COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

07/2014 VOL.57 NO.07

Your Phone as Quake Detector

Big Data and Its Technical Challenges

Who Must You Trust?

Licensing Professional Software Engineers

Robots Compete in Disaster Scenarios

Responsible Programming

Association for Computing Machinery acm

November 16-19
Dresden, Germany

ITS 2014

Interactive Tabletops and Surfaces

ITS.2014

@ITS_2014

We invite you to the

2014 ACM International Conference on Interactive Tabletops and Surfaces
in Dresden, Germany, where

Baroque Beauty meets Interactive Surfaces

Raimund Dachselt & Nicholas Graham
(General Chairs)

REGISTER NOW!

its2014.org

Upcoming Submission Deadlines:
Demos & Doctoral Symposium: **July 21, 2014**
Posters: **August 29, 2014**

Photo: Andreas Siegel | photocase.com

Association for Computing Machinery

SIGCHI
special interest group computer human interaction

COMMUNICATIONS OF THE ACM

Departments

Last Byte

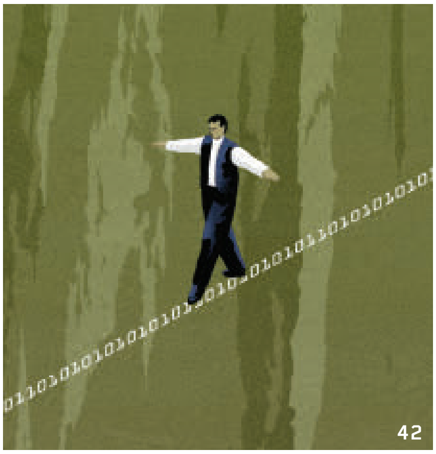
News

Viewpoints

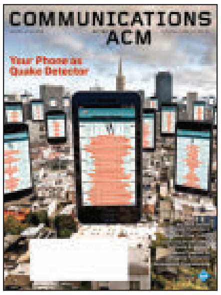
PHOTOGRAPH BY PETER CHUPAS, COURTESY OF ARGONNE NATIONAL LABORATORY

07/2014
VOL. 57 NO. 07

Practice

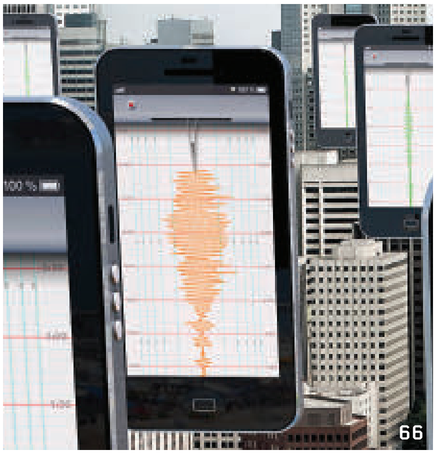


Articles' development led by acmqueue
queue.acm.org

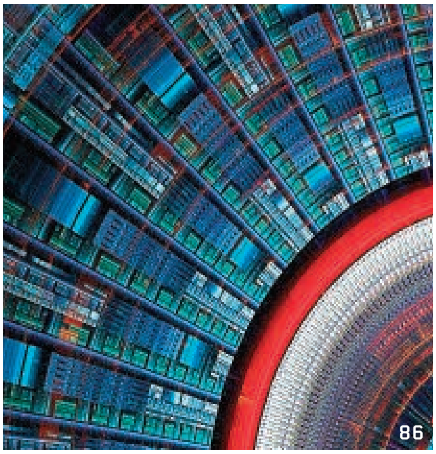


About the Cover:
A fascinating project conducted at Caltech harnesses sensor data from thousands of smartphones for real-time awareness of and response to threatening earthquakes. Authors trace the development and deployment of Caltech's Community Seismic Network beginning on page 66. Cover illustration by hitandrun.

Contributed Articles



Review Articles



Research Highlights

Association for Computing Machinery
Advancing Computing as a Science & Profession

ILLUSTRATIONS BY MICHAEL GLENWOOD, HITANDRUN, AMAZEME GRAPHICS

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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

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

Executive Director and CEO
John White
Deputy Executive Director and COO
Patricia Ryan
Director, Office of Information Systems
Wayne Graves
Director, Office of Financial Services
Russell Harris
Director, Office of SIG Services
Donna Cappo
Director, Office of Publications
Bernard Rous
Director, Office of Group Publishing
Scott E. Delman

ACM COUNCIL
President
Alexander L. Wolf
Vice-President
Vicki L. Hanson
Secretary/Treasurer
Erik Altman
Past President
Vinton G. Cerf
Chair, SGB Board
Patrick Manning
Co-Chairs, Publications Board
Jack Davidson and Joseph Konstan
Members-at-Large
Eric Allman; Ricardo Baeza-Yates; Cherri Pancake; Radia Perlman; Mary Lou Soffa; Eugene Spafford; Per Stenström
SGB Council Representatives
Paul Beame; Barbara Boucher Owens; Andrew Sears

BOARD CHAIRS
Education Board
Andrew McGettrick
Practitioners Board
Stephen Bourne

REGIONAL COUNCIL CHAIRS
ACM Europe Council
Fabrizio Gagliardi
ACM India Council
Srinivas Padmanabhuni
ACM China Council
Jiaguang Sun

PUBLICATIONS BOARD
Co-Chairs
Jack Davidson; Joseph Konstan
Board Members
Ronald F. Boisvert; Marie-Paule Cani; Nikil Dutt; Roch Guerrin; Carol Hutchins; Patrick Madden; Catherine McGeoch; M. Tamer Ozsu; Mary Lou Soffa

ACM U.S. Public Policy Office
Cameron Wilson, Director
1828 L Street, N.W., Suite 800
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
Chris Stephenson, Executive Director

STAFF

DIRECTOR OF GROUP PUBLISHING
Scott E. Delman
publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Larry Fisher
Web Editor
David Roman
Editorial Assistant
Zarina Strakhan
Rights and Permissions
Deborah Cotton

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Designer
Iwona Usakiewicz
Production Manager
Lynn D'Addesio
Director of Media Sales
Jennifer Ruzicka
Public Relations Coordinator
Virginia Gold
Publications Assistant
Emily Williams

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

CONTACT POINTS
Copyright permission
permissions@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/guidelines

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:10.1145/2631183

Vicki Hanson, Reyyan Ayfer, and Bev Bachmayer

European Women in Computing

For several years, ACM-W has been an active advocate for women in computing. With ACM's increasing international membership, regional Councils have been created and groups within them are taking up this same mission. Two years ago, the ACM Europe Council formed ACM-WE to sponsor European initiatives for women in computing.

Europe, of course, is by no means homogenous with respect to the involvement of women in computing, although many commonalities exist. A goal of ACM-WE is to promote an image of computing that would be more attractive to women. In addition, we seek to provide information about the many different kinds of careers available to women to help them envision a future in computing.

Inspired by events such as the Grace Hopper and ACM-W celebrations in the U.S., one of the first activities of ACM-WE was to create a European conference that would encourage women in their computing careers. The first ACM-WE *womENcourage* conference was held last March in Manchester, England. Attracting more than 200 attendees (both male and female) from 28 countries, this event sought to highlight technical accomplishments of women in the field as well as to provide opportunities for young women to meet others and develop professionally. Technical keynote presentations from women in computing as well as posters from young women currently studying computing furthered these goals.

A special emphasis was placed on providing networking and career opportunities for the next generation of computing professionals. Through funding from industry supporters and ACM-W, 54 computing students from 26 countries received travel stipends that allowed them to participate in the conference. A conference highlight for these participants was the career fair, featuring our seven industrial supporters (Google, Intel, Facebook, Bloomberg, Microsoft Research, Yahoo! Labs, and Cisco). An "unconference" feature also allowed attendees to cluster into facilitated discussion groups around topics selected that day by the participants themselves.

