuring the MSR bitmap to not cause an exit when the guest writes to the EOI register. Combining this interrupt completion technique with ELI IDT shadowing eliminates the exits on the critical interrupt handling path.

Guests are not aware of the distinction between physical and virtual interrupts. They signal the completion of all interrupts the same way, by writing the EOI register. When the host injects a virtual interrupt, the corresponding completion should go to the host for emulation and not to the physical EOI register. Thus, during injection mode (described in Section 4.1), the host temporarily traps accesses to the EOI register. Once the guest signals the completion of all pending virtual interrupts, the host leaves injection mode.

4.5. Protection

Full details of the considered threat model are available in the full paper. Here we briefly describe possible attacks and the mechanisms ELI employs to prevent them.

A malicious guest may try to steal CPU time by disabling interrupts forever. To prevent such an attack, ELI uses the *preemption timer* feature of x86, which triggers an unconditional exit after a configurable period of time elapses.

A misbehaving guest may refrain from signaling interrupt completion and thereby mask host interrupts. To prevent it, ELI signals interrupt completion for any assigned interrupt still in service after an exit. To maintain correctness, when ELI detects that the guest did not complete any previously delivered interrupts, it falls back to injection mode until the guest signals completions of all in-service interrupts. Since all of the registers that control CPU interruptibility are reloaded upon exit, the guest cannot affect host interruptibility.

A malicious guest can try to block or consume critical physical interrupts, such as a thermal interrupt. To protect against such an attack, ELI uses one of the following mechanisms. If there is a core which does not run any ELI-enabled guests, ELI redirects critical interrupts there. If no such core is available, ELI uses a combination of Non-maskable-Interrupts (NMIs) and IDT limiting.

NMIs trigger unconditional exits; they cannot be blocked by guests. ELI redirects critical interrupts to the core's single NMI handler. All critical interrupts are registered with this handler, and whenever an NMI occurs, the handler calls all registered interrupt vectors to discern which critical interrupt occurred. NMI sharing has a negligible run-time cost (since critical interrupts rarely happen). However, some devices and device drivers may lock up or otherwise misbehave if their interrupt handlers are called when no interrupt was raised.

For critical interrupts whose handlers must only be called when an interrupt actually occurred, ELI uses a complementary coarse grained *IDT limit* mechanism. The IDT limit is specified in the IDTR register, which is protected by ELI and cannot be changed by the guest. IDT limiting reduces the limit of the shadow IDT, causing all interrupts whose vector is above the limit to trigger the usually rare general protection exception (GP). A GP is intercepted and handled by the host similarly to the NP exception. No events take precedence over the IDTR limit check,[11] and all handlers above the limit are therefore guaranteed to trap to the host when called.

5. EVALUATION

We implement ELI within the KVM hypervisor. This section evaluates the performance of our implementation.

5.1. Methodology and experimental setup

We measure and analyze ELI's effect on high-throughput network cards assigned to a guest virtual machine. Network devices are the most common use-case of device assignment, due to their high throughput and because SR-IOV network cards make it easy to assign one physical network card to multiple guests. We use throughput and latency to measure performance, and we contrast the results achieved by virtualized and bare-metal settings to demonstrate that the former can approach the latter. As noted earlier, performance-minded applications would typically dedicate whole cores to guests. We limit our evaluation to this case.

Our test machine is an IBM System x3550 M2 server, equipped with Intel Xeon X5570 CPUs, 24GB of memory, and an Emulex OneConnect 10Gbps NIC. We use another similar remote server (connected directly by 10Gbps fiber) as a workload generator and a target for I/O transactions. Guest mode and bare-metal configurations execute with a single core; 1GB of memory is assigned for each. All setups run Ubuntu 9.10 with Linux 2.6.35.

We run all guests on the KVM hypervisor (which is part of Linux 2.6.35) and QEMU-KVM 0.14.0. To check that ELI functions correctly in other setups, we also deploy it in an environment that uses a different device (BCM5709 1Gbps NIC) and a different OS (Windows 7); we find that ELI indeed operates correctly. We evaluate and compare the performance using baseline device assignment (i.e., unmodified KVM), ELI, and bare-metal system without virtualization.

We configure the hypervisor to back the guest's memory with 2MB *huge pages* and two-dimensional page tables. Huge pages minimize two-dimensional paging overhead and reduce TLB pressure. We note that only the host uses huge pages; in all cases the guest still operates with the default 4KB page size. We quantify the performance without huge pages, finding that they improve performance of both baseline and ELI runs similarly (data not shown).

Recall that ELI makes use of the x2APIC hardware to avoid exits on interrupt completions. x2APIC is available in every Intel x86 CPU since Sandy Bridge microarchitecture. Alas, the hardware we used for evaluation does not support x2APIC. To nevertheless measure the benefits of ELI utilizing x2APIC hardware, we slightly modify our Linux guest to emulate the x2APIC behavior. Specifically, we expose the physical LAPIC and a control flag to the guest, such that the guest may perform an EOI on the virtual LAPIC (forcing an exit) or the physical LAPIC (no exit), according to the flag. We verified that our approach conforms to the published specifications.

5.2. Throughput

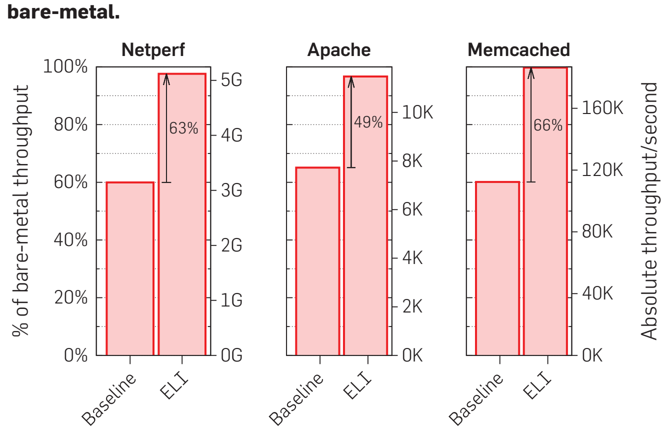
I/O virtualization performance suffers the most with workloads that are I/O intensive and which incur many interrupts. We start our evaluation by measuring three well-known examples of network-intensive workloads, and show that for these benchmarks ELI provides a significant (49–66%) throughput increase over baseline device assignment, and that it nearly (to 0–3%) reaches bare-metal performance.

We consider the following three benchmarks: **Netperf** TCP stream, which opens a single TCP connection to the remote machine, and makes as many rapid `write()` calls of a given size as possible; **Apache** HTTP server, measured using remote *ApacheBench* which repeatedly requests a static page from several concurrent threads; and **Memcached**, a high-performance in-memory key-value storage server, measured using the *Memslap* benchmark which sends a random sequence of `get` (90%) and `set` (10%) requests.

We configure each benchmark with parameters that fully load the tested machine's core (so that throughput can be compared), but do not saturate the tester machine. We configure Netperf to do 256-byte writes, ApacheBench to request 4KB static pages from 4 concurrent threads, and Memslap to make 64 concurrent requests from 4 threads.

Figure 3 illustrates how ELI improves the throughput of these three benchmarks. Each of the benchmarks was run on bare metal and under two virtualized setups: baseline device assignment, and device assignment with ELI.

Figure 3. Performance of I/O intensive workloads relatively to bare-metal.



The figure shows that baseline device assignment performance is still considerably below bare-metal performance: Netperf throughput on a guest is at 60% of bare-metal throughput, Apache is at 65%, and Memcached at 60%. With ELI, Netperf achieves 98% of the bare-metal throughput, Apache 97%, and Memcached 100%. It is evident that using ELI gives a significant throughput increase, 63%, 49%, and 66% for Netperf, Apache, and Memcached, respectively.

5.3. Execution breakdown

Breaking down the execution time to host, guest, and overhead components allows us to better understand how and why ELI improves the guest's performance. Table 1 shows this breakdown for the Apache benchmark (Netperf and Memcached appear in the full paper). We summarize here the results of the three benchmarks.

Guest performance should be better with ELI because the guest gets a larger fraction of the CPU (the host uses less), and/or because the guest runs more efficiently when it gets to run. With baseline device assignment, only 60–69% of the CPU time is spent in the guest; the rest is spent in the host, handling exits. ELI eliminates most of the exits, and thereby reduces both the fraction of time spent in the host (down to 1–2%) and the number of exits (down to 764–1118 per second).

In baseline device assignment, all interrupts arrive at the host and are then injected to the guest. The injection rate is slightly higher than the interrupt rate because the host injects additional virtual interrupts, such as timer interrupts. The number of interrupts "handled in host" is very low (103–207) when ELI is used, because the fraction of the time that the CPU is running the host is much lower.

Baseline device assignment is further slowed down by "IRQ window" exits: on bare metal, when a device interrupt occurs while interrupts are blocked, the interrupt will be delivered by the LAPIC hardware some time later. But when a guest is running, an interrupt always causes an immediate exit. The host wishes to inject this interrupt to the guest (if it is an interrupt from the assigned device), but if the guest has interrupts blocked, it cannot. The x86 architecture solution is to run the guest with an "IRQ window" enabled, requesting an exit as soon as the guest enables interrupts. We see 7801–9069 of these exits every second in the baseline device assignment run. ELI mostly eliminates IRQ window overhead, by eliminating most injections. Consequently, as expected, ELI slashes the number of exits, from 90,506 to 123,134 in the baseline device assignment runs, to just 764–1118.

Table 1. Apache benchmark execution breakdown.

Statistics	Baseline	ELI	Bare-metal
Exits/s	90,506	1118	
Time in guest	67%	98%	
Interrupts/s	36,418	66,546	68,851
Handled in host	36,418	195	
Injections/s	36,671	458	
IRQ windows/s	7801	192	
Requests/s	7729	11,480	11,875
Avg response ms	0.518	0.348	0.337

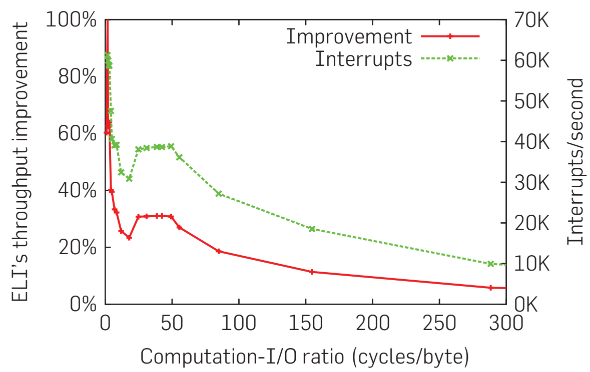
5.4. Impact of interrupt rate

The benchmarks in the previous section demonstrated that ELI significantly improves throughput over baseline device assignment for I/O intensive workloads. But as the workload spends less of its time on I/O and more of its time on computation, it seems likely that ELI's improvement will be less pronounced. Nonetheless, counterintuitively, we shall now show that ELI continues to provide relatively large improvements until we reach some fairly high computation-per-I/O ratio (and some fairly low throughput). To this end, we modify the Netperf benchmark to perform a specified amount of extra computation per byte written to the stream. This resembles many useful server workloads, where the server does some computation before sending its response.

A useful measure of the ratio of computation to I/O is *cycles/byte*, the number of CPU cycles spent to produce one byte of output; this ratio is easily measured as the quotient of CPU frequency (in cycles/second) and workload throughput (in bytes/second). Note that cycles/byte is inversely proportional to throughput. Figure 4 depicts ELI's improvement and the interrupt rate as a function of this ratio. As shown, until after 60 cycles/byte—which corresponds to throughput of only 50Mbps-ELI's improvement stays over 25% and the interrupt rate remains between 30K and 60K. As will be shown below, interrupt rates are kept in this range due to the NIC (which coalesces interrupts) and the Linux driver (which employs NAPI), and they would have been higher if it were not for these mechanisms. Since ELI lowers the overhead of handling interrupts, its benefit is proportional to their rate, *not* to throughput, a fact that explains why the improvement is similar over a range of computation-I/O values.

We now proceed to investigate the dependence of ELI's improvement on the amount of coalescing done by the NIC, which immediately translates to the amount of generated interrupts. Our NIC imposes a configurable cap on coalescing, allowing its users to set a time duration *T*, such that the NIC will not fire more than one interrupt per *T* μs (longer *T* implies less interrupts). We set the NIC's coalescing cap to the following values: 16μs, 24μs, 32μs, …, 96μs. Figure 5 plots the results of the associated experiments (the data along the curve denotes values of *T*). Higher interrupt rates

Figure 4. Throughput improvement and baseline interrupt rate of modified-Netperf workloads with various computation-I/O ratios.



imply higher savings due to ELI. Even with the maximal coalescing ELI still provides a 10% performance improvement over the baseline. ELI achieves at least 99% of bare-metal throughput in all of the experiments described in this section. These results indicate that when ELI is used, coalescing has lesser effect on throughput. The granularity of coalescing can therefore be made finer, so as to refrain from the increased latency that coarse coalescing induces.

5.5. Latency

By removing the exits caused by external interrupts, ELI substantially reduces the time it takes to deliver interrupts to the guest. This period of time is critical for latency-sensitive workloads. We measure ELI's latency improvement using Netperf UDP request-response, which sends a UDP packet and waits for a reply before sending the next. To simulate a busy guest that has work to do alongside a latency-sensitive application, we run a busy-loop within the guest. As the results in Table 2 show, baseline device assignment increases bare metal latency by 8.21 μs and that ELI reduces this gap to only 0.58 μs, which is within 98% of bare-metal latency.

6. AFTERMATH

In our original ASPLOS 2012 paper, we urged hardware vendors to add hardware support that would simplify implementing direct interrupt delivery to guest virtual machines. We made the case that the substantial performance improvement demonstrated by ELI merits the effort to add such support. We are happy to report that, since then, a few positive steps has been taken in this direction.

Figure 5. Throughput improvement and interrupt rate for Netperf benchmark with different interrupt coalescing intervals (shown in labels).

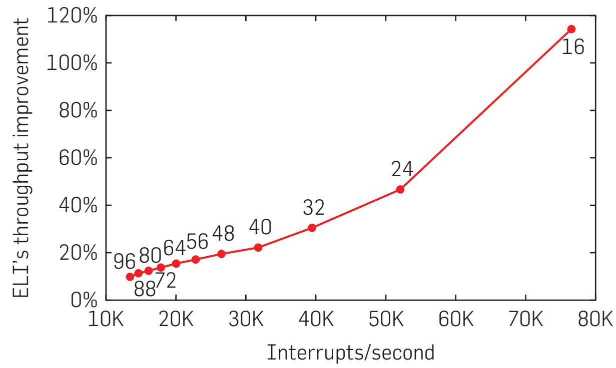


Table 2. Latency measured by Netperf UDP request-response benchmark.

Configuration	Latency (μs)	% Overhead
Baseline	36.14	29
ELI	28.51	2
Bare-metal	27.93	0

To mitigate some of the overheads caused by interrupts delivery, hypervisors can now use the Intel "virtual APIC" (APICv) feature. Assume that a guest currently runs on core C_1. The hypervisor can arrange things such that the relevant (physical) interrupts are triggered on a different core C_2 that runs in host mode. When an interrupt reaches C_2, APICv then allows the hypervisor to "forward" the corresponding (virtual) interrupts to C_1 to the guest *without* inducing an exit. Although such a scheme eliminates unwarranted exits for the guest, it is inferior to ELI due to two reasons. First, it requires dedicating special host cores (like C_2) for redirecting guest interrupts. Second, it increases interrupts delivery latency, as they must first be processed by the hypervisor at C_2 and only then can they be delivered to the guest at C_1.

Both Intel and AMD indicate that they intend to support direct ELI-like delivery in hardware.[12] Some ARM chips already support such delivery.[4] It is still unclear, however, whether this hardware support would live up to its promise. The first generation of Intel implementation, for instance, would deliver each guest interrupt to a certain core. As a result, this implementation may not be usable for multi-core guests whose OS spreads interrupts across the guest cores.