The second *womENcourage* conference will be held in Sweden, Sept. 24–26, 2015, hosted by Uppsala University. We invite anyone who wishes to help organize the conference to contact us at acm-w-europe@acm.org. Volunteers are welcome to help both with technical aspects (program committee positions) as well as operational aspects (such as career fair organization and networking events arrangements).

In addition to the womENcourage conferences, we are establishing *Friends of ACM-WE* to bring together industry, universities, and non-government organizations to help increase the visibility of women in computing. We will be looking to these organizations to donate time, expertise, and experience to help realize our mandate of the full engagement of women in computing. ACM-WE also is working to establish a Distinguished Speaker series featuring senior women in computing. Technical talks and informal group discussions will help students see and be inspired by women who have established successful careers.

A goal of ACM-WE is to promote an image of computing that would be more attractive to women.

ACM-WE supports not only early career professionals but also women in senior levels of computing. As part of this endeavor, we are working to establish a pipeline of European women who will serve in leadership positions within ACM and also become recognized for their computing achievements through ACM Senior Member, Distinguished Member, and ACM Fellow status. We encourage all European computing professionals to become members of ACM to be eligible for these honors.

With the support of the ACM Europe Council, and in collaboration with ACM-W on best practices, ACM-WE is actively working to encourage women to pursue opportunities in computing, providing them with role models and mentors, and working to increase the prominence of technical women in leadership roles. Join us for a better future!

Vicki Hanson currently serves as the ACM vice president and is on the executive committee of ACM-WE. She is a professor in the School of Computing at the University of Dundee in Scotland.

Reyyan Ayfer is chair of ACM-WE and is chair of the Department of Computer Technology and Programming at Bilkent University in Ankara, Turkey.

Bev Bachmayer is vice chair of ACM-WE and is employed by Intel GmbH in Munich, Germany.

SIGCOMM
CHICAGO 2014

DOI:10.1145/2631185

Vinton G. Cerf

Responsible Programming

WELCOME TO "CERF'S UP!" I am grateful for Editor-in-Chief Moshe Vardi's invitation to continue writing for *Communications*; this column succeeds the "From the President" column I penned during my service to ACM in that role.

Let me congratulate Alex Wolf, the newly elected ACM president. I know he will give exemplary service to our organization. Congratulations also go to Vicki Hanson and Erik Altman in their new roles as vice president and secretary/treasurer respectively. I know this team will provide first-rate leadership.

I also thank Alain Chenais, who ends his term as Past President and I begin mine. He has been a staunch, reliable, and active leader in ACM matters and I expect this will continue. There are many others elected to new positions or moving on as their terms in office end. I thank them all without enumeration, and commend them to your attention.

Lastly, allow me to note the enormous contributions of the ACM staff and, especially, the leadership of John White, CEO, and Pat Ryan, COO of ACM. They have accumulated a truly enviable record of steadfast leadership spanning the terms of many elected ACM officers.

Now to the substance of this column: responsible programming. What do I mean by that? In a nutshell, I think it means people who write software should have a clear sense of responsibility for its reliable operation and resistance to compromise and error. We do not seem to know how to write software that has no bugs…at least, not yet. But that, in a sense, is the very subject I want to explore.

My very good friend, Steve Crocker, drew me into a conversation about this topic a short while ago. As a graduate student, he had pursued a dissertation on provable correctness of programs. While this is not a new topic, the objective continues to elude us. We have developed related tactics for trying to minimize errors. Model checking is one good example of a systematic effort to improve reliability for which ACM gave the Turing Award in 2007 to Edmund Clarke, Allen Emerson, and Joseph Sifakis. What is apparent, and emphasized by Crocker, is the tools available to programmers for validating assertions about program operation are complex, with user interfaces only a mother could love (my characterization). Formal proofs are difficult, especially for anything but the simplest sort of program. Just conceiving the appropriate conditional statements to characterize program correctness is a challenge.

Despite the Turing Halting Problem, it is still possible to establish lines of reasoning to show a particular program terminates or achieves a repeatable state under the right conditions. One can make other kinds of statements about I/O checking (for example, buffer overflows). Some unending programs, like email user agents, can still have characterizations of well-defined states. It is clear, however, it is not easy to develop succinct and potentially demonstrable statements about program behavior that show the likelihood the program will behave as desired. Yet harder may be demonstrating the program does not do something undesired.

While I have no ready solution to the problem, I believe better interactive tools are needed to test assertions about the program's anticipated behavior while it is being written and to get some useful feedback from the composition and validation system that these assertions are somehow supportable. If not provable, then at least not disproved by counterexample perhaps. It seems fair to imagine that when a programmer is designing a program and actually writing the code, there is a model in the programmer's head of what the program is supposed to be doing and, presumably things it is not supposed to do or should avoid. Whether this model is sufficiently clear and complete to allow provable or verifiable assertions to be made could be the subject of considerable debate.

One intriguing example of programming environments that is tangentially relevant comes from Bret Victor (http://worrydream.com) who has conceived and implemented a programming environment that allows one to see immediately the results of executing the current program. Obviously, the system can only do this when the programmer has reached a point where the program can be parsed and interpreted. Imagine an environment fashioned for continuous validation of a set of assertions, as the program is developed. One suspects heavy use of libraries could either help or hinder the process of verifying program correctness. If the library of subroutines is opaque to the verifying tools, bugs could be hidden. However, if a set of assertions that are invariant for the subroutine could be codified, the use of such a library might actually help the validation process. I am fairly certain a body of prior work exists that can be cited here, but my impression is such tools are not used regularly by professional programmers today.

It seems timely to suggest responsible programming calls for renewed efforts to verify proper operation of software many may depend upon heavily to work as advertised. To do this, we need much better tools and programming environments than seem to be available today. I await with great interest responses from ACM members more knowledgeable than I in this area. **C**

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

DOI:10.1145/2622628.2622629

Snowden Weak Link: Copying to USB Device

I wish reality were as simple as Bob Toxen made it out to be in his article "The NSA and Snowden: Securing the All-Seeing Eye" (May 2014) where he said, "A simple one-minute scan on the way out by a handheld metal detector—'wanding,' as used by the Transportation Security Administration and at courthouses—would have found any flash memory device." However, flash devices have shrunk to minuscule size, even as their capacity has increased dramatically. Consider the micro-SD flash storage device in a typical smartphone; it can store more than 32GB and be small enough to be hidden practically anywhere. Moreover, its small mass makes detection especially difficult for a typical handheld metal detector. A spy could even attach one with chewing gum to a tooth, defeating practically any routine check.

So the real problem in the case of Edward J. Snowden is not that Snowden carried a flash memory device in and out of National Security Agency facilities but that he was able to transfer sensitive data to the device in the first place.

In most secure environments, it is extremely difficult, if not impossible, to attach an external device to a secure system. If it could be done, the system would no longer be secure, as the device would be able to transfer malware to, as well as steal data from, the secure system.

In 2008, an infected USB flash drive was famously connected to a military laptop. The malicious code uploaded itself to a secure network under the control of U.S. Central Command. This incident should have alerted the NSA to the dangers inherent in the use of removable memory devices. Moreover, the Stuxnet affair, two years later, demonstrated that U.S. security services were clearly aware that removable memory devices are potential attack vectors. The NSA should have anticipated these risks and taken necessary measures well in advance of Snowden's leaks.

The reason for the apparent indifference to such risks is that insider attacks are particularly difficult to address. The esprit-de-corps culture prevalent in the NSA made it essentially unthinkable that one in their midst could betray the organization, and is why Snowden was able, apparently, to convince coworkers to grant him additional access.