7. CONCLUSION

The key to high virtualization performance is for the CPU to spend most of its time in guest mode, running the guest, and not in the host, handling guest exits. Yet current approaches to x86 virtualization induce multiple exits by requiring host involvement in the critical interrupt handling path. The result is that I/O performance suffers. We propose to eliminate the unwarranted exits by introducing ELI, an approach that lets guests handle interrupts directly and securely. Building on many previous efforts to reduce virtualization overhead, ELI finally makes it possible for untrusted and unmodified virtual machines to reach nearly bare-metal performance, even for the most I/O-intensive workloads. Considering, it seems that the next logical step for chip vendors is extend the posted interrupts architecture so as to support the ELI paradigm in hardware, thereby simplifying its implementation.

Acknowledgments

The research leading to the results presented in this paper is partially supported by the European Community's Seventh Framework Programme ([FP7/2001–2013]) under grant agreements #248615 (IOLanes) and #248647 (ENCORE). C

References

1. Adams, K., Agesen, O. A comparison of software and hardware techniques for x86 virtualization. In *ACM Architectural Support for Programming Languages & Operating Systems (ASPLOS)* (2006).
2. Agesen, O., Mattson, J., Rugina, R., Sheldon, J. Software techniques for avoiding hardware virtualization exits. In *USENIX Annual Technical Conference (ATC)* (2012), 373–385.
3. Ahmad, I., Gulati, A., Mashtizadeh, A. vIC: Interrupt coalescing for virtual machine storage device IO. In *USENIX Annual Technical Conference (ATC)* (2011).
4. ARM Ltd. *Arm Generic Interrupt Controller Architecture Version 2.0.* ARM IHI 0048B, 2011.
5. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A. Xen and the art of virtualization. In *ACM Symposium on Operating Systems Principles (SOSP)* (2003).
6. Ben-Yehuda, M., Day, M.D., Dubitzky, Z., Factor, M., Har'El, N., Gordon, A., Liguori, A., Wasserman, O., Yassour, B.-A. The turtles project: Design and implementation of nested virtualization. In *USENIX Symposium on*

Operating Systems Design & Implementation (OSDI) (2010).
7. Codd, E.F. *Advances in Computers.* Volume 3, New York: Academic Press, 1962, 77–153.
8. Dong, Y., Xu, D., Zhang, Y., Liao, G. Optimizing network I/O virtualization with efficient interrupt coalescing and virtual receive side scaling. In *IEEE International Conference on Cluster Computing (CLUSTER)* (2011).
9. Dong, Y., Yang, X., Li, X., Li, J., Tian, K., Guan, H. High performance network virtualization with SR-IOV. In *IEEE International Symposium on High Performance Computer Architecture (HPCA)* (2010).
10. Dovrolis, C., Thayer, B., Ramanathan, P. HIP: Hybrid interrupt-polling for the network interface. *ACM SIGOPS Operat. Syst. Rev. 35* (2001), 50–60.
11. Intel Corporation. *Intel 64 and IA-32 Architectures Software Developer's Manual*, 2014.
12. Intel Corporation. Intel virtualization technology for directed I/O architecture specification, 2014.
13. Keller, E., Szefer, J., Rexford, J., Lee, R.B. NoHype: Virtualized cloud infrastructure without the virtualization. In *ACM/IEEE International Symposium on Computer Architecture (ISCA)* (2010), ACM.
14. Lange, J.R., Pedretti, K., Dinda, P., Bridges, P.G., Bae, C., Soltero, P., Merritt, A. Minimal-overhead virtualization of a large scale supercomputer. In *ACM/USENIX International Conference on Virtual Execution Environments (VEE)* (2011).
15. Larsen, S., Sarangam, P., Huggahalli, R., Kulkarni, S. Architectural breakdown of end-to-end latency in a TCP/IP network. *Int. J. Parallel Prog. 37*, 6 (2009), 556–571.
16. Liu, J. Evaluating standard-based self-virtualizing devices: A performance study on 10 GbE NICs with SR-IOV support. In *IEEE International Parallel & Distributed Processing Symposium (IPDPS)* (2010).
17. Liu, J., Huang, W., Abali, B., Panda, D.K. High performance VMM-bypass I/O in virtual machines. In *USENIX Annual Technical Conference (ATC)* (2006).
18. Mogul, J.C., Ramakrishnan, K.K. Eliminating receive livelock in an interrupt-driven kernel. *ACM Trans. Comput. Syst. 15* (1997), 217–252.
19. Raj, H., Schwan, K. High performance and scalable I/O virtualization via self-virtualized devices. In *International Symposium on High Performance Distributed Computer (HPDC)* (2007).
20. Ram, K.K., Santos, J.R., Turner, Y., Cox, A.L., Rixner, S. Achieving 10Gbps using safe and transparent network interface virtualization. In *ACM/USENIX International Conference on Virtual Execution Environments (VEE)* (2009).
21. Salah, K. To coalesce or not to coalesce. *Int. J. Electron. Commun. 61*, 4 (2007), 215–225.
22. Salah, K., Qahtan, A. Boosting throughput of Snort NIDS under Linux. In *International Conference on Innovations in Information Technology (IIT)* (2008).
23. Salim, J.H., Olsson, R., Kuznetsov, A. Beyond softnet. In *Annual Linux Showcase & Conference* (2001).
24. Santos, J.R., Turner, Y., Janakiraman, G.J., Pratt, I. Bridging the gap between software and hardware techniques for I/O virtualization. In *USENIX Annual Technical Conference (ATC)* (2008).
25. Willmann, P., Shafer, J., Carr, D., Menon, A., Rixner, S., Cox, A.L., Zwaenepoel, W. Concurrent direct network access for virtual machine monitors. In *IEEE International Symposium on High Performance Computer Architecture (HPCA)* (2007).
26. Zec, M., Mikuc, M., Žagar, M. Estimating the impact of interrupt coalescing delays on steady state TCP throughput. In *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* (2002).

Nadav Amit, Assaf Schuster, and Dan Tsafrir ([namit, assaf, dan]@cs.technion.ac.il), Technion, Haifa, Israel.

Abel Gordon and Muli Ben-Yehuda (abel@stratoscale.com, mulix@mulix.org), Stratoscale, Haifa, Israel.

Authors Gordon, Har'El, Ben-Yehuda, and Landau conducted the research discussed in the paper while employed at IBM.

Nadav Har'El (nadav@harel.org.il), Cloudius Systems, Herzliya Pituach, Israel.

Alex Landau (landau.alex@gmail.com), Facebook, Seattle, WA.

DOI:10.1145/2845614

To view the accompanying paper, visit doi.acm.org/10.1145/2845644

Technical Perspective
Enlisting the Power of the Crowd

By Tova Milo

CROWDSOURCING IS A powerful new project management and procurement strategy that enables the realization of values associated with an "open call" to an unlimited pool of people, typically through Web-based technology. Our focus here is on an important form of crowdsourcing where the crowd's task is to generate or source *data*. Generally speaking, crowd-based data sourcing is invoked to obtain data, to aggregate and/or fuse data, to process data, or, more directly, to develop dedicated applications or solutions over the sourced data.

Wikipedia is probably the earliest and best-known example of crowd-sourced data and an illustration of what can be achieved with a crowd-based data-sourcing model. Other examples include social tagging systems for images that harness millions of Web users to build searchable databases of tagged images, traffic information aggregators like Waze, and hotel and movie ratings like TripAdvisor and IMDb.

Crowd-based data sourcing democratizes the data-collection process, cutting companies' and researchers' reliance on stagnant, overused datasets, and can revolutionize our information world. But in order to work with the crowd, one must overcome several nontrivial challenges, such as dealing with users of different expertise and reliability, and whose time, memory, and attention are limited; handling data that is uncertain, subjective, and contradictory; and so on. Particular crowd platforms typically tackle these challenges in an ad hoc manner, which is application-specific and rarely sharable. These challenges along with the evident potential of crowdsourcing have raised the attention of the scientific community, and called for developing sound foundations and provably efficient approaches to crowdsourcing.

The crowd may be harnessed for various data-related tasks, which generally can be divided into two main types. First, the crowd can help in processing data already collected, by providing their judgments, comparing, cleaning, and matching data items. Second, the crowd could be engaged in harvesting *new* or *missing* data. An important contribution of the following paper is the observation that by using the crowd for the collection of new data, we are departing from the classical *closed word assumption*, which underlies traditional database systems, where the database is considered to be complete at the time a query is posed. That is, it contains all data needed to answer a user query. When the crowd can be enlisted to add new data during query processing, this assumption is violated, calling into question the meaning of even simple queries. In particular, a key question one needs to resolve when collecting data from the crowd to answer a query is: "Has all the data relevant for the query been gathered?" Consider, for example, a query that wishes to collect from the crowd names of companies in California interested in green technology, or of child-friendly chef restaurants in New York. How (and when) can we decide all relevant answers were indeed collected? How can we estimate how many more answers are needed to complete the task?

The authors demonstrate that when dealing with the crowd, the process of sampling significantly differs from what traditional estimators, for related problems, assume.

A natural way to approach the problem is to view the collected crowd answers as a sample taken from some unknown underlying distribution of possible answers, and use some statistical methods for estimating the actual distribution. The authors demonstrate that when dealing with the crowd, the process of sampling significantly differs from what traditional estimators, for related problems, assume. First, crowd members typically provide a list of answers without repetitions, or in other words, sample from an underlying distribution without replacement. Workers also might sample from different underlying distributions (for example, one might provide answers alphabetically, while others provide answers in a different order). Thus, the ordered stream of answers from the crowd may be viewed as a with-replacement sampling among workers who are each sampling a data distribution without replacement. Furthermore, when modeling the crowd, these distributions must account for common crowd behaviors: some workers do much more work than others; not all workers arrive at the same time; workers may have different opinions or bias. Moreover, when data is available on the Web, multiple workers may provide data in the same order (for example, follow, in the example queries here, the same companies or chef-restaurants directory), and so on.

A key contribution of this paper is the development of a simple and elegant formalization of the crowdsourcing process for such queries, along with an effective technique to estimate result set size and query progress in the presence of crowd-specific behavior.

Tova Milo (milo@post.tau.ac.il) is a professor in The Blavatnik School of Computer Science at Tel Aviv University, Israel.

DOI:10.1145/2845644

Answering Enumeration Queries with the Crowd

By Beth Trushkowsky, Tim Kraska, Michael J. Franklin, and Purnamrita Sarkar

Abstract

Hybrid human/computer database systems promise to greatly expand the usefulness of query processing by incorporating the crowd. Such systems raise many implementation questions. Perhaps the most fundamental issue is that the closed-world assumption underlying relational query semantics does not hold in such systems. As a consequence the meaning of even simple queries can be called into question. Furthermore, query progress monitoring becomes difficult due to nonuniformities in the arrival of crowdsourced data and peculiarities of how people work in crowdsourcing systems. To address these issues, we develop statistical tools that enable users and systems developers to reason about query completeness. These tools can also help drive query execution and crowdsourcing strategies. We evaluate our techniques using experiments on a popular crowdsourcing platform.

1. INTRODUCTION

A number of recent projects have demonstrated that leveraging crowdsourcing can greatly extend the usefulness of a query processing system, for example, CrowdDB,[11] Qurk,[19] and sCOOP.[20] In these systems, human workers can be called upon to perform query operations such as subjective comparisons, fuzzy matching for predicates and joins, entity resolution, etc.

Of course, many challenges arise when adding people to query processing due to the peculiarities in latency, cost, quality, and predictability of human workers. Researchers have addressed many of these concerns, but we observe that adding the crowd to a database query processor raises issues of an even more fundamental semantic nature. Relational query languages are based on the *closed-world assumption*, in which the database is considered to be complete at the time a query is posed. That is, it contains all data needed to answer the query. When the crowd can be enlisted to add new data during query processing, this assumption is violated, calling into question the meaning of even simple queries.

1.1. Can you get it all? Enumeration queries

In this paper, we consider one of the most basic operations in a Relational Database Management System (RDBMS), namely, scanning a single database table with filtering constraints, or *predicates*; this query enumerates a particular set of items of interest to the user. Consider, for example, a SQL query to list all house plants that can tolerate low-light environments: `SELECT DISTINCT name FROM Plants WHERE light-needs = 'low'`. With a traditional RDBMS and a given database state there is a single correct answer for this query, and it can be obtained by scanning the table, filtering the records, and returning all matching records. This approach works even for relations that are in reality unbounded, because the closed-world assumption dictates that any records not present in the database at query execution time do not exist.

In contrast, in a crowdsourced system such as CrowdDB, once the records in the stored table are exhausted, jobs can be sent to the crowd asking for additional records. The question then becomes: when is the query result set complete? Crowdsourced queries can be inherently fuzzy or have unbounded result sets, with tuples scattered over the web or only in human minds. For example, consider a query for a list of graduating Ph.D. students currently on the job market, or companies in California interested in green technology. These types of queries are the main use cases for crowd-enabled database systems, as each is labor-intensive for the user issuing the query to perform, but not executed frequently enough to justify the development, tuning, and use of a complex machine learning solution.

In this paper, we address the question of "How should users think about enumeration queries in the open world of a crowdsourced database system?" We develop statistical tools that enable users to reason about time/cost and completeness trade-offs, and that can be used to drive query execution and crowdsourcing strategies.

1.2. Counting species

The key idea of our technique is to use the arrival rate of new answers from the crowd to reason about the completeness of the query. Consider the execution of a "`SELECT DISTINCT *`" query in a crowdsourced database system where workers are asked to provide individual records of the table. For example, one could query for the names of the 50 US states using a microtask crowdsourcing platform such as Amazon's Mechanical Turk (AMT) by generating HITs (i.e., Human Intelligence Tasks) that would have workers provide the name of one or more states. As workers return results, the system collects the answers, keeping a list of the unique answers.

Figure 1 shows the results of running that query, with the number of unique answers received shown on the vertical axis, and the total number of answers received on the *x*-axis.

The original version of this paper is entitled "Crowdsourced Enumeration Queries" and was published in ICDE, 2013, IEEE.[22]

As would be expected, initially there is a high rate of arrival for previously unseen answers, but as the query progresses (and more answers have been seen) the arrival rate of new answers begins to taper off, until the full population (i.e., the 50 states, in this case) has been identified.

This behavior is well-known in fields such as biostatistics, where this type of figure is known as the *Species Accumulation Curve* (SAC).[9] Imagine you are trying to count the number of unique species of animals on an island by putting out traps overnight, identifying the unique species found in the traps the next morning, releasing the animals and repeating this daily. By observing the rate at which new species are identified, you can infer the true, albeit hidden, number of species. We can apply this reasoning to enumeration queries in a crowdsourced query processor.

1.3. Overview of the paper

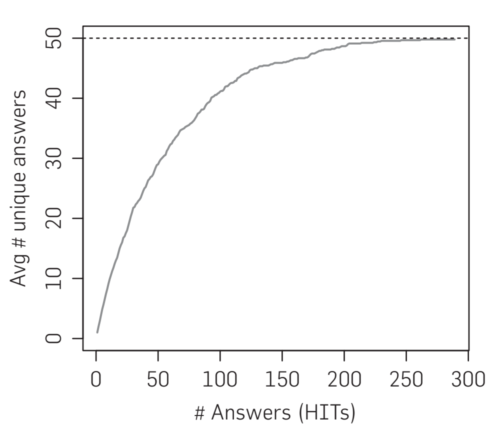
In this paper, we apply species estimation techniques from the statistics and biology literature to understand and manage the execution of enumeration queries in crowdsourced database systems. We find that while the classical theory provides the key to understanding the meaning of such queries, there are certain peculiarities in the behavior of microtask crowdsourcing workers that require us to develop new methods to improve the accuracy of result set size estimation in this environment.

We also describe methods to leverage these techniques to help users make intelligent trade-offs between time/cost and completeness. The usefulness of these techniques extends beyond crowdsourced databases, for example, to estimate the completeness of deep-web queries.

To summarize, we make the following contributions:

- We formalize the process of crowdsourced enumeration and describe how it violates fundamental statistical assumptions of existing species estimation techniques.
- We develop a technique to estimate result set size, or *cardinality*, and query progress in the presence of crowd-specific behaviors.

Figure 1. U.S. States: unique versus total number of answers.



- We devise a technique to determine if data scraping could be applied, as well as describe a pay-as-you-go approach to allow informed decisions about the cost/completeness tradeoff.
- We examine the effectiveness of our techniques via experiments using AMT.