Security is an overhead; by controlling access, security makes it inherently difficult for people to carry out their work, so a compromise between utility and security must be established. In the Snowden case, though the compromise went too far toward utility, it would be a mistake to go to the other extreme by imposing security procedures that impede the NSA's useful work.

Vassilis Prevelakis, Braunschweig, Germany

Author's Response:

Wanding would have caught a USB memory stick due to the metal in its plug. No security ring is perfect. Defeating the rings involving encryption, physical access to systems, and software limiting the number of documents one may access would be extremely difficult. I demonstrated that stopping even system administrator insider attacks can be done reasonably easily. The reason Prevelakis claimed for NSA "indifference" is unsubstantiated. Aldrich Ames, Robert Hanssen, and other convicted American traitors should have convinced the NSA (and the CIA) to avoid unlimited trust. (I do not consider Snowden a traitor, as he was alerting Americans to the apparently unconstitutional and illegal actions of the government.)

Bob Toxen, Duluth, GA

The esprit-de-corps culture prevalent in the NSA made it essentially unthinkable that one in their midst could betray the organization.

I was disturbed by the cover headline—"The NSA and Snowden: How better security measures could have stopped the leak"—publicizing Bob Toxen's article (May 2014) for implying that Snowden simply produced "leaks" that should have been "stopped." Moreover, I found it odd that the article focused on how the NSA's poor security allowed these leaks to take place. It would have been more appropriate to acknowledge the alternative interpretation, that Snowden's revelations brought to light abhorrent violations of privacy on the part of the U.S. and U.K. governments. After all, the constitutionality of the NSA's spying was critiqued in the article's sidebar. Why not follow through to address the apparent contradiction between "good security practices" and the supposed "transparency" of agencies with the power to tap all our communications (including this one)?

William Gaver, London U.K.

Objects Are Patterns, Not Language

To those who care about the writing of software, should object-oriented language constructs recede in today's shattered-object world? Back in the early days of design patterns, the question arose as to why languages do not support pattern-oriented constructs. The answer came back from the Gang of Four of design-patterns fame that languages should not be pattern-oriented, since modern languages provide tools to create patterns effectively without pattern-specific features.

Maybe objects are more for our heads than for our languages when it comes to modern systems.

Instead of this purist point of view, modern languages do provide features that support common patterns, with iterators as an important example. So pattern support exists, at least partially, but there is no reason to think any language will have a goal of being pattern-oriented, as it is largely true that the other constructs of modern languages express the nature of patterns well without additional notation. So patterns live comfortably in the minds of software developers largely without specific notation to support them.

Not so with objects, for which most popular modern languages devote a vast amount of notation and through which many modern APIs are structured. Moreover, what programmer, when introduced to object-oriented design, does not enjoy a simple taxonomy of an animal farm as a clear approach to structuring problems in a step toward solving them? But our modern animal farm is more George Orwell than Old MacDonald, as once-comfortable objects fight to escape the boundaries being placed on them.

Object-oriented design was born in a world of one place and one time; the cow lived on a farm, the farm lived on a server, and the cow was alive for the duration of the time we had tasks to ask of her: be born, give milk and calves, and then provide steaks and burgers during systematic deconstruction. The cow lived in our minds and in our code as a structured entity with which we could converse.

But objects got big and devices got small and the world moved on. Servers today live on the farm and the cows (objects) no longer live at all, certainly not like they did; but just as shards of the cow—a little on a phone here as JavaScript to render a moo, a little on a server there as SQL to keep track of her progeny. A nice object-oriented library is likely available to manage talking to the database or updating the user interface to render a comforting sound. But this is the story of the blind men and the elephant now told in code; "the object" is described only as a collection of amalgamated ideas in disparate languages on systems at different times. A single object-oriented language has no purpose to serve here, because the unified description of these objects is bigger than the purpose of programming languages.

For small objects, object-oriented programming can still be useful, but our problems have outgrown our systems, and object-oriented languages want objects to be inside the systems instead of the systems being inside our objects. We no longer have cows, at least in terms of whole cows in one place at one time. So, are cow parts best described as objects or are these object shards too disconnected and independently represented to be viewed as objects?

REpresentational State Transfer, or REST, is not object-oriented but does allow interaction with the conceptual objects defined by interacting systems. Maybe objects are more for our heads than for our languages when it comes to modern systems. For a related perspective on objects, see Mordechai Ben-Ari's Viewpoint "Objects Never? Well, Hardly Ever!" (Sept. 2010), and for more on REST and related technologies, see Ian Foster et al.'s article "How Do I Model State?: Let Me Count the Ways" (Sept. 2008).

Warren MacEvoy, Grand Junction, CO

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.

Coming Next Month in COMMUNICATIONS

Efficient Maximum Flow Algorithms

Example-based Learning in Computer-Aided STEM Education

Reshaping Terrorist Networks

Researching the Robot Revolution

Slow Search

Quantifying Eventual Consistency with PBS

Quality Software Costs Money— Heartbleed Was Free

Bringing Arbitrary Compute to Authoritative Data

Undergraduate Software Engineering: Addressing the Needs of Professional Software Development

And the latest news about the power of obfuscation, surgical robots, and privacy concerns regarding facial recognition technologies.

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:10.1145/2617658 http://cacm.acm.org/blogs/blog-cacm

The Difficulty of Teaching Programming Languages, and the Benefits of Hands-on Learning

Mark Guzdial considers the "poor learnability" of programming languages, while Philip Guo enumerates some practical benefits to working in a CS lab.

Mark Guzdial "Programming Languages Are the Most Powerful, and Least Usable and Learnable User Interfaces"

http://bit.ly/1g39mAX

March 27, 2014

Andy Ko wrote a recent blog post with an important claim: "Programming languages are the least usable, but most powerful human-computer interfaces ever invented" (http://bit.ly/1iVxF3A). Andy argues the "powerful" part with points about expressiveness and political power. He uses HCI design heuristics to show how programming languages have poor usability. Obviously, some people can *use* programming languages, but too few people and with great effort.

I see Andy's argument extends to learnability. There are two ways in which programming languages have poor *learnability* today: in terms of expectancy-value and in terms of social cost.

What Is the Benefit of a Closure?

Eugene Wallingford tweeted a great quote the other day:

"Think back to before you understood closures. I'm sure you couldn't even imagine it. Now imagine them away. See, you can't do that either."

Educational psychologists measure the cognitive load (http://bit.ly/1lSmG0f) of instruction, which is the effort a student makes to learn from instruction. Every computer scientist can list a bunch of things that were really hard to learn, and maybe could not even be imagined to start, like closures, recursion in your first course, list comprehensions in Python, and the type systems in Haskell or Scala.

Expectancy-value theory (http://bit.ly/1sctSGD) describes how individuals balance out the value they expect to get from their actions. Educational psychologists talk about how that expectation motivates learning (http://edcate.co/1iefV80). Students ask themselves, "*Can* I learn this?" and "Do I *want* to learn this? Is it *worth* it?" You do not pursue a degree in music if you do not believe you have musical ability. Even if you love art history, you might not get a degree in it if you do not think it will pay off in a career. Most of us do not learn Dvorak keyboards (http://bit.ly/1jvaFNC), even though they are provably better than Qwerty, because the *perceived* costs just are not worth the *perceived* benefit. The actual costs and benefits do not really play a role here; perception drives motivation to learn.

If you cannot imagine closures, why would you want to learn them? If our programming languages have inscrutable features (i.e., high cognitive load to learn them) with indeterminate benefits, why go to the effort? That is low learnability. If students are not convinced they can learn it and they are not convinced of the value, then they do not learn it.

The Social Cost of Going in a New Direction

I was at a workshop on CS Education recently, where a learning scientist talked about a study of physicists who did their programming in Fortran-like languages and only used arrays for all their data structures. Computer scientists in the room saw this as a challenge. How do we get these physicists to learn a better language with a better design, maybe object-oriented or functional? How do we get them to use better data structures? Then one of the other learning scientists asked, "How do we *know* that our way is *better*? Consider the possibility that we're *wrong*."

We computer scientists are always happy to argue about the value of one programming paradigm over another. But if you think about it from Andy Ko's usability perspective, we need to think about it for *specific* users and uses. How do we know that we can make life *better* for these Fortran-using physicists?

What if we convinced some group of these Fortran-using physicists to move to a new language with a new paradigm? Languages don't get used in a vacuum; they get used in a community. We have now cut our target physicists off from the rest of their community. They cannot share code. They cannot use others' libraries, tools, and procedures. The costs of learning a new language (with new libraries, procedures, and tools) would likely reduce productivity enormously. Maybe productivity would be greater later. Maybe. The value is uncertain and in the future, but the cost is high and immediate.

Maybe we should focus on students entering the Fortran-using physics community, and convince them to learn the new languages. Learning scientists talk about student motivation to join a "community of practice" (http://bit.ly/1kDIhFJ). Our hypothetical physics student wants to join *that* community. They are learning to value what the community values. Trying to teach them a new language is saying: "Here, use this—it's way better than what the people you admire use." The student response is obvious: "Why should I believe *you*? How do *you* know it's *better*, if it's not what my community uses?"

Solution: Focus on Usability

Communities change, and people learn. Even Fortran-using physicists change how they do what they do. The point is that we cannot impose change from the outside, especially when value is uncertain.

The answer to improving both usability and learnability of programming languages is in another HCI dictum: *"Know thy users, for they are not you."* We improve the usability and learnability of our programming languages by working with our users, figuring out what they want to do, and help them to do it. Then the value is clear, and the communities will adopt what they see as valuable.

Philip Guo "The Benefits of Working on Research as an Undergraduate Student"

http://bit.ly/1neTOSe

April 8, 2014

As an undergraduate student studying, say, computer science, what are some of the practical benefits of working in a research lab? I can think of three off the top of my head.

Inventing the Future

At a high level, working on research is awesome because you get a chance to preview and invent the future. In classes, summer internships, and most full-time jobs you will get after graduation, you are either studying the past or doing work that will be immediately used in the present or near future. In industry, the main priorities for junior employees such as yourself are to deliver projects with near-term value in the coming week, month, or year. Only in a research lab can you prototype high-risk ideas that are five, 10, or even 20 years ahead of the state of the art in industry.

Sure, every individual researcher gets to work on only a small, specialized part of a bigger research problem. But just getting the chance to participate is an interesting opportunity. One of the main purposes of college is to expand your intellectual horizons, and hands-on experience in a research lab is a good way to do so. The broader ideas you will be exposed to in a research lab might transfer over to your future professional life in unexpected ways, even if you do not end up working in the same subfield.

Rapidly Improving Your Technical Skills

A more concrete benefit of doing research is the chance to rapidly improve your technical skills in a realistic setting outside of the classroom. I became a much better programmer and scientist throughout my three years of assisting on research projects as an undergraduate student. I had to learn new programming languages, libraries, tools, and technologies on-demand to meet project requirements. This sort of hands-on knowledge cannot easily be taught in textbooks or classes, since it requires an authentic setting where people are doing real work and not just preset exercises with known results.

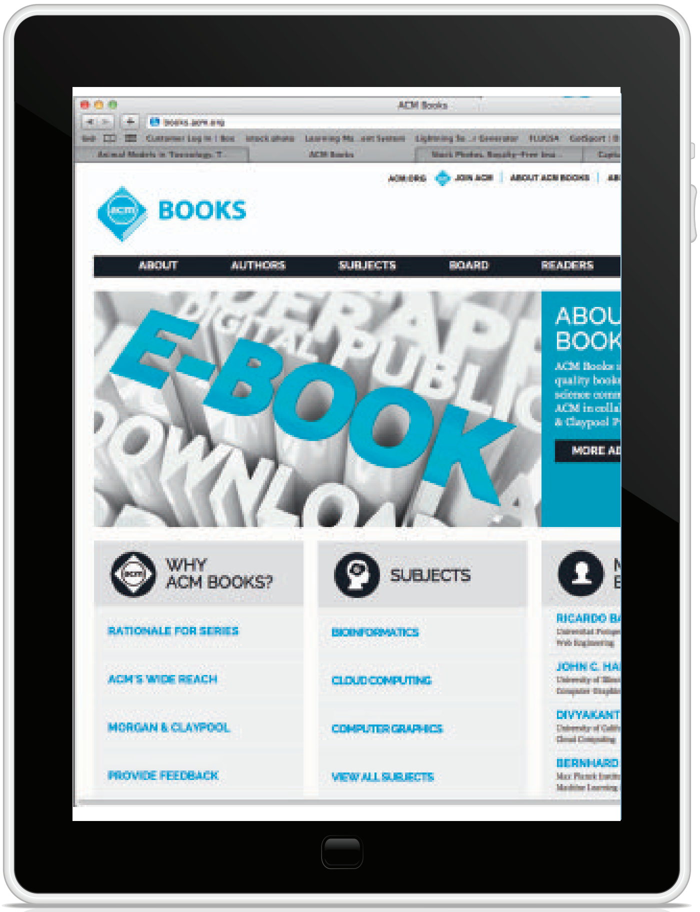
If all works out, working on research will feel like a super-intensive yet satisfying lab class where you complete an innovative project that you are proud of. You will also get to practice writing up and presenting your work to an audience, which is great training for many kinds of jobs. You might also get credited as a co-author on a published research paper, which is important if you want to pursue a Ph.D. in the future. And best of all, since you are working as an apprentice, you usually get one-on-one mentorship from a more senior researcher. This sort of personalized interaction rarely happens in even the smallest and most intimate of university classes.

Potential for Professional Advancement

One final benefit is the potential for professional advancement. If you do compelling work, then your research advisor can write recommendation letters and make personal referrals for you to get either a good job in industry or admitted into graduate school. These letters and referrals are a lot more meaningful than having a high GPA or a professor mentioning that you got an A+ in their class. My own undergraduate research advisors helped to kick-start my career in significant and often unexpected ways. In contrast, professors who taught my classes don't really remember me, since I was one of hundreds of students they saw each year in large lecture halls.

(This blog post was adapted from my undergraduate researcher recruiting article at http://bit.ly/1kFMOc9.)

Mark Guzdial is a professor at the Georgia Institute of Technology. **Philip Guo** is a postdoctoral scholar in the Massachussetts Institute of Technology Computer Science and Artificial Intelligence Laboratory.


BOOKS
ABOUT
AUTHORS
SUBJECTS
BOARD
READERS
E-BOOK
WHY ACM BOOKS?
RATIONALE FOR SERIES
ACM'S WIDE REACH
MORGAN & CLAYPOOL
PROVIDE FEEDBACK
SUBJECTS
BIOINFORMATICS
CLOUD COMPUTING
COMPUTER GRAPHICS
VIEW ALL SUBJECTS