2. BACKGROUND: CROWDDB

CrowdDB is a hybrid human–machine database system that uses human input to process queries. CrowdDB supports different crowdsourcing platforms; we focus on AMT in this paper, the leading platform for the so-called microtasks. Microtasks usually do not require any special training and do not take more than a few minutes to complete. AMT provides a marketplace for requesters to post microtsks and workers to search for and work on these tasks for a small reward, typically a few cents.

CrowdDB incorporates traditional query compilation, optimization, and execution components, which are extended to cope with human-generated input. The system is extended with crowd-specific components, such as a user interface (UI) manager and quality control/progress monitor. Users issue queries using CrowdSQL, an extension of standard SQL. The UI Manager is responsible for generating the task interface for the microtasks that crowd workers interact with. CrowdDB automatically generates UIs as HTML forms based on `CROWD` annotations. Note that in contrast to traditional database system front end UIs that enable end users to construct and submit query requests, these UIs are part of the database system back end and are presented to crowd workers who are assisting with the actual query execution. Figure 2 shows an example HTML UI that would be presented to a worker for the following crowd table definition:

```
CREATE CROWD TABLE ice_cream_flavor {
     name VARCHAR PRIMARY KEY
}
```

During query processing, the system automatically posts one or more HITs using the AMT web service API and collects the answers as they arrive. After receiving the answers, CrowdDB performs simple quality control using a majority vote across crowd workers before it passes the answers to the query execution engine. Finally, the system continuously updates the query result and estimates the quality of the current result based on the new answers. The user may thus stop the query as soon as the quality is sufficient or intervene if a problem is detected. More details about the CrowdDB components and query execution are given in Franklin et al.[11] This

Figure 2. Ice cream flavors task UI on AMT.

paper focuses on the progress component that allows the user to continuously reason about query completeness and cost.

3. PROGRESS ESTIMATION

To evaluate progress as answers are arriving, the system needs an estimate of the result set's size, also called *cardinality*, in order to calculate the percentage complete. Species estimation algorithms tackle a similar goal: an estimate of the number of distinct species is determined using observations of species in the locale of interest. In this section, we describe observations of how the crowd answers enumeration queries and the challenges of estimation for crowdsourced queries. We present a model for crowdsourced enumerations and list the requirements for human-tolerant estimators.

3.1. The problem with existing estimators

Various techniques have been devised to estimate the number of species[3, 6] and to estimate the number of distinct values in a database table.[14] They all operate similarly: a sample is drawn at random from a population (e.g., the entire table) and based on the frequency of observed items (distinct values), the true and unknown number of distinct values is estimated. The techniques differ most notably in their assumptions, in particular distinct value estimation techniques assume that the population (i.e., table) size is known. Unfortunately, knowledge of the population size is only possible in the closed world; in systems that allow crowdsourced enumerations, records can be acquired on-demand, thus the table size is potentially infinite. We focus on estimators suitable for the open world as they allow for an infinite population.

To gain an understanding of the crowd's ability to answer enumeration queries and the impact of crowd behaviors on existing estimation techniques, we crowdsourced the elements of sets for which the true cardinality is known. We use the open-world-safe estimator "Chao92"[7] as it is widely used in the species estimation literature.[4] Figure 3 shows the observed Chao92 estimate ("actual") evaluated as answers arrive in one AMT experiment in which we crowdsourced the names of the 192 United Nations (UN) member countries and compares it to the expected behavior using simulation with the empirical data distribution (DD) derived from all runs of the UN experiment. We focus on a single representative experiment rather than an average over multiple runs to investigate the behavior a user would observe; averaging can also disguise the effects we describe next.

Note in Figure 3 that the value of the estimate begins approaching the true value of 192, however it then significantly *overestimates* the true value for most of the remaining time of the experiment. This is surprising as our simulation shows that the estimate should become more accurate and stable as it receives more data ("expected" in Figure 3). As it turns out, the way in which crowd workers each provide their answers deeply impacts the behavior of an estimation algorithm. For example, some workers enumerated the UN countries by traversing an alphabetical list. However, some workers began their answer sequence with a few countries they knew of (e.g., United States, India, Pakistan, China, etc.), or to provide a completely non-alphabetical sequence. In general, people may use different internal biases or techniques for finding items in the set (we discuss full list traversals in Trushkowsky et al.[22]). Furthermore, individual workers complete different amounts of work and arrive/depart from the experiment at different points in time.

3.2. A model for human enumerations

Species estimation algorithms assume a with-replacement sample from some unknown distribution describing item likelihoods (visualized in Figure 4a). The order in which elements of the sample arrive is irrelevant in this context.

After analyzing crowdsourced enumerations (e.g., in the previously mentioned UN experiment), we found that this assumption does not hold. In contrast to with-replacement samples, individual workers typically provide answers from an underlying distribution *without* replacement. Furthermore, workers might sample from different underlying distributions (e.g., one might provide answers alphabetically, while others provide answers in a different order).

This process of sampling significantly differs from what traditional estimators assume, and it can be represented as

Figure 3. Chao92 cardinality estimate evaluated for increasing number of samples, or answers from the crowd, for the United Nations use case.

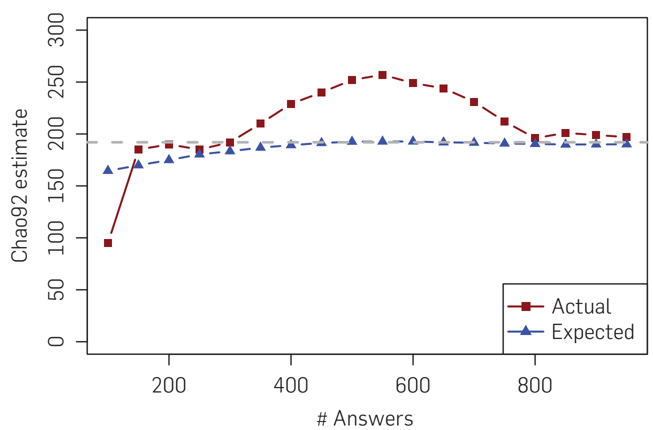
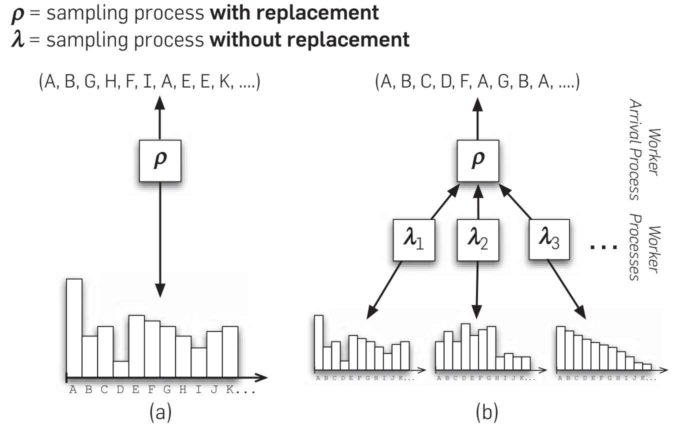


Figure 4. Comparison of the sampling processes, (a) assumed by traditional algorithms and (b) observed in crowd-based enumerations.



a two-layer sampling process as shown in Figure 4b. The bottom layer consists of many sampling processes, each corresponding to an individual worker, that sample from some DD *without replacement*. The top layer processes samples *with* replacement from the set of the bottom-layer processes (i.e., workers). Thus, the ordered stream of answers from the crowd represents a with-replacement sampling among workers who are each sampling a DD without replacement.

Next we discuss the impact of the parameterization of the two-layer sampling process (e.g., the number of worker processes, different underlying distributions, etc.) on estimation.

3.3. Sampling without replacement and worker skew

Individuals are sampling without replacement from some underlying distribution that describes the likelihood of selecting each item. This behavior is beneficial with respect to the goal of acquiring all the items in the set, as low-probability items become more likely after the high-probability items have already been provided by that worker (we do not pay for duplicated work from a single worker). A negative side effect, however, is that the estimator receives less information about the relative frequency of items, and thus the skew, of the underlying DD. This can cause the estimator to over-predict due to the more rapid arrival of unseen items than would occur in a with-replacement sample.

Over-prediction also results when some workers complete many more HITs than others; workers who do significantly more work have been called "streakers."[15] In the two-layer sampling process, worker skew (WS) dictates which worker supplies the next answer—skew in the ρ process; streakers are chosen with higher frequency. High WS can cause the arrival rate of unique answers to be more rapid than that caused by sampling without replacement alone, causing the estimator to over-predict.

In an extreme scenario in which one worker provides all answers, the two-layer process reduces to one process sampling from one underlying distribution without replacement. In this case, completeness estimation becomes impossible because no inference can be made regarding the underlying distribution. Another extreme is if an infinite number of workers each provide one answer using the same underlying distribution, the resulting sample would correspond to the sampling with replacement scenario (Figure 4a). The latter is the reason why it is still possible to make estimations despite human-generated enumerations.

To illustrate the impact on the Chao92 estimator of the number of workers participating in a crowd-based enumeration, we simulated different numbers of workers sampling from a uniform distribution over 200 items, averaging over 100 runs. Figure 5a depicts the values of the Chao92 estimator calculated for increasing numbers of samples for three scenarios: a with-replacement sample (equivalent to an infinite number of workers), and three or five workers each sampling without replacement from the item distribution. As expected, the with-replacement sample overestimates slightly because of the uniform DD, but quickly approaches the true value of 200. The without-replacement samples, having fewer workers, overestimate even more and remain in that state for longer.

3.4. Different and skewed data distributions

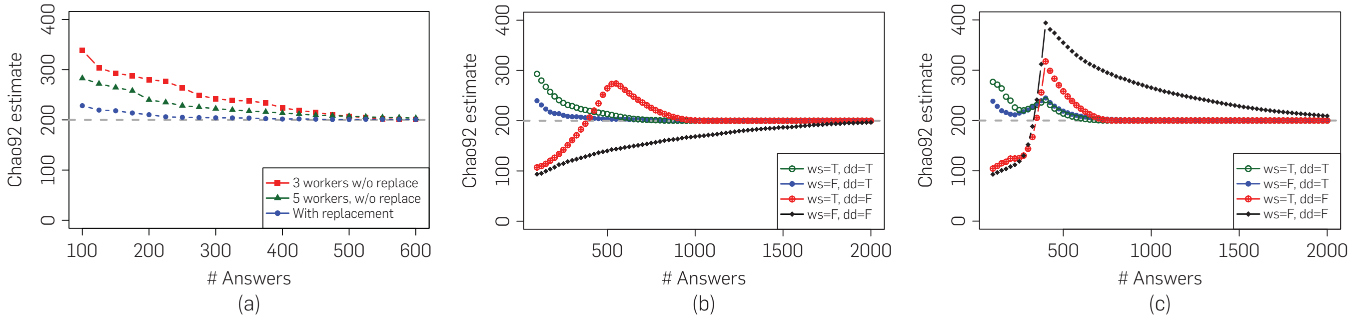
Individual workers may draw their answers from different DDs: the most likely item for one worker could be the least likely for another. These differences could arise from varying cultural or regional biases, or alternate techniques for finding data on the web. A mixture of multiple distributions over the same data yields a combined distribution that is "flatter" than its constituent parts, becoming less skewed. In contrast, when the underlying DD is heavily skewed and shared amongst workers, the estimator will underestimate because there will not be a sufficient number of items representing the tail of the distribution.

Figure 5b shows the impact of different DDs combined with different WS on the Chao92 estimate. It shows four combinations: the absence/presence of WS and shared/different DDs for workers. For all cases, we use a power law DD in which the most likely item has probability p, the second most likely has probability $p(1-p)$, and the kth most likely item has probability $p(1-p)^{k-1}$, etc.; we set $p = 0.03$. To simulate different DDs, we randomly permute the original distribution for each worker.

The simulation shows that the worst scenario is characterized by a high WS and a single shared DD (WS = T and DD = F). With a shared skewed distribution, Chao92 will start out underestimating because all workers are answering with the same high-probability items. With high WS, the streaker(s) provide(s) many unique answers quickly causing many more unique items than encountered with sampling with replacement.

On the other hand, the best scenario is when there is no WS but there are different DD (WS = F and DD = T). By using

Figure 5. Cardinality estimation simulations illustrating the impact of worker behaviors, (a) With vs. without replacement, (b) Forms of skew, and (c) Impact of streaker.



different DDs without overemphasizing a few workers, the overall sample looks more uniform, similar to Figure 5a with replacement, due to the flattening effect of DD on skewed data.

3.5. Worker arrival

Finally, the estimate can be impacted by the arrival and departure of workers during the experiment. Not all workers provide answers during the lifetime of a query. However, the estimator can be strongly impacted when streakers arrive who then suddenly dominate the total number of answers.

Figure 5c demonstrates the impact a single worker can have. It uses the same simulation setup as in Figure 5b, but also includes an additional single streaker starting at 200 HITs who continuously provides all 200 answers before anyone else has a chance to submit another answer. As the figure shows, it causes Chao92 to over-predict in all four cases. However, if workers use different DDs the impact is not as severe. Again, this happens because DD makes the sample appear more uniformly distributed.

3.6. Discussion

Some of the behaviors that workers exhibit as they respond to enumeration queries are inherent in a marketplace such as AMT, such as how many tasks an individual worker chooses to complete. The order in which each worker provides his/her answers and how many he/she gives can depend on individual biases and preferences. The four elements of crowd behavior we outlined above (without-replacement sampling, WS, different distributions, and worker arrival) can each cause Chao92 to perform poorly. The most volatile of these behaviors is WS, particularly when the DD itself is skewed; a single overzealous worker could cause massive fluctuations in the estimate. Overestimation in particular is problematic because it may lead to unnecessary crowdsourcing costs in an attempt to enumerate more items of the set that do not actually exist. However, we do not want to prohibit individual workers, especially highly productive ones, from submitting responses, as doing so would slow or limit the progress of the query. Thus we want to make Chao92 more tolerant to the impact of such a streaker while still allowing the streaker to submit answers; we discuss our technique for a streaker-tolerant cardinality estimator next.

4. STREAKER-TOLERANT ESTIMATOR

Our goal is to provide a progress estimate for an open-world query based on the answers that have been gathered so far. In this section, we extend the Chao92 estimator to be more robust to the impact of individual workers, focusing our effort on reducing the impact of streakers and worker arrival. We first introduce the basic estimator model and Chao92 more formally before we present our extension to handle streakers. We evaluate our technique by first proposing a new metric that incorporates notions of estimate stability and fast convergence to the true cardinality, then applying this metric to measure our technique's effectiveness.

4.1. Basic estimator model and *f*-statistic

Receiving answers from workers is analogous to drawing samples from some underlying distribution of unknown size N; each answer corresponds to one sample from the item distribution. We can rephrase the problem as a species estimation problem as follows: The set of HITs received from AMT is a sample of size n drawn from a population in which elements can be from N different classes, numbered $1 - N$ (N, unknown, is what we seek); c is the number of unique classes (species) seen in the sample. Let n_i be the number of elements in the sample that belong to class i, with $1 \leq i \leq N$. Of course some $n_i = 0$ because they have not been observed in the sample. Let p_i be the probability that an element from class i is selected by a worker, $\sum_{i=1}^{N} p_i = 1$.

Burnham and Overton[5] show that the aggregated "frequency of frequencies"-statistic (hereon *f*-statistic) is sufficient for estimating the number of unobserved species for nonparametric algorithms. The *f*-statistic captures the relative frequency of observed classes in the sample. Let f_j be the number of classes that have exactly j members in the sample. The goal is to estimate the cardinality by predicting f_0, the number of unseen classes.

4.2. The Chao92 estimator

The Chao92[7] estimator uses *sample coverage* to predict N. The sample coverage C is the sum of the probabilities p_i of the observed classes. Since the underlying distribution $p_1 \cdots p_N$ is unknown, the Good-Turing estimator[12] using the *f*-statistic is used:

$$\hat{C} = 1 - \frac{f_1}{n} \tag{1}$$

The Chao92 estimator attempts to explicitly characterize and incorporate the skew of the underlying distribution using a *coefficient of variance* (CV) γ, a metric that is used to describe the variance in a probability distribution[7]; we can use the CV to compare the skew of different class distributions. The CV is defined as the standard deviation divided by the mean. Given the p_i's ($p_1 \cdots p_N$) that describe the probability of the ith class being selected, with mean $\bar{p} = \sum_i p_i / N = 1/N$, the CV is expressed as $\gamma = [\sum_i (p_i - \bar{p})^2 / N]^{1/2} / \bar{p}$.[7] Higher CV means higher variance in the p_i's; a CV of 0 means each item is equally likely.