Science | DOI:10.1145/2617660

Alex Wright

Big Data Meets Big Science

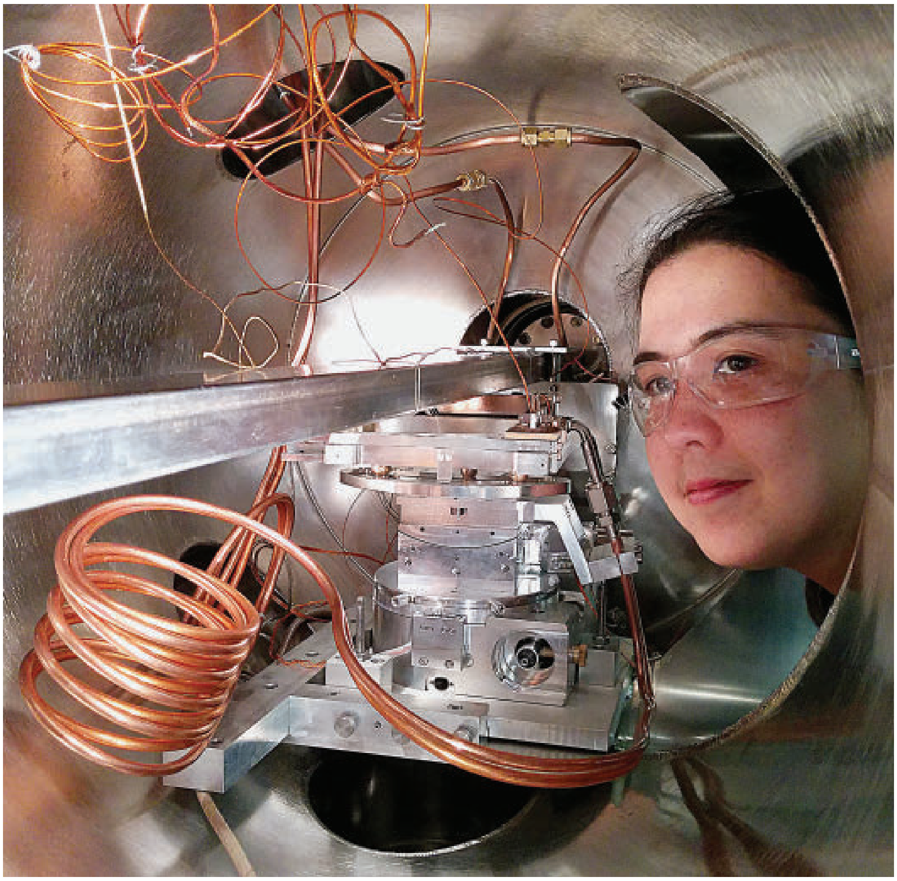
Next-generation scientific instruments are forcing researchers to question the limits of massively parallel computing.

On a secluded hilltop outside Palo Alto, CA, Jacek Becla leads a team of researchers at the SLAC National Accelerator Laboratory who are quietly building one of the world's largest databases.

Scheduled to go live in 2020, the Large Synoptic Survey Telescope (LSST) will feature a 3.2-gigapixel camera capturing ultra-high-resolution images of the sky every 15 seconds, every night, for at least 10 years. Ultimately, the system will store more than 100 petabytes (about 20 million DVDs' worth) of data, but that is barely a fraction of the data that will actually pass through the camera.

"Even though we are dealing with huge amounts of data, there is even more data that we are not saving," says Becla. With 40 billion–50 billion potential astronomical objects in the camera's purview, he explains, it would be all but impossible to store every pixel in perpetuity. Instead, the system will extract critical data from the images in real time, then simply discard the source images.

As increasingly powerful large-scale scientific instruments come online—from the Large Hadron Collider to advanced light beam processors and mo-



Argonne National Laboratory chemist Karena Chapman peers inside the vacuum tank of the new high-energy Si Laue monochromator recently installed in the Argonne Advanced Photon Source, an upgrade that increased the X-ray flux (the number of photons focused on the sample being studied) by a factor of 17.

PHOTOGRAPH BY PETER CHUPAS, COURTESY OF ARGONNE NATIONAL LABORATORY

lecular imaging tools—they are starting to churn out more data than even the most powerful massively parallel supercomputers can handle. As a result, scientists are exploring new approaches to reducing those datasets to manageable size, incorporating new learning from the private sector about cloud-based computing, and in a few cases exploring the possibilities of emerging frameworks like quantum computing.

Those strategies stand in stark contrast to the traditional scientific approach to high-performance computing, which has long relied on a "brute force" approach involving stringing together greater and greater numbers of CPUs and disk arrays. After a decades-long infatuation with parallel supercomputing, however, some researchers are beginning to butt up against the limitations of that approach.

"Moore's Law is effectively already broken down," says Massachussetts Institute of Technology professor Scott Aaronson, who argues the laws of physics have caught up with Intel founder Gordon Moore's famous dictum that the number of transistors on integrated circuits would double every two years.

Researchers also are grappling with both economic and algorithmic constraints that force them to explore methods beyond the tried-and-true technique of throwing ever more processors at a problem.

At the Argonne National Laboratory in Illinois, Chris Jacobsen leads a team working on the Advanced Photon Source (APS), an enormous, football field-sized synchrotron that produces X-ray photons by swirling electrons around a circular apparatus at nearly the speed of light. Researchers from 65 different field stations rely on the machine to gather imaging data about a wide range of subjects: from proteins and nanoribbons to lithium-ion batteries and catalytic converters.

The experiments vary tremendously in scope, but the data they collect always comes in intense bursts of up to 11 gigabytes of raw data per minute. In a typical month, APS distributes about 112 terabytes of data. "We get so much data that we can't just sit there and examine it by hand," says Jacobsen. "It takes time for all these processors to communicate with each other, and they can't send messages faster than the speed of light—that's sort of the ultimate limit."

"For most applications, the real bottleneck is not the processing time," but "the need to constantly retrieve stuff from memory."

Given those constraints, the team is constantly looking for more efficient ways to help researchers interpret their test results. "What are the features we can pull out from the raw data? What can we understand, rather than just measure?"

With so many far-flung researchers, Jacobsen's team has also been wrestling with how best to deliver datasets to its many end users. In the past, several teams relied on their own ad hoc "sneakernets," lugging their hard drives to the facility for a few days before bringing them back home. As the lab continues to improve its detectors, however, the data rates keep going up, forcing the Argonne team to explore new cloud-based approaches to providing data to researchers.

"We are trying to move towards a more cohesive computing strategy," says Jacobsen.

Recently, Argonne's physicists have been collaborating with their colleagues from the applied math and computer science departments to develop new tools to allow researchers to automate the transfer of data from the beamline computer to a central data store where it can be optimized, backed up, and managed. That data is then made available via a secure TCP/IP connection, using a tool called Globus Online (globus.org), and stored using Amazon Web Services—allowing for multiple parallel connections.

In a similar vein, researchers at the U.S. Department of Energy's Brookhaven National Laboratory are exploring cloud-based approaches to harnessing the vast troves of data currently being produced by the ATLAS experiment at Europe's Large Hadron Collider (LHC), famous for discovering the elusive Higgs boson.

ATLAS has already generated 140 petabytes of data, distributed between 100 different computing centers, with most of it concentrated in 10 large computing centers like CERN and Brookhaven.

Physicist Alexei Klimentov has been working on a framework for managing this enormously complex computational enterprise—which involves an estimated 3,000 physicists creating more than two million computing jobs per day—using a system called PanDA (Production and Distributed Analysis),

"PanDA is a pilot system," says Klimentov. "It knows about the site, the software, the storage, and the available CPU slots. Then, according to the available resources, it matches them against the payload." For example, a simulation project typically requires a lot of processing but little data storage, whereas a complex data analysis job requires fast access to large hard drives.

By distributing these jobs across the cloud to the most appropriate available system, PanDA can make the best use of available resources while minimizing system downtime. Even so, shuttling 140 petabytes of data around the world is no small undertaking.

Engineering fast and reliable data transfer mechanisms is emerging as one of the critical challenges for scientists working with big data—not just for moving files from computer to computer, but for shuttling data in and out of memory as well.

"In traditional high-performance computing, you have very little data and very little I/O," says Becla, "so you are basically reading the data into memory and doing the processing in memory. But in the big data world, you cannot do this. You cannot have a trillion pieces of data in memory at the same time."