The true CV cannot be calculated without knowledge of the p_i's, so Chao92 estimates $\hat{\gamma}$ based on the *f*-statistic:

$$\hat{\gamma}^2 = \max\left\{ \frac{\frac{c}{\hat{C}} \sum_i i(i-1) f_i}{n(n-1)} - 1, 0 \right\} \tag{2}$$

The final estimator is then defined as:

$$\hat{N}_{chao92} = \frac{c}{\hat{C}} + \frac{n(1-\hat{C})}{\hat{C}} \hat{\gamma}^2 \tag{3}$$

4.3. Estimator for crowdsourced enumeration

The Chao92 estimator is heavily influenced by the presence of rare items in the sample; the coverage estimate $\hat{C}$ is based entirely on the percentage of singleton answers (f_1s). Recall from Section 3 the discussion of different crowd behaviors—many of these result in rapid arrival of previously unseen answers. When these new f_1 items appear "too quickly," the estimator interprets this as a sign the complete set size is larger than it truly is. We develop an estimator based on Chao92 that ameliorates some of the overestimation issues caused by an overabundance of f_1 answers.

Most of the dramatic overestimation occurs due to streakers, that is, significant skew in the number of answers provided by each worker. Notably, problems occur when one or a few workers contribute substantially more answers than others, possibly also drawing answers from a different DD. Since other workers are not given the opportunity to provide answers that would subsequently increase the f_2s, f_3s, etc. in the sample, Chao92 predicts a total set cardinality that is too large. Thus our estimator is designed to identify any worker(s) who are outliers with respect to their contribution of unique answers in the sample (their f_1 answers).

The idea behind making the Chao92 estimator more resilient against streakers is to alter the *f*-statistic. The first step is to identify those workers who are "f_1 outliers." We define outlier traditionally, namely, two standard deviations outside the mean of all workers *W*. To avoid false negatives due to a true outlier's influence on the mean and standard deviation, both statistics are calculated without including the potential outlier's f_1 count. The f_1 count of worker *i* is compared to the mean $\bar{x}_i$ and the sample standard deviation $\hat{\sigma}_i$:

$$\bar{x}_i = \sum_{\forall j, j \neq i} \frac{f_1(j)}{W-1}, \quad \hat{\sigma}_i = \sqrt{\sum_{\forall j, j \neq i} \frac{(f_1(j) - \bar{x}_i)^2}{W-2}} \tag{4}$$

We create $\tilde{f}_1$ from the original f_1 by reducing each worker *i*'s f_1-contribution to fall within $2\hat{\sigma}_i + \bar{x}_i$:

$$\tilde{f}_1 = \sum_i min(f_1(i), 2\hat{\sigma}_i + \bar{x}_i) \tag{5}$$

The final estimator is similar to Equation (3) except that it uses the $\tilde{f}_1$ statistic. For example, with a CV $\hat{\gamma}^2 = 0$, it would simplify to:

$$\hat{N}_{crowd} = \frac{cn}{n - \sum_i min(f_1(i), 2\hat{\sigma}_i + \bar{x}_i)} \tag{6}$$

Although a small adjustment, $\hat{N}_{crowd}$ is more robust against the impact of streakers than the original Chao92, as we show next.

4.4. Experimental results

We ran more than 30,000 HITs on AMT for enumeration tasks. The CROWD tables we experimented with include small and large well-defined sets like NBA teams, US states, UN member countries, as well as sets that can truly leverage human perception and experience such as indoor plants with low-light needs, restaurants in San Francisco serving scallops, slim-fit tuxedos, and ice cream flavors. Workers were paid \$0.01–\$0.05 to provide one item in the result set using a UI similar to that in Figure 2; they were allowed to complete multiple tasks if they wanted to submit more than one answer. In the remainder of this paper we focus on a subset of the experiments, two with known cardinality and fixed membership, US states (nine experiment runs) and UN member countries (five runs), as well as more open-ended queries (one run each).

Error metric. Due to a lack of a good metric to evaluate estimators with respect to stability and convergence rate, we developed an error metric Φ that captures bias (absolute distance from the true value), as well as the estimator's time to convergence and stability. The idea is to weight the magnitude of the estimator's bias more as the size of the sample increases. Let *N* denote the known true value, and $\hat{N}_i$ denote the estimate after *i* samples. After *n* samples, Φ is defined:

$$\Phi = \frac{\sum_{i=1}^{n} \left|\hat{N}_i - N\right| i}{\sum i} = \frac{2\sum_{i=1}^{n} \left|\hat{N}_i - N\right| i}{n(n+1)} \tag{7}$$

Lower Φ means a smaller averaged bias and thus a better estimate. The weighting gives a harsher penalty for incorrectness later on than in the beginning, in addition to penalizing an estimator that takes longer to reach the true value, addressing the convergence rate criteria. The metric also rewards estimators for staying near the true value.

Results: UN and states. We first illustrate how $\hat{N}_{crowd}$ behaves for a representative set of UN member countries and US states experiments; we elide the full set for space reasons. As discussed in Section 3, we do not average the results across experiment runs because each run may have different data and WSs; averaging could disguise the impacts of worker behaviors on the estimators.

Figure 6a–g show cardinality estimates as well as the Φ metric for these experiments. Each graph shows the Chao92 algorithm estimates (labeled "original") and the value of the error metric calculated for those estimates (Φ_{orig}), as well as the estimates and error (Φ_{new}) for the streaker-tolerant estimator (labeled "crowd estimator"). We observed that our estimate has an improvement over Chao92 for most UN experiments.

In the experiment run labeled UN 1, our estimates avoids the overestimation of Chao92 that occurred during the middle of the experiment. In the UN 2 experiment, one streaker dominated the total answer set at the beginning—a substantial outlier. Once his/her contribution was reduced dramatically, the remaining workers' answers had significant overlap because most were enumerating the list of nations alphabetically, resulting in a low cardinality because of the heavily skewed DD this scenario creates. Recall from the previous section that the expected behavior of the estimator in this case is to *underpredict*. In contrast, the third UN experiment run had several streakers at the beginning who each had very different DDs (i.e., enumerating the list of nations from different alphabetical start points). While the heuristic helped level the f_1 contribution from these workers, overestimation still occurs due to the combined number of singleton answers from them. In a few cases, our estimator performs worse than Chao92, for example, experimental run UN 4. Underestimation is expected when workers share a highly skewed distribution; a streaker causing an estimate to be higher than it should incidentally yields a value closer to the true value.

The effect of our estimate compared to Chao92 is less significant in the States experiments, which had less WS. Figure 6e and f show two US states experiments that have moderate streaker issues, helped by $\hat{N}_{crowd}$. In a third state experiment (Figure 6g), our estimator reduces the streakers' impact but takes longer to converge for similar reasons as in run 4 of the UN experiment.

Results: Open-ended use cases. The UN countries and U.S. states use cases are both sets for which the true cardinality, as well as the sets' contents, is known; use cases with known cardinality allow us to evaluate the accuracy of the estimation algorithms. Here we look at use cases that are "open-ended," that is, the set contents and cardinality are not known. For these results, we can only compare estimation algorithms. The open-ended use cases demonstrate several of the worker behaviors that we observed in the UN experiments; in particular, the presence streakers. Figure 7a–d show Chao92 and our $\hat{N}_{crowd}$ for the plants, restaurants, tuxedos, and ice cream flavors experiments.

In all cases, our estimator successfully reduces the impact these streakers have on the prediction of complete set cardinality. Note that we cannot evaluate the error Φ for these experiments because the true cardinality is unknown. During the plant experiment (Figure 7a), one worker from the beginning consistently contributed more unique answers than the other workers, for example, a less well-known plant called "rabbit's foot"; many workers stuck to well-known answers (e.g., snake plant, peace lily). In contrast, in the restaurant experiment (Figure 7b) a streaker contributed many f_1 answers at the beginning, but other workers eventually provided many of those same answers. The tuxedos experiment (Figure 7c) shows the impact of a streaker arriving later in the experiment, causing a sharp increase in the Chao92 estimate which is helped by $\hat{N}_{crowd}$.

4.5. Discussion

We showed that our estimator successfully provides more accurate prediction for crowd-based enumerations in the presence of overzealous workers. Our technique specifically tackles cardinality overestimation, which can cause the user issuing the query to think the result set is less complete than it really is. However, any estimator can only cope with a certain range of worker behavior. The extreme cases in which only one worker provides answers, or if workers share a heavily skewed distribution, will prove difficult for an estimator. Most of the experiments we ran did not have these issues, and the heuristic is able to ameliorate the impact of worker behavior on estimation.

5. LIST WALKING

When workers share the same or multiple heavily skewed DD, the estimator may under-predict the total set size. Such a heavily skewed distribution can occur if workers are traversing the same list for answers; we refer to this effect as *list walking*. Detecting list walking makes it possible to change the crowdsourcing strategy to save money. In cases where one or two lists containing the full set exists, such as the UN countries, this switch could be helpful for getting them all. However, switching strategies for which no single list exists would not make sense.

The goal of detecting list walking is to differentiate between samples drawn from a skewed item distribution and the existence of a list, which leads to a deterministic answer

Figure 6. Estimator results on representative UN country and U.S. states experiments, (a) UN 1, (b) UN 2, (c) UN 3, (d) UN 4, (e) States 1, (f) States 2, and (g) States 3.

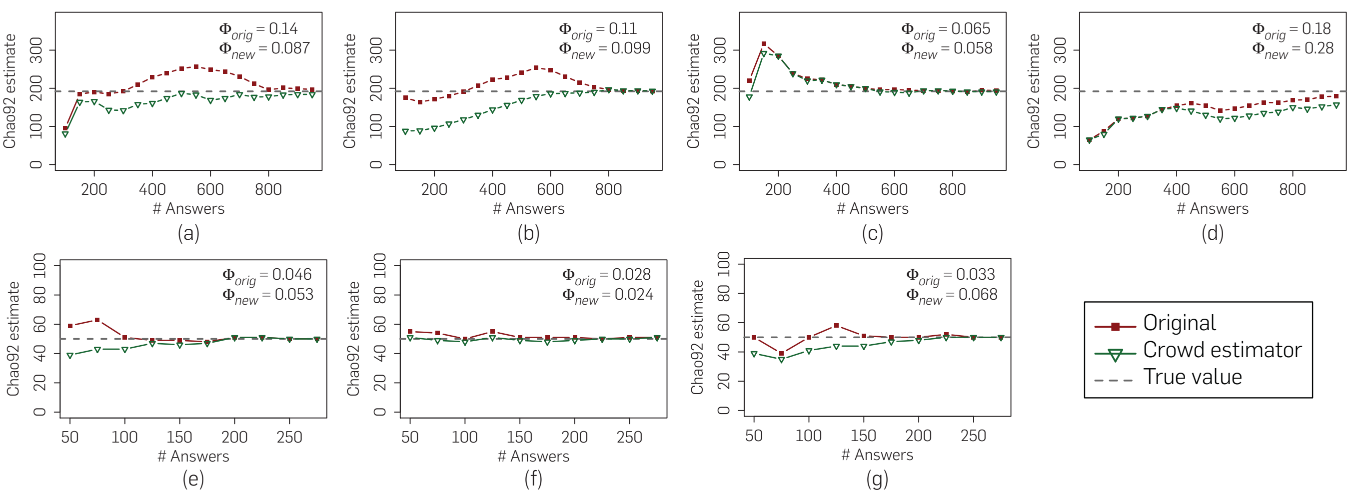
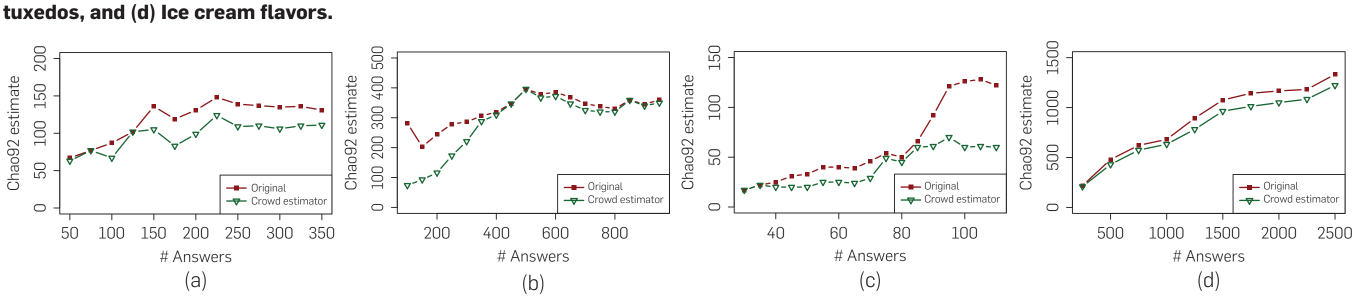


Figure 7. Estimator results for the open-ended use cases, (a) Plants for indirect sunlight, (b) Restaurants serving scallops, (c) Slim-fit tuxedos, and (d) Ice cream flavors.



sequence. In this section, we develop a heuristic to determine the probability that a given number of workers w would respond with s answers in the exact same order. If this probability is below a threshold (we use 0.01), we conclude that list walking is likely to be present.

5.1. Preliminary setup: Binomial distribution

Let W be the total number of workers who have provided answer sequences of length s or more. Among these, let w be the number of workers who have the same sequence of answers with length s starting at the same offset o in common. We refer to this sequence as the *target sequence* α of length s, which itself is composed of the individual answers α_i at every position i starting with offset o ($\alpha = (\alpha_{o+1}, \ldots, \alpha_{o+s})$). If p_α is the probability of observing that sequence from some worker, we are interested in the probability that w out of W total workers would have that sequence. Furthermore, in our scenario, we do not necessarily care about the probability of exactly w workers providing the same sequence, but rather the probability of *w or more* workers with the same answer sequence This probability can be expressed using the binomial distribution: W corresponds to the number of trials and w represents the number of successes:

$$Pr \geq (w; W, p_\alpha) = 1 - \sum_{i=0}^{w-1} \binom{W}{i} p_\alpha^i (1 - p_\alpha)^{W-i} \quad (8)$$

The probability in Equation (8) determines if the target sequence shared among w out of W workers is likely caused by list walking. We now discuss p_α, the probability of observing a particular target sequence α of length s.

5.2. The probability of a target sequence

Not all workers use the same list or use the same order to walk through the list, so we want p_α to reflect the observed answer sequences from workers. We do this by estimating the probability $p_\alpha(i)$ of encountering answer α_i in the ith position of the target sequence by the fraction of times this answer appears in the ith position among all W answers. Let $r(i)$ be the number of times answer α_i appears in the ith position among all the sequences W being compared, $p_\alpha(i)$ is defined as r_i/W. For example, if the target sequence α starting at offset o is "A, B, C" and the first answers for four workers are "A," "A," "A," and "B," respectively, r_{o+1}/W would be 3/4. Now the probability of seeing α is a product of the probabilities of observing α_{o+1}, then α_{o+2}, etc.

$$p_\alpha = \prod_{i=0}^{o+s} \frac{r_i}{W} \quad (9)$$

Relying solely on the data in this manner could lead to false negatives in the extreme case where $w = W$, that is, where all workers use the same target sequence. Note that in this case p_α attains the maximum possible value of 1. We need to incorporate *both* the true data via r_i/W as well as a pessimistic belief of the underlying skew. As a pessimistic prior, we choose the highly skewed Gray's self-similar distribution,[13] often used for the 80/20 rule. Only if we find a sequence which can not be explained (with more than 1% chance) with the 80/20 distribution, we believe we have encountered list walking. Assuming a high skew distribution is conservative because it is more likely that workers will answer in the same order if they were truly sampling than with, say, a uniform distribution. We assume that the target sequence follows the self-similar distribution exactly by always choosing the most likely sequence. In this case α is simply a concatenation of the most likely answer, followed by the second most likely answer, and so on. The likelihood of selecting this sequence under our prior belief is $(1 - h)^s$ and the likelihood that a set of w workers select this sequence is:

$$(1 - h)^{sw} \quad (10)$$

To combine the distribution derived from data and our prior belief in the maximum skew, we use a smoothing factor β to shift the emphasis from the data to the distribution; higher values of β put more emphasis on the data. Using β to combine Equation (9) with Equation (10), we yield the probability of having the target sequence α (of length s) in common:

$$p_\alpha = \prod_{i=1}^{s} \left(\beta \frac{r_i}{W} + (1-\beta)(1-h) \right) \quad (11)$$

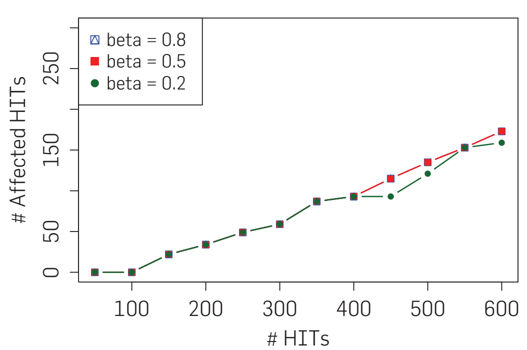
5.3. Experimental results

We apply our heuristic to the AMT experiments, looking at target sequences α of at least length $s = 5$. That is, for a sequence of answers of at least size s that have more than one worker in common, we compute the probability of that sequence using Equation (8). A sequence is considered a list if the probability falls below the threshold 0.01. We check for list use over time (number of HITs) and quantify how many of the observed HITs were part of a list; this gives a sense of the impact of list use in the experiment. Due to space, we describe results for one run of the UN experiment.

Figure 8 shows the number of affected HITs in one of the UN experiments. The lines correspond to using Equation (11) with different β values 0.2, 0.5, 0.8. Lower β values detect fewer lists (or it takes more HITs to detect lists).

Our results show that our heuristic is able to detect when multiple workers are consulting the same list and how severe list walking is. For example, it reports that for the UN 2 experiment around 20–25% of all HITs are impacted by list walking. The States experiments had little or no list walking. Despite webpages that show the list of US states, we posit

Figure 8. List walking in UN experiment run 2.



that workers did not find thinking of the states' names too difficult to warrant consulting the web.

The open-ended use case also contained little or no list walking. In the future, we plan to detect list walking and switch to scraping information from the web.

6. COST VERSUS BENEFIT: PAY-AS-YOU-GO

For sets with finite membership, it makes sense to estimate the set size and employ the crowd to provide the complete set. However, the result set for some queries may have unbounded size, a highly skewed distribution and/or extreme worker behavior that make predicting its size nonsensical, as discussed in Section 3. For these cases, it makes more sense to try to estimate the benefit of spending more money, that is, predicting the shape of the SAC (e.g., Figure 1) in the near future.

Again, we leverage species estimation algorithms to create *pay-as-you-go* techniques to predict this cost versus benefit tradeoff of getting more answers by expending additional effort. We evaluated the effectiveness of an estimator developed by Shen et al.[21] in determining how many unseen items would arrive with m additional answers from workers; this analysis would be done after having already received n answers (HITs). In general, we found that predictions for small m are more accurate since only the near future is considered. The larger the m, the further the prediction has to reach and thus the more error-prone the result, particularly if m exceeds the current HITs size n.[21] A full description of the pay-as-you-go approach and experimental results is in Trushkowsky et al.[22]

7. RELATED WORK

In this paper, we focused on estimating progress toward completion of a query result set, an aspect of query quality. Techniques have been proposed for quality control that can be used for verification of individual set elements.[10, 16] This work was done as part of CrowdDB,[11] but it could be applied to other hybrid database systems that use the crowd to enumerate sets.

Our estimation techniques build on top of existing work on species estimation.[3, 6, 9] These techniques have also been extended in database literature for distinct value estimation,[8, 14] but this work does not consider how human behaviors impact the sampling process of crowdsourced queries.

Species estimation techniques have been explored for search and meta-search engines. In Broder et al.[2] the authors develop an algorithm to estimate the size of any set of documents defined by certain conditions based on previous queries. Liu et al.[17] describes an algorithm to estimate the corpus size for a meta-search engine to better direct queries to search engines. Similar techniques are also used to measure the quality of search engines.[1] Recent work explores species estimation techniques for the deep web,[18] however, it does not consider the specific worker behavior and assumes sampling with replacement.

8. CONCLUSION

People are particularly well-suited for gathering new information because they have access to both real-life experience and online sources of information. Incorporating crowdsourced information into a database, however, raises questions regarding the meaning of query results without the closed-world assumption—how does one even reason about a simple `SELECT *` query? We argue that progress estimation allows the user to make sense of query results in the open world. By calculating an estimate of the expected result set size, or cardinality, for the enumeration query, an estimate of how complete the set is can be formed.

The species estimation literature provides a starting point to tackle cardinality estimation. However, applying existing estimators to sequences of responses from the crowd yields inaccurate results.

Instead of a with-replacement sample from a single item distribution, crowdsourced enumeration is a with-replacement sampling process drawing from many without-replacement sampling processes over potentially different distributions.

A particularly troublesome issue is the presence of "streakers," workers who complete many more HITs than other workers, causing the estimator to wildly overestimate cardinality. To ameliorate the problems caused by these overzealous workers, we develop a *streaker-tolerant* estimator. Using enumeration experiments run on AMT, we show that this estimator successfully reduces the impact of streakers and generates a more accurate estimate.

In this paper, we investigated one common operation used in a RDBMS: enumerating the items from a crowdsourced relation. Of course, modern database systems support many other operations, called relational operators, that can be composed to form more expressive queries.

We believe the result set size estimation techniques we described can be used with these more expressive queries. Some operators will be simple to apply as a subsequent processing step on the result of the crowdsourced enumeration. If each item provided by the crowd is processed to produce a single item, it is straightforward to apply our cardinality estimation techniques to reason about completeness of the final query result. An example is the `PROJECT` operator, which returns to the user only part of the information about each item, such as returning only the scientific name for each plant in the plants query. In contrast, consider the `JOIN` operator, which combines data from multiple relations, a process which typically matches pairs of items from two relations with respect to some predicate. The estimation of result set size in this case may need to incorporate statistics about the likelihood of items matching. Another common operation in RDBMSs is aggregation, for example, determining the average height of plants that can tolerate low light. The average could be calculated using the current set of responses received from crowd workers; cardinality estimation techniques may then be helpful in estimating the confidence of this average, or how close the current calculation is to the true value. Additionally, recent work has explored how to use crowd input to reduce the uncertainty of query answers over dirty and incomplete datasets for such operators.[23]

Incorporating the crowd into query processing systems presents the opportunity to extend the usefulness of these systems, but also raises fundamental questions about the meaning of query results. The cardinality estimation methods we developed and described in this paper enable query progress estimation, thereby allowing users to make sense of query results from hybrid human/machine database systems. By adapting statistical techniques, we enable users to reason about query completeness in the open world.

Acknowledgments

This work was inspired by the CrowdDB project, and we would like to thank our CrowdDB collaborators Donald Kossmann, Sukriti Ramesh, and Reynold Xin.

This research is supported in part by a NSF graduate fellowship, in part by NSF CISE Expeditions award CCF-1139158, and gifts from Amazon, Google, SAP, Blue Goji, Cisco, Cloudera, Ericsson, General Electric, Hewlett Packard, Huawei, Intel, Microsoft, NetApp, Oracle, Quanta, Splunk, VMware, and by DARPA (contract #FA8650-11-C-7136).

References

1. Bar-Yossef, Z., Gurevich, M. Efficient search engine measurements. *ACM Trans. Web 5*, 4 (Oct. 2011), 18:1–18:48.
2. Broder, A., Fontura, M., Josifovski, V., Kumar, R., Motwani, R., Nabar, S., Panigrahy, R., Tomkins, A., Xu, Y. Estimating corpus size via queries. In *Proceedings of CIKM* (2006).
3. Bunge, J., Fitzpatrick, M. Estimating the number of species: A review. *J. Am. Stat. Assoc. 88*, 421 (1993), 364–373.
4. Bunge, J., et al. Comparison of three estimators of the number of species. *J. Appl. Stat. 22*, 1 (1995), 45–59.
5. Burnham, K.P., Overton, W.S. Estimation of the size of a closed population when capture probabilities vary among animals. *Biometrika 65*, 3 (1978), 625–633.
6. Chao, A. Species estimation and applications. In *Encyclopedia of Statistical Sciences*, 2nd edn. N. Balakrishnan, C.B. Read, and B. Vidakovic, eds. Wiley, New York, 2005, 7907–7916.
7. Chao, A., Lee, S. Estimating the number of classes via sample coverage. *J. Am. Stat. Assoc. 87*, 417 (1992), 210–217.
8. Charikar, M., et al. Towards estimation error guarantees for distinct values. In *Proceedings of the PODS* (2000).
9. Colwell, R.K., Coddington, J.A. Estimating terrestrial biodiversity through extrapolation. *Philos. Trans. Biol. Sci. 345*, 1311 (1994), 101–118.
10. Doan, A., et al. Crowdsourcing applications and platforms: A data management perspective. *PVLDB 4*, 12 (2011), 1508–1509.
11. Franklin, M.J., et al. CrowdDB: Answering queries with crowdsourcing. In *Proceedings of the SIGMOD* (2011).
12. Good, I.J. The population frequencies of species and the estimation of population parameters. *Biometrika 40*, 3/4 (1953), 237–264.
13. Gray, J., et al. Quickly generating billion-record synthetic databases. In *Proceedings of the SIGMOD* (1994).
14. Haas, P.J., et al. Sampling-based estimation of the number of distinct values of an attribute. In *Proceedings of the VLDB* (1995).
15. Heer, J., et al. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *Proceedings of the CHI* (2010).
16. Ipeirotis, P.G., Provost, F., Wang, J. Quality management on Amazon mechanical turk. In *Proceedings of the HCOMP* (2010).
17. Liu, K.-L., Yu, C., Meng, W. Discovering the representative of a search engine. In *Proceedings of the CIKM* (2002).
18. Lu, J., Li, D. Estimating deep web data source size by capture–recapture method. *Inf. Retr. 13*, 1 (Feb. 2010), 70–95.
19. Marcus, A., Wu, E., Madden, S., Miller, R. Crowdsourced databases: Query processing with people. In *Proceedings of the CIDR* (2011).
20. Parameswaran, A., Polyzotis, N. Answering queries using humans, algorithms and databases. In *Proceedings of the CIDR* (2011).
21. Shen, T., et al. Predicting the number of new species in further taxonomic sampling. *Ecology 84*, 3 (2003).
22. Trushkowsky, B., Kraska, T., Franklin, M.J., Sarkar, P. Crowdsourced enumeration queries. In *Proceedings of the ICDE* (2013).
23. Wang, J., et al. A sample-and-clean framework for fast and accurate query processing on dirty data. In *Proceedings of the SIGMOD* (2014), 469–480.

Beth Trushkowsky (beth@cs.hmc.edu), Department of Computer Science, Harvey Mudd College, Claremont, CA.

Tim Kraska (tim_kraska@brown.edu), Department of Computer Science, Brown University, Providence, RI.

Michael J. Franklin (franklin@cs.berkeley.edu), Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA.

Purnamrita Sarkar (purna.sarkar@austin.utexas.edu), Department of Statistics and Data Sciences, University of Texas at Austin, Austin, TX.

Boise State University
Department of Computer Science
Eight Open Rank, Tenured/Tenure-Track Faculty Positions

The **Department of Computer Science at Boise State University** invites applications for **eight** open rank, tenured/tenure-track faculty positions. Seeking applicants in the areas of big data (including distributed systems, HPC, machine learning, visualization), cybersecurity, human computer interaction and computer science education research. Strong applicants from other areas of computer science will also be considered.

Applicants should have a commitment to excellence in teaching, a desire to make significant contributions in research, and experience in collaborating with faculty and local industry to develop and sustain funded research programs. A PhD in Computer Science or a closely related field is required by the date of hire. For additional information, please visit http://coen.boisestate.edu/cs/jobs.

Bowling Green State University
Tenure-track Assistant Professor of Computer Science

We are seeking applicants for a tenure-track position at the assistant professor level to teach a variety of courses at the undergraduate and graduate levels and be productive in scholarly research and sponsored projects. Preferred area of specialization is software engineering, including but not limited to: software testing and quality assurance, software architecture and design, usability engineering, and software verification. Applicants must hold a Ph.D. in CS (or closely related field) or complete it by August 2016, and be committed to excellence in teaching, scholarly research, and external funding. BGSU is an AA/EEO/Vet employer. We encourage applications from women, minorities, veterans, and individuals with disabilities regardless of age, gender identity, genetic information, religion, or sexual orientation. Email a letter of interest, along with curriculum vitae, statement of teaching philosophy and research agenda, contact information for three professional references, and copies of all transcripts by Sunday, January 10, 2016 to cssearch@bgsu.edu. We will contact your references. We will select a small number of finalists to come to campus for an interview. An official transcript of the terminal degree and a background check are also required for employment. For details, go to http://www.bgsu.edu/arts-and-sciences/computer-science.html/jobs

Bradley University
Computer Science and Information Systems Department
Tenure-Track Assistant Professor

The Computer Science and Information Systems Department at **Bradley University** invites applications for a tenure-track Assistant Professor position starting in August 2016. 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 2016 will be considered. Please visit www.bradley.edu/humanresources/opportunities for full position description and application process.

California State University, Chico
Assistant Professor

Dept. of Computer Science has one fulltime tenure track Asst. Prof position, starting 8/2016. EOE Employer. Please see the full Announcement at: http://jobs.csuchico.edu/postings/3255

California State University, Sacramento
Department of Computer Science
Two Tenure-Track positions

California State University, Sacramento, Department of Computer Science. Two Tenure-Track positions to begin with the Fall 2016 semester. One position is in Database Systems (Assistant or Associate Professor) and the other is open to any area of Computer Science (Assistant Professor). Ph.D. in Computer Science, Computer Engineering, or closely related field required by the time of appointment. For detailed position information, including application procedure, please see http://www.csus.edu/about/employment/. Screening will begin February 1, 2016, and continue until positions are filled. AA/EEO employer. Clery Act statistics available. Mandated reporter requirements. Background check (including criminal) will be required.

Carnegie Mellon University
Human-Computer Interaction Institute
Tenure-Track Faculty

The **Human-Computer Interaction Institut**e at **Carnegie Mellon University** is hiring tenure-track faculty this year. We are looking for a wide range of expertise. More information can be found at: http://hcii.cmu.edu/careers/2015/carnegie-mellon-hcii-hiring-tenure-track-faculty

Carnegie Mellon University
Computer Science Department
Teaching Track Position

Applications are invited for a teaching-track position in Computer Science, beginning Fall 2016. This is a renewable, career-oriented position with an initial appointment for three years. We seek highly qualified applicants with a strong commitment to excellence in teaching and the ability to teach at all levels in the undergraduate curriculum.

Applicants for the position must have a Ph.D. in Computer Science or a related field, and demonstrated excellence in teaching Computer Science courses. Teaching-track appointments are typically at the rank of Assistant Teaching Professor, with the possibility of promotion to the ranks of Associate Teaching Professor and Teaching Professor. None of these ranks are tenured; applicants seeking a tenure-track position at a research university are therefore not a good match for these positions.

In order to receive full consideration, applicants should submit a letter of application, curriculum vitae, a statement of teaching philosophy, and the names and email addresses of three or more individuals whom the applicant has asked to provide letters of reference. All of this information should be provided at the website listed below. Note that applicants should also arrange for reference letters to be sent directly to this site. The review process will start January 4, 2016, and continue until the position is filled.

Additionally, applicants are encouraged to submit a video sample of their teaching, demonstrating their excellence.

Please upload your applications and accompanying materials at

https://csd.cs.cmu.edu/content/faculty-hiring

Carnegie Mellon is an affirmative action/equal opportunity employer and we invite and encourage applications from women and underrepresented minorities.