MIT's Aaronson echoes that concern. "For most applications, the real bottleneck is not the processing time," he says, but "the need to constantly retrieve stuff from memory. For a lot of programs, the processor is sitting idle, waiting for the memory to come back." The challenge, then, is how to design memories that are fast, large, and responsive.

Aaronson likens the problem of classical computing to the eternal conundrum of finding an apartment in New York City: "You could get it in Manhattan where it's close to everything but small and expensive, or you go to Long Island and it's cheaper but farther from everything." Similarly, computer scientists must navigate complex trade-offs in trying to optimize system performance with large datasets. "Registries are super-scarce, then you go out to L1 and L2 cache and then the RAM, then you're in the boonies of the hard disk. How do you optimize the trade-off?"

That tension captures the challenge of cloud computing: how to take advantage of the economies of the cloud without losing the gains of having everything in close proximity?

"You can just throw more parallelism at things, but the amount of memory and the amount of disk space has been blowing up tremendously," Aaronson says. "Even if in principle you have all these parallel processes, it can be harder to write code that takes advantage of the parallelism."

The traditional approach to high-performance computing relied on millions of CPUs to perform many calculations on relatively small chunks of data. Up until recently, most large-scale systems fell into this category. Yet in the scientific world, where data is increasingly interrelated, problems are becoming tougher to parallelize.

Some researchers hold out hope for quantum computing, a much-hyped field that promises enormous computational speed gains. However, Aaronson advises caution. "There's a temptation for people to look at quantum computing and say, 'this must be the thing that will continue Moore's Law,' but a lot of that relies on misconceptions about what a quantum computer is. It's a fundamentally different kind of computer."

Unlike a classical computer that can perform a large number of calculations at the same time, quantum computers rely on subtle effects from quantum mechanics that can solve certain classes of problems much faster; for example, breaking cryptographic codes, factoring large numbers, or simulating quantum physics. For more traditional computing tasks, like combinatorial optimization, airline scheduling, or adiabatic algorithms, it is not at all clear that quantum computers will offer any meaningful performance gain.

"It's conceivable that a quantum computer could help with protein folding or DNA sequencing, but the advantages are not obvious," says Aaronson. "You'll get an advantage from a quantum computer only when you can figure out how to exploit quantum interference."

In the near term, scientific researchers may take solace in knowing they are scarcely alone in grappling with the challenges of big data. The explosive growth of the consumer Internet has thrust many of the Internet's leading companies into similar territory.

In 2007, Becla organized a 60-person workshop called the Extremely Large Database group (XLDB), which has since grown into a network of more than 1,000 members spanning numerous scientific research centers, as well as private-sector participants from Google, Amazon, eBay, LinkedIn, Yahoo!, and elsewhere.

Increasingly, these organizations find themselves operating in overlapping territories: working with large collections of images, time series, or determining how best to detect outliers in large datasets, whether in the form of gamma-ray bursts or security intrusions.

"We see commonalities between what astronomers are doing and what eBay and Wall Street are doing," says Becla.

Who would have thought that the path to unlocking the mysteries of the universe might run through eBay? Says Becla, "It's an eye-opener."

Further Reading

Francesco De Carlo Dŏga Gürsoy et al.
Scientific Data Exchange: A schema for HDF5-based storage of raw and analyzed data. Submitted to *J. Synchrotron Radiation.*

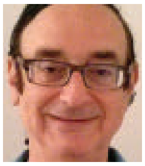
Rachel Mak, Mirna Lerotic, Holger Fleckenstein, Stefan Vogt, Stefan M. Wild, Sven Leyffer, Yefim Sheynkin, and Chris Jacobsen.
Non-negative matrix analysis for effective feature extraction in X-ray spectromicroscopy. Submitted to the Royal Society of Chemistry for a Faraday Discussion meeting. DOI: 10.1039/c000000x

Michael Stonebraker, Paul Brown, Donghui Zhang and Jacek Becla.
SciDB: A database management system for applications with complex analytics. *Computing in Science & Engineering*, 15, 54-62 (2013), DOI:http://dx.doi.org/10.1109/MCSE.2013.19

Alex Wright is a writer and information architect based in Brooklyn, NY.

ACM Member News

SECURITY NEEDS TO BE FLUID, PRAGMATIC: YUNG



"Cryptography theory is great," says Moti Yung, a Google research scientist and 2013 ACM Fellow. "There are terrific security tools and they work well, assuming no one has penetrated your computing systems." In an era of global and mobile systems, however, Yung believes businesses need more pragmatic and fresher approaches to secure systems already "partially penetrated."

Yung knows what he is talking about; he has over 30 years' experience as a cryptographer. He earned his Ph.D. from Columbia University, where he currently serves as an adjunct professor and visiting senior research scientist. Prior to joining Google in 2007, he was a cryptographer and scientist at IBM's Thomas J. Watson Research Center; chief scientist at CertCo, and director of Advanced Authentication Research at RSA Laboratories.

In 2004, Yung and Adam Young co-authored "*Malicious Cryptography: Exposing Cryptovirology*," a book focusing on cryptovirology (the use of rogue code to invade systems) and kleptography (a targeted stealth-attack mechanism using a crypto system to try to break or defeat another cryptographic system from within).

Privacy in the cloud computing era brings new challenges daily, Yung says. Ongoing and growing security threats make privacy and securing data crucial.

"It makes no sense to build a Maginot line that people can circumvent; security systems need to be fluid," Yung says. "Security professionals must thrive on challenges, accept failures as learning experiences, and move on." That is why he monitors hackers and continually updates his security research directions.

His most significant accomplishment, Yung says, is "still in the future. You can't rest on your laurels."

—Laura DiDio

Technology | DOI:10.1145/2617752

Logan Kugler

Robots Compete in Disaster Scenarios

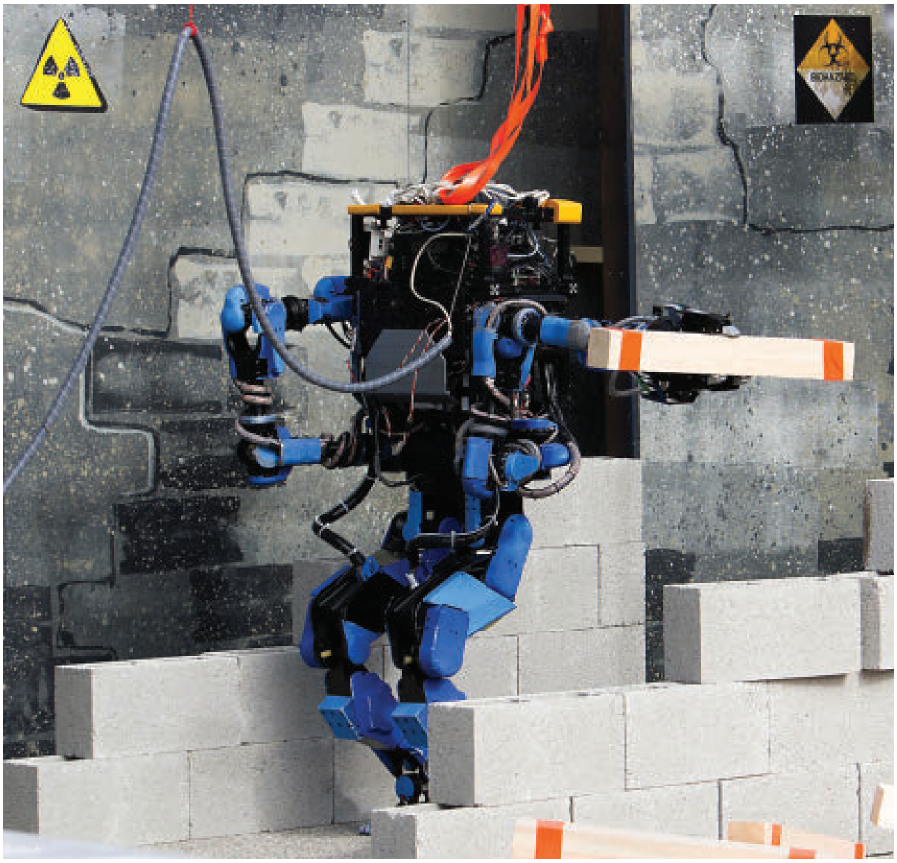
The DARPA Robotics Challenge pitted teams from around the world against each other in a series of disaster-themed tasks.