Florida State University
Department of Computer Science
Tenure-Track Assistant Professor Positions

The Department of Computer Science at the Florida State University invites applications for two tenure-track Assistant Professor positions to begin August 2016. The positions are 9-mo, full-time, tenure-track, and benefits eligible. We are seeking outstanding applicants with strengths in the broad areas of Data Sciences or Trust-worthy Computing. The Data Sciences area may involve Artificial Intelligence, Data Mining, Data Engineering, Databases, Data Analytics, Big Data Applications and Systems, Computer Graphics and Visualization, as well as other Data Sciences related research. The Trust-worthy Computing area includes Security, Digital Forensics, Software Engineering, Formal Methods and Verification, Programming Languages and Compilers, Embedded Systems, Computer Architecture, and Cyber-Physical Systems among others. Applicants should hold a PhD in Computer Science or closely related field at the time of appointment, and have excellent research and teaching accomplishments or potential. The department offers degrees at the BS, MS, and PhD levels. The department is an NSA Center of Academic Excellence in Information Assurance Education (CAE/IAE) and Research (CAE-R).

FSU is classified as a Carnegie Research I university. Its primary role is to serve as a center for advanced graduate and professional studies while emphasizing research and providing excellence in undergraduate education. Further information can be found at:

http://www.cs.fsu.edu

Screening will begin January 1, 2016 and will continue until the positions are filled. Please apply online with curriculum vitae, statements of teaching and research philosophy, and the names of three references, at:

http://www.cs.fsu.edu/positions/apply.html

Questions can be e-mailed to Prof. Xiuwen Liu, Faculty Search Committee Chair, recruitment@cs.fsu.edu.

Equal Employment Opportunity

An Equal Opportunity/Access/Affirmative Action/Pro Disabled & Veteran Employer committed to enhancing the diversity of its faculty and students. Individuals from traditionally underrepresented groups are encouraged to apply.

FSU's Equal Opportunity Statement can be viewed at:

http://www.hr.fsu.edu/PDF/Publications/diversity/EEO_Statement.pdf

Maharishi University of Management

Assistant/Associate/Full Professor of Computer Science

A Unique Opportunity to Combine Personal Growth with Career Development

The Computer Science Department of Maharishi University of Management (M.U.M.) is seeking additional full-time faculty at the Assistant, Associate, and Full Professor levels beginning Spring 2016, or Fall 2016. The primary responsibility is teaching graduate-level computer science courses in our rapidly growing M.S. program with an emphasis on applied software development. With over 400 new M.S. students entering each year, it is one of the most successful programs of its kind.

Qualifications include Ph.D. in Computer Science (or closely related area), and significant experience with the Java programming language and its associated libraries and frameworks. We are particularly seeking candidates with the ability to teach courses in Web Application Programming, Cloud and Mobile Computing, Big Data, and Object-Oriented Software Engineering. Salary is competitive and based on background and experience.

The University: M.U.M. provides a unique approach to education where the most current values of modern education combine with Consciousness-Based Education, which includes systematic methods to develop the creativity and inner intelligence of the student, to fulfill the highest ideals of education. This innovative educational program has gained worldwide recognition for academic excellence, and creates an exciting and dynamic educational environment for both students and faculty. While it is not uncommon for a university to have such a goal as a part of its philosophy, M.U.M. is the only university that has a systematic and proven concrete technology to realize this goal.

Campus: M.U.M. is located on a 370-acre campus in Fairfield, a small town in southeast Iowa. Located in a quiet rural setting, Fairfield is a vibrant community and offers many of the advantages of larger city life, but also with the benefits of smaller town living. The University creates a rich multi-cultural environment in the town. Fairfield is home to a wide range of unique social and business activities, and a wonderful place for both personal and family life. The Stephen Sondheim Center for the Performing Arts in the heart of the city has weekly performances on stage, including concerts, dance, plays, and comedy.

To apply: Email curriculum vitae (pdf file) to csprof@mum.edu. Applications will be reviewed as they are received until all positions are filled. For further information about the University and its programs, see http://www.mum.edu/ and http://mscs.mum.edu/. M.U.M. is an equal opportunity employer.

Max Planck Institute for Informatics

Junior Research Group Leader

The Max Planck Institute for Informatics, as the coordinator of the Max Planck Center for Visual Computing and Communication (MPC-VCC), invites applications for Junior Research Group **Leaders in the Max Planck Center for Visual Computing and Communicatio**n.

The Max Planck Center for Visual Computing and Communications offers young scientists in information technology the opportunity to develop their own research program addressing important problems in areas such as image communication

Tenure-Track Faculty Information Science Department – Cornell University

Cornell is a community of scholars, known for intellectual rigor and engaged in deep and broad research, teaching tomorrow's thought leaders to think otherwise, care for others, and create and disseminate knowledge with a public purpose.

The Information Science Department at Cornell University invites applications for tenure-track faculty positions. Exceptional candidates in all areas related to the department's current research trajectories and priorities will be given serious consideration; these include human-computer interaction (HCI) and interaction design; computer-supported cooperative work (CSCW) and computer-mediated communication (CMC); information policy; network science; crowdsourcing; the history and anthropology of computing and data; the interface of economics and information; critical and interpretive analysis of information systems; human-robot interaction (HRI); ubiquitous computing; applications and analysis of large datasets; information visualization; the sociology of organizations and innovation; and information science approaches to societal challenges. We invite applicants at any rank. Assistant Professor candidates must receive a Ph.D. or equivalent degree by August 2016, and must demonstrate the potential to achieve excellence in research and teaching at both the graduate and undergraduate levels. More senior candidates should hold a Ph.D. or equivalent degree and must have an established record of outstanding research and excellent teaching at both the graduate and undergraduate levels; salary and rank will be commensurate with qualifications and experience. Experienced applicants may merit a tenured Associate Professor or Professor position, depending on their qualifications.

Applicants should submit a cover letter, curriculum vita (CV), brief statements of research and teaching interests, and arrange to have at least three reference letters submitted. In the cover letter, CV, or research statement, applicants should identify a small set of their most significant pieces of work.

Information Science at Cornell University brings together faculty, students and researchers who share an interest in advancing our understanding of how people and society interact with computing and information. The Information Science Department is housed in the Faculty of Computing and Information Science, and located in Cornell's new Gates Hall. It has strong connections with several other units on campus, including: Computer Science (with which it shares Gates Hall), Communication, Economics, Sociology, Science and Technology Studies, Operations Research and Information Engineering, and Cognitive Science.

The Information Science Department at Cornell embraces diversity and seeks candidates who will create a climate that attracts students of all races, nationalities and genders. Cornell University is an affirmative action, equal opportunity employer and educator.

Cornell University seeks to meet the needs of dual career couples, has a Dual Career program, and is a member of the Upstate New York Higher Education Recruitment Consortium to assist with dual career searches.

Applications will be accepted until the position is filled. Information about the Information Science Department appears at **www.infosci.cornell.edu**.

Application Deadline: 12.15.2015 or until positions are filled.

Cornell University is an innovative Ivy League university and a great place to work. Our inclusive community of scholars, students and staff impart an uncommon sense of larger purpose and contribute creative ideas to further the university's mission of teaching, discovery and engagement. Located in Ithaca, NY, Cornell's far-flung global presence includes the medical college's campuses on the Upper East Side of Manhattan and in Doha, Qatar, as well as the new CornellNYC Tech campus to be built on Roosevelt Island in the heart of New York City.

Diversity and Inclusion are a part of Cornell University's heritage. We're an employer and educator recognized for valuing AA/EEO, Protected Veterans, and Individuals with Disabilities.

- ▸ computer graphics
- ▸ geometric computing
- ▸ imaging systems
- ▸ computer vision
- ▸ human machine interface
- ▸ distributed multimedia architectures
- ▸ multimedia networking
- ▸ visual media security.

The center includes an outstanding group of faculty members at Stanford's Computer Science and El¬ectrical Engineering Departments, the Max Planck Institute for Informatics, and Saarland University.

The program begins with a preparatory 1-2 year postdoc phase (**Phase P**) at the Max Planck Institute for Informatics, followed by a two-year appointment at Stanford University (**Phase I**) as a visiting assistant professor, and then a position at the Max Planck Institute for Informatics as a junior research group leader (**Phase II**). However, the program can be entered flexibly at each phase, commensurate with the experience of the applicant.

Applicants to the program must have completed an outstanding PhD. Exact duration of the preparatory postdoc phase is flexible, but we typically expect this to be about 1-2 years. Applicants who completed their PhD in Germany may enter Phase I of the program directly. Applicants for Phase II are expected to have completed a postdoc stay abroad and must have demonstrated their outstanding research potential and ability to successfully lead a research group.

Reviewing of applications will commence on **01 Jan 2016**. The final deadline is **31 Jan 2016**. Applicants should submit their CV, copies of their school and university reports, list of publications, reprints of five selected publications, names of 3-5 references, a brief description of their previous research and a detailed description of the proposed research project (including possible opportunities for collaboration with existing research groups at Saarbrücken and Stanford) to:

Prof. Dr. Hans-Peter Seidel
Max Planck Institute for Informatics,
Campus E 1 4, 66123 Saarbrücken, Germany;
Email: mpc-vcc@mpi-inf.mpg.de

The Max Planck Center is an equal opportunity employer and women are encouraged to apply.

Additional information is available on the website www.mpc-vcc.org

Max Planck Institute for Software Systems

Tenure-track openings

Applications are invited for tenure-track faculty positions in all areas related to the theory and practice of software systems, including security and privacy, embedded and mobile systems,

computational social science, legal, economic, and social aspects of computing, NLP, machine learning, information and knowledge management, programming languages, verification, parallel and distributed systems.

A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups.

MPI-SWS, founded in 2005, is part of a network of over 80 Max Planck Institutes, Germany's premier basic research facilities. MPIs have an established record of world-class, foundational research in the sciences, technology, and the humanities. The institute offers a unique environment that combines the best aspects of a university department and a research laboratory: Faculty enjoy academic freedom, receive institutional funding and attract additional third-party funds to build and lead a team of graduate students and post-docs; they supervise doctoral theses, and have the opportunity to teach graduate and undergraduate courses. The institute offers outstanding technical infrastructure and administrative support, as well as internationally competitive compensation.

The institute is located in Kaiserslautern and Saarbruecken, in the tri-border area of Germany, France and Luxembourg. We maintain an international and diverse work environment and seek applications from outstanding researchers worldwide. The working language is English;

knowledge of the German language is not required for a successful career at the institute.

Qualified candidates should apply at "https://apply.mpi-sws.org/". To receive full consideration, applications should be received by December 15, 2015.

The institute is committed to increasing the representation of minorities, women and individuals

FIU | FLORIDA INTERNATIONAL UNIVERSITY

Florida International University is a comprehensive university offering 340 majors in 188 degree programs in 23 colleges and schools, with innovative bachelor's, master's and doctoral programs across all disciplines including medicine, public health, law, journalism, hospitality, and architecture. FIU is Carnegie-designated as both a research university with high research activity and a community-engaged university. Located in the heart of the dynamic south Florida urban region, our multiple campuses serve over 55,000 students, placing FIU among the ten largest universities in the nation. Our annual research expenditures in excess of $100 million and our deep commitment to engagement have made FIU the go-to solutions center for issues ranging from local to global. FIU leads the nation in granting bachelor's degrees, including in the STEM fields, to minority students and is first in awarding STEM master's degrees to Hispanics. Our students, faculty, and staff reflect Miami's diverse population, earning FIU the designation of Hispanic-Serving Institution. At FIU, we are proud to be 'Worlds Ahead'! For more information about FIU, ***visit fiu.edu***.

The School of Computing and Information Sciences (SCIS) seeks exceptionally qualified candidates for tenure-track and tenured faculty positions at all levels as well as non-tenure track faculty positions at the level of Instructor, including visiting instructor appointments. SCIS is a rapidly growing program of excellence at the University, with 30 tenure-track faculty members and over 2,000 students, including over 80 Ph.D. students. SCIS offers B.S., M.S., and Ph.D. degrees in Computer Science, an M.S. degree in Telecommunications and Networking, an M.S. degree in Cybersecurity, and B.S., B.A., and M.S. degrees in Information Technology. SCIS has received over $22M in the last four years in external research funding, has six research centers/clusters with first-class computing and support infrastructure, and enjoys broad and dynamic industry and international partnerships.

Open-Rank Tenure Track/Tenured Positions (Job ID# 508676)

SCIS seeks exceptionally qualified candidates for tenure-track and tenured faculty positions at all levels. We seek well-qualified candidates in all areas; researchers in the areas of computer systems, cybersecurity, cognitive computing, data science, health informatics, and networking are particularly encouraged to apply. Preference will be given to candidates who will enhance or complement our existing research strengths.

Ideal candidates for junior positions should have a record of exceptional research in their early careers. Candidates for senior positions must have an active and proven record of excellence in funded research, publications, and professional service, as well as a demonstrated ability to develop and lead collaborative research projects. In addition to developing or expanding a high-quality research program, all successful applicants must be committed to excellence in teaching at both the graduate and undergraduate levels. An earned Ph.D. in Computer Science or related disciplines is required.

Non-tenure track instructor positions (Job Opening 507474)

We seek well-qualified candidates in all areas of Computer Science and Information Technology. Ideal candidates must be committed to excellence in teaching a variety of courses at the undergraduate level. A graduate degree in Computer Science or related disciplines is required; significant prior teaching and industry experience and/or a Ph.D. in Computer Science is preferred.

HOW TO APPLY:

Qualified candidates for open- rank faculty positions are encouraged to apply to (Job Opening ID #508676); and candidates for instructor positions are encouraged to apply to (Job Opening ID # 507474). **Submit applications at *facultycareers.fiu.edu*** and attach cover letter, curriculum vitae, statement of teaching philosophy, research statement, etc as individual attachments. Candidates will be required to provide names and contact information for at least three references who will be contacted as determined by the search committee. To receive full consideration, applications and required materials should be received by December 31st, 2015. Review will continue until position is filled.

If you are interested in a visiting appointment please contact the department directly by emailing Dr. Mark Weiss at *Weiss@cis.fiu.edu*. All other applicants should apply by going to *facultycareers.fiu.edu*.

FIU is a member of the State University System of Florida and an Equal Opportunity, Equal Access Affirmative Action Employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, disability status, protected veteran status, or any other characteristic protected by law.

with physical disabilities in Computer Science. We particularly encourage such individuals to apply.

The initial tenure-track appointment is for five years; it can be extended to seven years based on a midterm evaluation in the fourth year. A permanent contract can be awarded upon a successful tenure evaluation in the sixth year.

National College of Ireland
Lecturing Vacancies

National College of Ireland is a third-level education provider committed to advancing knowledge in its specialist areas of computing, human resource management, psychology, entrepreneurship, accountancy, finance and business. The college offers full and part-time courses to over 3,500 students in our schools of Computing and Business.

The School of Computing has greatly expanded in the last few years, becoming one of the biggest computing departments in Ireland with over 1,400 students at undergraduate and postgraduate level including a PhD programme in Technology Enhanced Learning and a world-class Cloud Competency Centre. The School has over 60 faculty members with strong industrial links and research excellence in technology-enhanced learning, cloud computing, mobile technologies, data analytics, and parallel & distributed computing.

The School of Computing requires:
► Lecturer in Computing, 2 Permanent positions & 2 Three-year (Fixed Term Contract) positions (NCI0660), applicants must have a PhD in Computing, Computer Science or Informatics. Applicants should have lecturing and/or industry experience.

An established research track record in Data Analytics, Cloud Computing or eLearning is highly desirable. Other current areas of interest to school are Mobile computing, Internet of Things, Social Computing, Pervasive Computing and Enterprise Computing.

Appointments to the Lecturer positions will be made at either Academic I €36,000 - €54,000 or Academic II €52,000 - €78,000 Salary Bands with 34 days annual leave per annum.

To view the job description or for further information, please visit: www.ncirl.ie and select Job Vacancies.

To apply for a position, please e-mail your Cover Letter and CV, quoting the reference number to recruitment@ncirl.ie Shortlisting will take place of applications received by the 15h February 2016. The closing date may be extended if necessary. It is anticipated that interviews will take place in March 2016.

National College of Ireland is an equal opportunities employer.

National Taiwan University
Professor-Associate Professor-Assistant Professor

The Department of Computer Science and Information Engineering, and the Graduate Institute of Networking and Multimedia at National Taiwan Univ. have faculty openings at all ranks beginning in August 2016. Highly qualified candidates in all areas of computer science are invited to apply. A Ph.D. or its equivalent is required. Applicants are expected to conduct outstanding research and be committed to teaching. Candidates should check http://www.csie.ntu.edu.tw/faculty_recruiting/ for submitting applications. The deadline is February 15, 2016. Contact Prof. Chih-Jen Lin at faculty_search@csie.ntu.edu.tw for any questions. An early submission is strongly encouraged.

Portland State University
Computer Science Department
Assistant Professor

The Computer Science Department at Portland State University (PSU) invites applications for multiple tenure-track assistant professor faculty positions to begin Fall 2016. Exceptional applicants at other ranks will also be considered.

Specific areas of computer science under consideration are: artificial intelligence; machine learning; data mining; data-intensive systems; secure and trustworthy cyberspace, and networks and systems of connected things.

Qualifications
A Ph.D. in Computer Science or other relevant field is required, and must be awarded prior to September 1, 2016. All applicants are expected to show great potential for future external research support and a demonstrated record of research excellence.

Faculty Positions in
Computer Science

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of Science and Technology

The Computer, Electrical, and Mathematical Sciences and Engineering Division at King Abdullah University of Science and Technology (KAUST) invites applications for faculty positions at all levels (Full, Associate, and Assistant Professor) in the following areas of Computer Science:

- Data Mining and analytics with emphasis on Big Data
- Machine Learning
- Artificial intelligence
- Emerging Architecture (GPU's, FPGA's, etc)
- High performance computing
- Computer security
- Visualization
- Computer graphics

High priority will be given to the overall originality and promise of the candidate's work rather than the candidate's sub-area of specialization within Computer Science. An earned PhD in Computer Science, Computer Engineering, Electrical Engineering, or a related field, and strong publication record in top-tier conferences and journals, are required. Senior candidates must have demonstrated strong leadership in the field. Women are strongly encouraged to apply. Also of particular interest are interdisciplinary candidates who can open new areas of investigation. A successful candidate will be expected to teach courses at the graduate level and to build and lead a research group of postdoctoral fellows and graduate students. Faculty members enjoy secure research funding from KAUST and have opportunities for additional funding through several KAUST provided sources and through industry collaborations.

KAUST is an international, graduate-only research university, located on the shores of the Red Sea in Saudi Arabia. KAUST offers superb research facilities, including the 5 Petaflop/s Shaheen-2 supercomputer and a large-scale virtual reality installation; generous assured research funding; and internationally competitive salaries. KAUST attracts top international faculty, scientists, engineers and students to conduct curiosity-driven and goal-oriented research to address the world's pressing scientific and technological challenges.

Please apply via the **http://apptrkr.com/705641** employment site. Please include the names of three references for Assistant Professor positions and six for senior positions. Applications will be considered until the positions are filled but not later than April 15, 2016. Prospective candidates are advised to apply as soon as possible.

www.kaust.edu.sa

Job Specifications
The faculty member will maintain scholarly activity in funded research and publications; teach undergraduate and graduate classes; provide professionally related public service; advise students, and support University activities through committee service.

To Apply
For more information and for instructions on how to apply, please visit https://jobs.hrc.pdx.edu/postings/17409. For inquiries about this position, please contact cssearch@pdx.edu. Review of applications will begin immediately and will continue until the positions are filled.

Portland State University is an Affirmative Action, Equal Opportunity Institution and welcomes applications from diverse candidates and candidates who support diversity.

Princeton University
Department of Computer Science
Postdoctoral Research Associate- Theoretical Computer Science

The Department of Computer Science at Princeton University is seeking applications for postdoctoral or more senior research positions in theoretical computer science and theoretical machine learning. Positions are for one year with the possibility of renewal. Candidates must have a PhD in Computer Science or a related field by August 2016. To ensure full consideration, we encourage candidates to complete their applications, (including letters of recommendation) by December 15, 2015. Applicants should submit a CV and research statement, and contact information for three references. Princeton University is an Equal Opportunity /Affirmative Action Employer and all qualified applicants will receive consideration for employment without regard to race, color, religion, sex, sexual orientation, gender identity, national origin, disability status, protected veteran status, or any other characteristic protected by law. These positions are subject to the University's background check policy. Apply to: http://jobs.princeton.edu/ Requisition number: 1500926

Queens College
Assistant Professor of Computer Science

The Department of Computer Science at Queens College of CUNY is accepting applications for a tenure-track position in Computer Science at the Assistant Professor level starting Fall 2016. Consult http://www.cs.qc.cuny.edu for further information. Apply URL: http://www.qc.cuny.edu/HR/Pages/JobListings.aspx

State University of New York at Binghamton
Department of Computer Science
Four Tenure-Track Positions

As part of our continuing growth, the Department of Computer Science in the Thomas J. Watson School of Engineering and Applied Science at Binghamton University (The State University of New York at Binghamton) invites applications for four tenure-track positions as follows beginning in Fall 2016:

1) One Assistant Professor position in the healthcare informatics/healthcare systems area.
2)Three other positions, two at the Assistant professor level and one at the Assistant/Associate Professor level. For these 3 positions, we are especially interested in candidates in the following areas: (a) Data Analytics, (b) Operating Systems or Embedded Systems, (c) Machine Learning.

The Department has established graduate and undergraduate programs, including 60 full-time PhD students and 31 Faculty members. Junior faculty have a significantly reduced teaching load for at least the first three years. We are dedicated to the goal of building a diverse and inclusive teaching, research, and working environment. Potential applicants who share this goal, especially underrepresented minorities and women, are strongly encouraged to apply for the position described below. Further details and application information are available at: http://binghamton.interviewexchange.com

For the faculty position in the healthcare informatics/healthcare systems area the Department seeks a research scholar with research that will affiliate with the Binghamton University Transdisciplinary Areas of Excellence Initiative in Healthcare systems (http://www.binghamton.edu/tae/health-sciences/index.html) respectively. Applications will be reviewed until positions are filled. First consideration will be given to applications received by **February 15, 2016.**

We are an EE/AA employer.

Texas Southern University
Assistant Professor

Assistant Professor positions at CS Department (http://cs.tsu.edu/) of Texas Southern University. Candidates should have a Ph.D. in CS or a related discipline from an accredited institution, a record of research publications, a coherent research and teaching plan. Apply URL: https://jobs.tsu.edu/postings/1108

University College London
UCL Computer Science
Lecturer/Senior Lecturer/Reader in Computer Systems and Networking

The Department of Computer Science at University College London (UCL) invites applications for a faculty position (Lecturer/Senior Lecturer/Reader) in the areas of Computer Systems and Networking. We seek world-class talent; candidates must have an outstanding research track record.

Areas of interest for this position include operating systems, systems security, distributed systems, networking, and their intersection, with an emphasis on experimental system-building. Appointments will be made at the level of Lecturer, Senior Lecturer, or Reader (equivalent to Assistant Professor, junior Associate Professor, and senior Associate Professor, respectively, in the US system), commensurate with qualifications.

Candidates must hold an earned Ph.D. in Computer Science or a closely related field by the time they begin their appointment. They will be evaluated chiefly on the significance and novelty of their research to date, and their promise for leading a group in a fruitful program of research. They must also demonstrate a zest for innovative and challenging teaching at the graduate and undergraduate levels. A proven record of ability to manage time and evidence of ability to teach and to supervise academic work by undergraduates, masters, and doctoral students are desirable. Our department is a highly collaborative environment, and we seek future colleagues who enjoy working collaboratively. Candidates should further be committed to public communication, and to UCL's policy of equal opportunity, including working harmoniously with colleagues and students of all cultures and backgrounds.

Candidates seeking appointment at Senior Lecturer level must have experience in developing a new course. To be appointed at Reader-level (Grade 9) candidates should have shown leadership in their field as well as having an established research record. The appointment will be full time on UCL Grade 8/9. Plus a Networks specialism market supplement of £20,000 per annum.

For further details about the vacancy and

how to apply on line please go to http://www.ucl.ac.uk/hr/jobs/ and search on Reference Number: 1522730

Questions about this vacancy may be directed to Professor Brad Karp (Head of Systems and Networks Research Group, b.karp@cs.ucl.ac.uk), or Professor John Shawe-Taylor (Head of Department, j.shawe-taylor@ucl.ac.uk).

Closing Date: Friday 15th January 2016

Latest time for the submission of applications: 23:59.

Interview Date: TBC

We particularly welcome female applicants and those from an ethnic minority, as they are under-represented within UCL at this level.

UCL Taking Action for Equality

University of California, Santa Barbara

Department of Computer Science

Computer Science Lecturer with Potential for Security of Employment

The Department of Computer Science at the **University of California, Santa Barbara** seeks applications for a full-time **Lecturer** with Potential Security of Employment (similar to tenure-track) in the area of Computer Science with a start date of Fall quarter, 2016.

The primary criterion for this position is teaching ability of exceptional quality, and the promise of future growth. We seek an individual with enthusiasm for teaching undergraduate courses in a variety of areas of computer science, with a breadth of knowledge in computer science and its emerging applications in diverse fields. A successful applicant should be committed to teaching excellence, curriculum leadership and development, and student mentoring, all in collaboration with other faculty. Research, although not required for Lecturer with Security of Employment track positions, is also reviewed when available.

At the University of California, the positions of Lecturer with Potential Security of Employment lead to security of employment (similar to tenure), and are faculty positions designed to meet the long-term instructional needs of the University. Individuals appointed into this title engage in teaching, professional activities, and University and public service. The position is to begin with the 2016-17 academic year, and salary will be commensurate with experience.

A Ph.D. degree in Computer Science, Computer Engineering or a related field is required. Preference will be given to candidates with a demonstrated record of teaching excellence. Applications should include a cover letter, curriculum vitae, teaching statement (2-page limit), and 3 reference letters.

Applications must be submitted electronically at: https://recruit.ap.ucsb.edu/apply/JPF00636

Applications received by February 1, 2016 will be given priority consideration, but the position will remain open until filled. Candidates are urged to learn more about UCSB Computer Science Department at http://www.cs.ucsb.edu.

The department is especially interested in candidates who can contribute to the diversity and excellence of the academic community through research, teaching and service. The University of California is an Equal Opportunity/Affirmative Action Employer and all qualified applicants will receive consideration for employment without regard to race, color, religion, sex, sexual orientation, gender identity, national origin, disability status, protected veteran status, or any other characteristic protected by law.

University of Central Florida

Leader for Faculty Cluster in Cyber Security and Privacy

The University of Central Florida (UCF) is hiring 100 new faculty members to join us in Fall 2016. As part of this hiring campaign, UCF recently established six interdisciplinary clusters to strengthen its academic offerings and research mission. In support of this effort, we are recruiting faculty in the broad area of cyber security and privacy. To begin this process we will hire a tenured (or tenure-track) associate or full professor to lead the cyber security and privacy cluster. This position has a start date of Fall 2016.

This will be an interdisciplinary position that will be expected to strengthen both a chosen tenure home department and a possible combination of joint appointments. A strong advantage of this position is the ability of the candidate to choose a combination of units from the cluster for their appointment. (See http://www.ucf.edu/research/cyber for a complete list of all the units involved.) Both individual and interdisciplinary infrastructure and startup support will be provided with this new position.

The ideal candidate will have a strong background in cyber security and privacy, and will be recognized for leadership in this area and research impact, as reflected in a sustained record of high quality publications and external funding. We are looking for a leader to bring together all the current campus efforts in cyber security and privacy, and to play a key role in hiring four additional faculty that will constitute the cluster's core expertise in this area. These four additional cluster hires will happen in the 2016-17 academic year. The cluster includes core members from Computer Science, Electrical and Computer Engineering, Industrial Engineering and Management Systems, the Institute for Simulation and Training, Political Science, Psychology, and Statistics, and affiliated members from several other departments. This cluster will be the focal point of the University's research and educational efforts in cyber security and privacy.

Applicants must have a Ph.D. from an accredited institution in an area appropriate to the cluster, and a record of high impact research related to cyber security and privacy demonstrated by a strong scholarly record and significant amount of sustained funding. A history of working with teams, especially teams that span multiple disciplines, is a strongly preferred qualification. The position will carry a rank of (full) professor, or associate professor commensurate with the candidate's prior experience and record.

Located in Orlando, UCF is one of the nation's most dynamic metropolitan research universities. UCF has the top-tier Carnegie Foundation designation of a "very high research activity" university, is the nation's second-largest university, and is ranked by U.S. News and World Report as among the top up-and-coming universities in terms of innovative changes in the areas of academics, faculty, and student life. UCF's Orlando location also puts it at the center of the Florida High Tech Corridor. The corridor has an excellent industrial base that includes: software, defense, space, simulation and training, and a world-renowned entertainment industry. Adjacent to UCF is a thriving research park that conducts over $2 billion in funded research, hosting more than 100 high-technology companies and UCF's Institute for Simulation and Training. The Central Florida area is designated by the State of Florida as the Center of Excellence in Modeling and Simulation. UCF also has an accredited medical school, which was established in 2006. UCF is a neighbor to large corporations, such as Disney, Harris Corporation, Lockheed Martin, Siemens, and many others, all of which have a strong interest in cyber security and privacy. Great weather, easy access to the seashore, one of the largest convention centers in the nation, and one of the world's best airports are just a few features that make Orlando an ideal location.

Candidates must apply online at http://www.jobswithucf.com/postings/43416 (Position #37317) and attach the following materials: a cover letter, curriculum vitae, teaching statement, research statement, and contact information for three professional references. In the cover letter candidates should address their background in cyber security and privacy, and should identify the department for their potential tenure home and the joint appointments they would desire.

UCF is an Equal Opportunity/Affirmative Action employer. All qualified applicants are encouraged to apply, including minorities, women, veterans and individuals with disabilities. As a Florida public university, UCF makes all application materials and selection procedures available to the public upon request.

For more information about these positions please contact the Cluster Search Chair, Gary T. Leavens, at Leavens@ucf.edu.

University of Delaware

Tenure-Track Assistant Professor in Networking

Department of Computer and Information Sciences

Applications are invited for a tenure-track assistant professor position in Networking and associated research areas to begin Fall 2016. For application, please visit apply.interfolio.com/32786.

EOE

University of Illinois at Chicago

Department of Computer Science

Computer Security Faculty Positions

The University of Illinois at Chicago Computer Science Department invites applications for one or more tenure-track positions in cybersecurity, broadly defined, at the rank of Assistant Professor. Exceptional candidates at other ranks may also be considered. Candidates in related areas such as programming languages and compilers, computer systems and software engineering whose research has a strong connection to cybersecurity are also encouraged to apply.

The University of Illinois at Chicago (UIC) is ranked fourth among US Universities under 50 years old. The Computer Science department has recently grown to 29 tenure-track faculty, and offers BS, MS, and PhD degrees. Our faculty includes 11 NSF CAREER award recipients. UIC is an excellent place for interdisciplinary work—with the largest medical school in the country and faculty engaged in several cross-departmental collaborations with health sciences, social sciences and humanities, urban planning, and the business school.

Our faculty have a broad range of research interests in fundamental and practical aspects of cybersecurity. Key research areas include the security of cyber-physical systems, web and mobile applications, operating systems, applications and network security, cryptography and protocols, online fraud detection, privacy and information flow including privacy-preserving data management and mining, and technology policy. Our research is funded by grants from NSF, DARPA, AFOSR, ONR, and DHS and our annual research expenditures for cybersecurity are at $1.5M. Our research program is complimented by a strong educational program with federally funded fellowships at the BS, MS, and PhD levels, including one of the country's three IGERT programs in cybersecurity.

Chicago epitomizes the modern, livable, vibrant city. Located on the shore of Lake Michigan, it offers an outstanding array of cultural and culinary experiences. As the birthplace of the modern skyscraper, Chicago boasts one of the world's tallest and densest skylines, combined with an 8100-acre park system and extensive public transit and biking networks. Its airport provides daily non-stop flight service to about 150 US cities and 50 international destinations. Yet the cost of living—whether in a 99th floor condominium downtown or on a tree-lined boulevard in one of the nation's finest school districts—is surprisingly low.