In December, 16 robots engaged in spirited competition at Florida's Miami Homestead Speedway as part of the trials of the DARPA Robotics Challenge (DRC) sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA). At stake: a chance to compete in the finals for a $2-million prize.

The robots represented 13 different universities and research labs around the U.S., plus one each from Japan, Korea, Israel, and Hong Kong (with other countries partnering). The competitors were divided into two tracks: nine teams all built their own robots and programmed them, while the other seven worked with Atlas humanoid robots from Boston Dynamics and only did the programming.

The trials were designed to be representative of tasks a robot might be called upon to perform at the scene of a natural or manmade disaster. The human operators had to direct their robots through a series of tasks and obstacles, including driving a Polaris Ranger XP 900 utility vehicle through a course, then exiting the vehicle, climbing a ladder, and closing a series of valves. The organizers of the challenge made the tasks even more difficult by degrading communications between the operators and the robots, reducing the available bandwidth and introducing latency—again, a scenario characteristic of a disaster situation.

DARPA organizers were impressed with the results. "Teams did somewhat better in the DRC trials than we expected," says Gill Pratt, program manager in DARPA's Defense Sciences Office. "For the top half of the pack, hardware performed with almost zero breakdowns. Teams also performed better under communications degradation than we expected; for the top half of the pack, robots made substantial progress, even during the minutes when communications was degraded to 100kbit/s of latency and 500ms of latency. And entrants in the top quarter of the pack earned more than half of the total possible 32 points."



Team SCHAFT's humanoid robot navigates through debris in one of the DARPA Robotics Challenge's disaster scenarios.

PHOTOGRAPH BY RAYMOND SHEH. COURTESY OF DARPA/U.S. DEPARTMENT OF DEFENSE

The team from Japan represented SCHAFT, a company founded by two researchers (Junichi Urata and Yuto Nakanishi) from the JSK Robotics Laboratory of the University of Tokyo. The SCHAFT team, using a humanoid robot of its own design, won the competition handily, amassing 27 points—7 points more than the second-place robot from the Florida Institute for Human & Machine Cognition. "The Japanese team had a long history of very good academic work from the University of Tokyo and other laboratories on which to build," says Pratt. "They also were extremely well-organized and did their hardware and software development early. Finally, they built a water-cooled electric robot with powerful actuators, which allowed their robot to generate a lot of torque without having to worry so much about overheating."

Seth Teller of the Massachusetts Institute of Technology (MIT) team says the most difficult tasks for the robots were those involving a rapidly changing environment. For example, the test that required the robot to go through a door was conducted on a gusty day, with the door swinging back and forth. The robots performed less effectively when objects in the environment didn't move "in a slow, predictable way," says Teller.

Humanoids from MIT

The MIT team, headed up by Teller and Russ Tedrake of MIT's Electrical Engineering & Computer Science department, came in fourth. The MIT team chose to work with the Boston Dynamics Atlas robot: "it was actually a pretty easy decision," says Teller. "Russ Tedrake and I are much more software than hardware people. By going with that track, we reasoned that we would effectively have one of the world's most capable hardware providers on our team." The fact that MIT and Boston Dynamics are neighbors was a factor as well—Teller cites the quick response they were able to get whenever something broke on the robot.

Teller also sees advantages to using a bipedal robot for the kinds of tasks the Challenge presented. "(The robot) can stand up and have both hands free," he says. "The world is made for people. You should have roughly human morphology to use all the tools and affordances and spaces that are out there in the world."

Teller and Tedrake's approach to programming their robot was based on assisted perception and assisted planning: the operator views the world through the robot's eyes and uses an interface, such as a touchscreen and stylus, to point out objects the robot can work on—a valve, a steering wheel, or a door handle, for example. The robot is equipped with a dictionary of objects to serve as templates, and can search through adjustable parameters to match the appropriate template to the real-world instance of an object. It also knows the degrees of freedom inherent in an object. "A valve has one degree of freedom," explains Teller. "Once the robot knows it is a valve, when we say 'turn the valve clockwise,' the motion plan it has to generate is quite simple." This approach allows the human operator to perform executive functions—deciding which door to enter, which way to turn after entering, and what to look for, for example—while the robot handles lower-level tasks of generating motion plans to perform those actions.

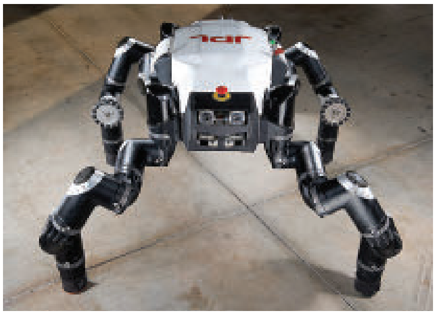
This approach provided the team a path to improving its robot's performance. "Since December, we have significantly increased the autonomy level of the robot," Teller says. Whereas during the trials, a task like picking up a drill required multiple separate commands to the robot—locate the drill, walk to the table, grab the drill—now they can include all those steps when issuing a single command like "find the drill and pick it up." That increased autonomy should help the robot deal with degraded communications, especially if the DARPA organizers introduce actual network dropouts. Such dropouts would severely impact teams that relied on "puppeteering," or real-time teleoperation, to steer and control their robots.

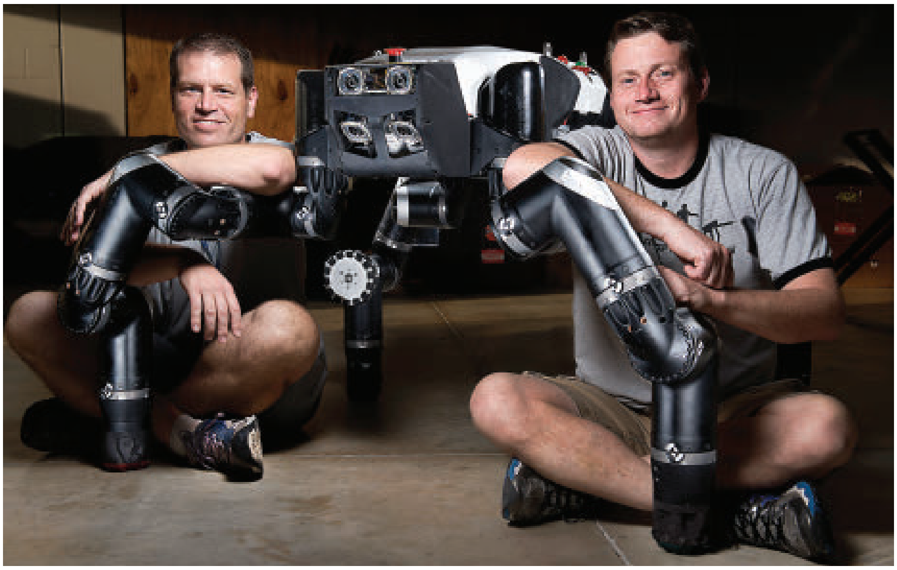
"You should have roughly human morphology to use all the tools and affordances and spaces that are out there in the world."