Applications must be submitted at https://jobs.uic.edu/. Please include a CV, teaching and research statements, and names and addresses of at least three references in the online application. For full consideration, applications must be received by Jan 3rd, 2016. Applicants needing additional information may contact the Faculty Search Chair for Security at security-search@cs.uic.edu. The University of Illinois is an Equal Opportunity, Affirmative Action employer. Minorities, women, veterans and individuals with disabilities are encouraged to apply.

University of Illinois at Chicago

Department of Computer Science

Data Science/Human-Computer Interaction Faculty

The Computer Science Department at the University of Illinois at Chicago (UIC) invites applications for multiple full-time tenure-track positions at the rank of Assistant Professor (exceptional senior level candidates will also be considered). All candidates must have a doctorate in Computer Science or a related field by the starting date of the appointment. Candidates will be expected to conduct world class research, collaborate with faculty from a wide range of disciplines, and teach effectively at the undergraduate and graduate levels. Senior candidates must have an outstanding research record, a strong record of funded research, demonstrated leadership in collaborative research, and an excellent teaching record at the undergraduate and graduate level.

This search seeks candidates in the following two areas. Please clearly indicate for which one of those areas you wish to be considered. Exceptional candidates from closely related areas may also be considered.

Data Science. Research spanning all aspects of scalable information retrieval, data management, and data integration are of particular interest.

Human-Computer Interaction. Research spanning all areas of humans interacting with computers, ubiquitous computing, wearable technology, computer-supported cooperative work (CSCW), and crowdsourcing.

The University of Illinois at Chicago (UIC) ranks is ranked 4th best U.S. University under 50 years old by Times Higher Education. The Computer Science department has 29 tenure-track faculty and offers BS, MS and PhD degrees. Our faculty includes 11 NSF CAREER award recipients. UIC has an advanced networking infrastructure in place for data-intensive scientific research that is well-connected regionally, nationally and internationally.

Chicago epitomizes the modern, livable, vibrant city. Its airport is the second busiest in the world, with frequent non-stop flights to virtually anywhere. Yet the cost of living, whether in an 88th floor condominium downtown or on a tree-lined street in one of the nation's finest school districts, is surprisingly low.

Applications must be submitted at https://jobs.uic.edu/ for the Data Science/Human-Computer Interaction search. Please include a curriculum vitae, teaching and research statements, and names and addresses of at least three references in the online application. Applicants needing additional information may contact the Faculty Search Chair at DS_HCI-search@cs.uic.edu. For fullest consideration, apply by January 4, 2016, but applications will be accepted until the positions are filled. The University of Illinois is an Equal Opportunity, Affirmative Action employer. Minorities, women, veterans and individuals with disabilities are encouraged to apply.

University of Maryland Baltimore County (UMBC)

An Honors University in Maryland

Information Systems Department

Tenure Track position in Artificial Intelligence/Knowledge Management

The Information Systems (IS) Department at UMBC invites applications for a **tenure-track** faculty position at the Assistant Professor level starting August 2016. *We are searching for a candidate with research interests and experience in Artificial Intelligence (AI) and/or knowledge management (KM).* The ideal candidate should have expertise in conducting AI/KM research to improve decision making in application domains closely relevant to one or more active research areas in the IS department, preferably health IT, social computing, and smart environments.

The research areas of current faculty in the Department of Information Systems include Artificial Intelligence/Knowledge Management, Databases and Data Mining, Human Centered Computing, Software Engineering, and Health Information Technology. Candidates must have earned a PhD in Information Systems or a related field no later than August 2016. Candidates should be engaged in research that fosters collaboration with at least one of these areas. Preference will be given to those who can collaborate with current faculty. Candidates should have a strong potential for excellence in research, the ability to develop and sustain an externally funded research program, and the ability to contribute to our mission of excellent graduate and undergraduate education. Exceptionally qualified candidates in an area other than AI/KM that aligns well with the department's mission may be considered.

The Department offers undergraduate degrees in Information Systems and Business Technology Administration. Graduate degree programs, MS and PhD, are offered in both Information Systems and Human-Centered Computing, including an innovative online MS in IS program. Consistent with the UMBC vision, the Department has excellent teaching facilities, state-of-the-art laboratories, and outstanding technical support. UMBC's Technology Center, Research Park, and Center for Entrepreneurship are major indicators of active research and outreach. Further details on our research, academic programs, and faculty can be found at http://www.is.umbc.edu/. Members of underrepresented groups, including women and minorities, are especially encouraged to apply.

Applications will not be reviewed until the following materials are received: a cover letter, a one-page statement of teaching interests, a one to two-page statement of research interests, and a CV. Electronic copies should be submitted to the IS department through the following link: http://apply.interfolio.com/31466. For inquiries, please contact Barbara Morris (410) 455-3795 or bmorris@umbc.edu or Dr. Dongsong Zhang (410) 455-2851 or zhangd@umbc.edu. Review of applications will begin immediately and will continue until the position is filled. These positions are subject to the availability of funds.

UMBC is an Affirmative Action/Equal Opportunity Employer and welcomes applications from minorities, women, veterans and individuals with disabilities.

University of Maryland

Maryland Cybersecurity Center

Senior Faculty Search

The University of Maryland is seeking candidates for a senior faculty position in the Maryland Cybersecurity Center (http://www.cyber.umd.edu). Founded in 2010, the Center has vibrant research and educational components involving over 25 faculty members from several academic departments across campus, including five hired in the Department of Computer Science and the Department of Electrical and Computer Engineering over the past three years. The University of Maryland is also home to ACES, the nation's first undergraduate honors program dedicated to cybersecurity. Candidates for the senior position are expected to be prominent researchers in the field of cybersecurity, defined broadly, with a strong publication and funding record. The new hire will have a joint appointment in the Department of Computer Science and the Department of Electrical and Computer Engineering, as well as in the University of Maryland Institute for Advanced Computer Studies. The opening is for an appointment at the level of (tenured) Full Professor or (tenured) Associate Professor, depending on the level of experience. The appointment can be effective as early as July 1, 2016.

Applications from women and other under-represented groups are especially welcome.

Please apply online at https://ejobs.umd.edu (position number 121592). Applicants are strongly encouraged to submit their applications by January 31, 2016.

The University of Maryland is an affirmative action, equal opportunity employer. Applications completed by January 31, 2016 will receive full consideration. The position will remain open until filled.

University of South Florida
Department of Computer Science and Engineering
Department Chair

The Department of Computer Science and Engineering is seeking an outstanding scholar with credentials commensurate with appointment to full professor to chair the Department. The candidate is expected to provide a vision and to work with the faculty to lead the Department to international prominence. The Chair reports to the Dean of the College of Engineering and will lead efforts to achieve the strategic goals of the Department as well as the College. These goals include increasing the number of PhD graduates, and developing cross-disciplinary, multi-institution, and industry relationships for large team research efforts. The ideal candidate has an outstanding record of both scientific accomplishments and administrative leadership, ability to recruit, mentor and retain diverse first-rate faculty, and ability to maintain a collegial and ethical environment.

The Department of Computer Science and Engineering (http://www.usf.edu/engineering/cse/) is on an upward path in increasing faculty and instructional staff, student enrollment and external research funding. It currently has 24 research-focused tenure-track faculty and seven full-time instructors. Among the faculty are Distinguished University Professors, ACM, IEEE, and IAPR fellows, and nine NSF CAREER awardees. Research strengths cross areas including computer vision, cybersecurity, distributed systems, VLSI Design and CAD. The Department offers three BS and MS degrees (in Computer Science, Computer Engineering, and Information Technology) and a PhD in Computer Science and Engineering. The total enrollment is approximately 650 undergraduate students, 120 MS students, and 77 PhD students. Enrollment is growing in all programs at this time.

The University of South Florida is a high-impact, global research university dedicated to student success. USF is a Top 50 research university among both public and private institutions nationwide in total research expenditures, according to the National Science Foundation. Serving nearly 48,000 students, the USF System has an annual budget of $1.5 billion and an annual economic impact of $4.4 billion. USF is a member of the American Athletic Conference.

An application package should include a cover letter, curriculum vitae, statements describing research and administrative experience, and the names and contact information of professional references. Application materials are to be submitted online. See http://www.usf.edu/engineering/cse/about/chair-search.aspx for instructions. Send email to chair-search@cse.usf.edu for questions. Applications will be considered starting immediately until the position is filled.

The University of South Florida is an Equal Opportunity/Equal Access/Affirmative Action Institution. Women and minorities are strongly encouraged to apply. Dual career couples with questions about opportunities are encouraged to contact the Chair of the search committee.

University of South Florida
Computer Science, Computer Engineering, and Information Technology
Instructor Position

Applications are invited for an open Instructor position in the Department of Computer Science and Engineering. We are seeking an instructor who can teach a broad range of Computer Science, Computer Engineering, and/or Information Technology core and elective courses at the undergraduate and graduate level. The University offers a promotion path for instructors. Salary will be commensurate with qualifications and experience. Candidates must have completed, or be near completion of, a Ph.D. degree in computer science, computer engineering, information technology, or a related discipline. Successful candidates are expected to start fall 2016.

The Department of Computer Science and Engineering at USF is on an upward path in increasing faculty and instructional staff, student enrollment, and external research funding. It currently has 24 research-focused tenure-track faculty and seven full-time instructors. Among the faculty are Distinguished University Professors, ACM, IEEE, and IAPR fellows, and nine NSF CAREER awardees. The Department offers three BS and MS degrees (in Computer Science, Computer Engineering, and Information Technology) and a PhD in Computer Science and Engineering. The total enrollment is approximately 650 undergraduate students, 120 MS students, and 77 PhD students. Enrollment is growing in all programs at this time.

For further information and for application instructions, please see our faculty search website: http://www.usf.edu/engineering/cse/about/faculty-search.aspx. For questions please send email to faculty-search@cse.usf.edu. Applications will be considered starting immediately until the positions are filled.

The University of South Florida is an Equal Opportunity / Equal Access / Affirmative Action Institution. Women and minorities are strongly encouraged to apply.

The University of Texas at San Antonio
Department of Computer Science
Faculty Positions in Computer Science

The Department of Computer Science at The University of Texas at San Antonio invites applications for a tenured/tenure-track position at the assistant or associate professor level, starting Fall 2016. Interested candidates with research focus in one or more areas of system software, data science, high performance computing, or cloud computing are encouraged to apply.

See http://www.cs.utsa.edu/fsearch for information on the Department and application instructions. Screening of applications will begin immediately. Full consideration will be given to applications received by January 4, 2016, and the search will continue until the positions are filled or the search is closed. The University of Texas at San Antonio is an Affirmative Action/Equal Opportunity Employer.

Department of Computer Science
RE: Faculty Search
The University of Texas at San Antonio
One UTSA Circle
San Antonio, TX 78249-0667
Phone: 210-458-4436

University of Wisconsin – Stevens Point
Department of Computing and New Media Technologies
Assistant/Associate Professor

The Department of Computing and New Media Technologies at UW-Stevens Point invites applications for a tenure-line faculty position, starting August 2016. We are particularly interested in candidates with skills in:
- Application development
- Web/mobile technologies
- Data science and management
- High performance computing

Required Masters or Ph.D. preferred in computer science or related field. There may be opportunity to teach graduate courses in an online Master's program. See http://www.uwsp.edu/hr/jobs/Pages/default.aspx for application instructions and additional information on our department.

Screening of applicants will begin on January 31, 2016 and continue until the position is filled. UW-Stevens Point is an Affirmative Action/Equal Opportunity Employer.

Washington State University
Data Science TT Faculty

Data Science faculty - WSU School of Electrical Engineering & Computer Science is hiring a full-time tenure-track faculty, all ranks, located in Pullman, WA. For more information and to apply, please visit https://www.wsujobs.com/postings/22472. WSU is an EO/AA Educator and Employer.

Washington State University
Machine Learning TT Faculty

Machine Learning faculty - WSU School of Electrical Engineering & Computer Science is hiring a full-time tenure-track faculty, all ranks, located in Pullman, WA. For more information and to apply, please visit https://www.wsujobs.com/postings/22397. WSU is an EO/AA Educator and Employer.

DOI:10.1145/2851212

Dennis Shasha

Upstart Puzzles
Ice Trap

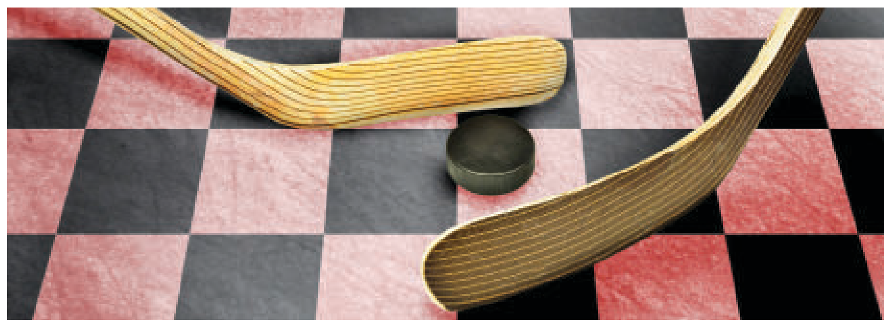


IMAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

IMAGINE A 500-SQUARE-BY-500-SQUARE red/black ice checkerboard with walls along all four edges. A frictionless hockey puck is moving diagonally above the checkerboard in a northwest, northeast, southeast, or southwest direction (you do not know which or where the puck started) at a speed of one diagonal square in each time unit. And you want to trap it in a one-square-by-one-square location.

You can put up horizontal (east-west) or vertical (north-south) walls of any length across the checkerboard, but your total wall length is limited to some total *T*. In order to build your first wall of length *L*, you must wait at least ceiling ($L/10$) time units from the start of the game. To build any subsequent wall of length L', you must wait at least ceiling ($L'/10$) time units from the time you built your last wall. Once you are allowed to put up a wall, it appears instantly. You may likewise tear down your built walls instantly at any time. If, as you attempt to put up a wall, the puck is in one of the squares the wall covers, then the wall will not be built (and you will be told the wall would have hit a puck). You can build a new wall after $L/10$ further time units.

If the puck hits one of your walls or a fixed wall, it will ricochet to the reflecting diagonal; for example, if the puck moves southeast and hits your north-south wall, it will ricochet in a southwest direction one square south of the square from where it hit the wall, while also changing the color square on the checkerboard from black to red or red to black.

We consider three detection scenarios:

Which wall, side, where, hit. If the puck hits any one of the walls you put up, you know which wall was hit, on which side, and where. But you do not learn when the puck hits the permanent side walls.

Which wall hit. If the puck hits any one of your walls, you know which wall was hit but not where the puck hit.

Know a wall was hit. If the puck hits any one of your walls, you know only that some wall has been hit but not which one.

For each of these scenarios, your goal is to confine the puck to a region of area one square by one square in as little time as possible (worst case), no matter where the puck began or which direction it was going initially.

How would you guarantee to confine the puck in a one-by-one square given only a length of 502 squares in wall length in the most difficult scenario (the second one)?

Solution. The following solution sketch is due to NYU freshman Dan Simon. Let us fix the southwest checker square to be (0,0), where the first coordinate represents the north-south direction. After waiting 50 time units, put up a north-south wall from (0,1) to (499,1). The puck will eventually hit that wall. If it hits soon after again, then you have trapped the puck in the narrow alleyway between (0,0) and (499,0). Otherwise the puck is traveling east, so 50 units later, you can tear down that wall and put up another from (0,53) to (499,53). That wall will reflect the puck back. You can then tear down that wall, and 52 time units later, you can put up a new wall from (0,1) to (499,1).

You have now trapped the puck in an alleyway of one-unit thickness. Now put a single barrier at (250,0). The puck will hit it in at most 500 time units. Now put up a barrier at (252,0). If the puck hits it, you have trapped the puck. Otherwise, put up a barrier at (246, 0) and at (248,0) after the puck hits.

Upstart 1. The minimum conceivable wall length that could be sufficient is three checkerboard squares if you could somehow force the puck into a corner with the fixed walls. Can you indeed trap the puck that way? If not, what is the minimum length you need to trap the puck and how?

Upstart 2. Find optimal worst-case guarantees in all three detection scenarios for the minimum wall length.

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html

Dennis Shasha (dennisshasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, as well as the chronicler of his good friend the omniheurist Dr. Ecco.