Simians from JPL

The ability to deal with interrupted communications is a strength of RoboSimian, the entry from the U.S. National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL). JPL built its own four-legged robot, drawing on its experience with robots. "We have a robot on Mars that needs to be very safe and very patient," says Brett Kennedy, supervisor at JPL's Robotic Vehicles and Manipulators Group. "It may only talk to its operators every now and again, and it is always going to be kind of out there on its own." That philosophy guided the design of RoboSimian as well—it has the ability to come to a stable stop and await further instructions if it runs into a problem or completes its immediate goal.

RoboSimian's stability is also appropriate for working in a disaster area. "We are always going to be a three-legged stool," explains Kennedy. "The robot can come to a stop at any time and not fall over, as opposed to a bipedal mode where you have to be actively balancing yourself. If we need to get up to something on a wall, we can move up into a bipedal position, supporting ourselves on the wall. We always have three points of contact and are statically stable because of that."





NASA JPL entry RoboSimian, alone (above) and with JPL Robotic Vehicles and Manipulators Group supervisor Brett Kennedy (left) and RoboSimian Integration lead Chuck Bergh.

PHOTOGRAPHS COURTESY OF NASA/JPL-CALTECH

The Carnegie Mellon University (CMU) entry, called CHIMP (for CMU Highly Intelligent Mobile Platform robot), also featured a quadrupedal design, "but I think we were by far the furthest over on the humanoid vs. non-humanoid spectrum," Kennedy says.

Kennedy acknowledges the advantages Teller cites in using a humanoid-style robot, particularly the ability to carry something while still moving around a site, but the JPL team found another way to accomplish the same goal. "If you look at our simian cousins, they find ways to hold things in their hands and still walk on them," Kennedy says. "We probably would have found ways to do that as well." Instead, noting that almost all of the DRC challenges provided a flat floor, RoboSimian was designed with a set of wheels in addition to its four limbs, which enabled it to roll from one place to another while using its limbs to carry something.

Assessing Results

Kennedy and Teller agree one of the main reasons the team from SCHAFT did so well was it had much lengthier experience with their robot. "They are extremely talented roboticists," says Kennedy, "there's no two ways around that, but they came out of a lab that has been building humanoids for well over a decade at this point. They've been working on a problem set that is closer to what we did for the Challenge than any other team, for longer than anybody else. That makes a big difference in terms of performance."

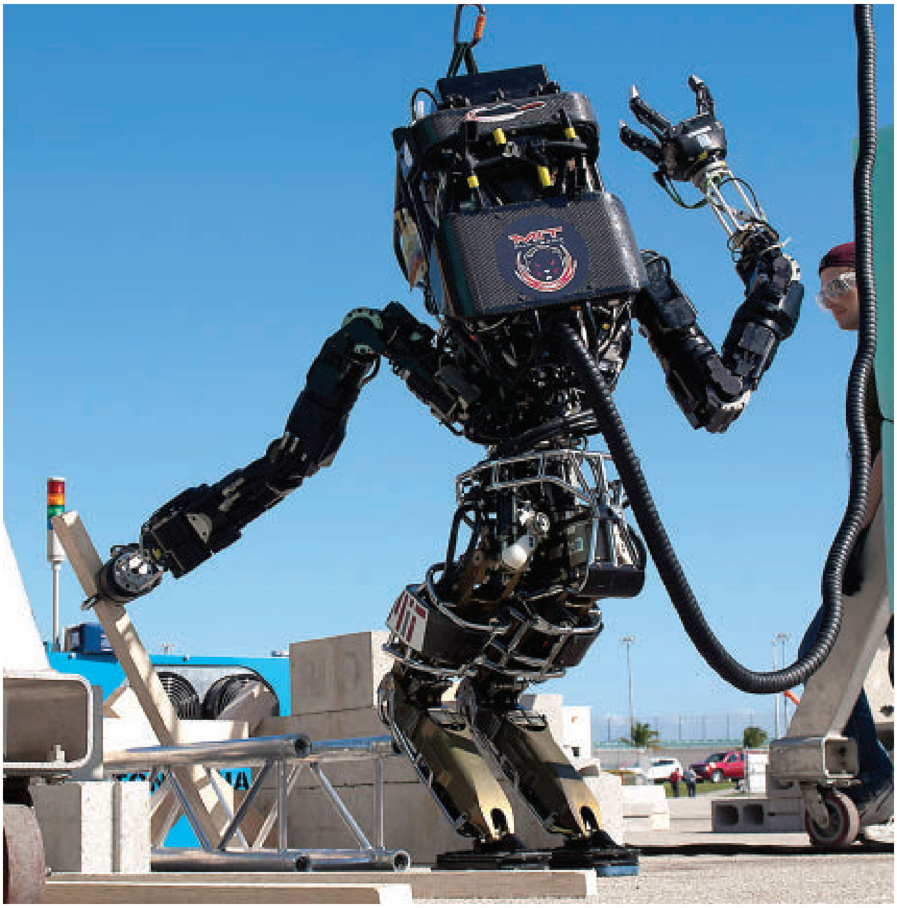
The DRC was announced in April 2012, and since SCHAFT was going to use its own system, "they had 20 months to work with their robot; we had four months," says Teller. The Boston Dynamics robot is classified as a munition by the U.S. Department of Commerce, and MIT, like most universities, will not permit munitions on campus. "So we couldn't accept delivery of the robot until August, after we resolved the bureaucracy around having it reclassified as a 'sensitive device' and not a munition."

Teller does not discount the SCHAFT team's technical excellence. "They had a nice physical platform that had powerful and very precise actuators, so they could position their effector exactly as they wanted to. They had a camera in the palm of their non-manipulating hand that they were using for a close-in view so they could see exactly what they were doing. That was something that was hard for us—something about the Atlas platform we weren't entirely satisfied with was its visibility on the task. And they had a very skilled puppeteering operator who was using these precise motions very effectively."

Still, Teller is confident about the MIT robot's abilities in the finals. "I think we're fairly well caught up at this point," he says. If communications are cut at times during the tasks, he observes, that should reduce the advantage of the SCHAFT robot, which relies on real-time teleoperation.

The issue may be moot. Last year, Google acquired SCHAFT (as well as Boston Dynamics), and rumors suggest the company will withdraw its robot from the competition, preferring to focus on commercial robotics rather than seek funding from the military. The SCHAFT website has been taken down (except for a home page with no information), and Google did not respond to a request for comment. Still, according to DARPA's Pratt, "We are not presently aware of any plans to withdraw."

DARPA says the DRC Finals will be held sometime between December 2014 and June 2015.



The MIT team used assisted perception and assisted planning to program this Boston Dynamics Atlas robot, which achieved fourth place in the DARPA Robotic Challenge trials.

PHOTOGRAPH COURTESY OF TEAM MIT

Further Reading

Walter, M.R., Hemachandra, S., Homberg, B., Tellex, S., Teller, S.
Learning Semantic Maps from Natural Language Descriptions, *Robotics: Science and Systems*, June 2013, Berlin

Walter, M.R., Friedman, Y., Antone, M., Teller, S.
One-shot Visual Appearance Learning for Object Reacquisition, *International Journal of Robotics Research* (Special Issue on Robot Vision), vol. 31, no. 4, pp. 554-567, April 2012

DARPA YouTube channel, with videos of the Trials: https://www.youtube.com/user/DARPAtv

Logan Kugler is a freelance technology writer based in Silicon Valley.

Society | DOI:10.1145/2617664

Esther Shein

Holographic Projection Systems Provide Eternal Life

Optical tricks help deceased entertainers keep on performing.

WHEN ACTOR PHILIP Seymour Hoffman died unexpectedly in February, he still had at least a week of filming left on *Mockingjay*, the third and fourth installments of the wildly successful *Hunger Games* movie series. Luckily Hoffman, like other deceased entertainers before him, including rapper Tupac Shakur and singers Michael Jackson, Frank Sinatra, and Elvis Presley, has the potential to live on and charm new audiences through the use of so-called 3D holographic projection systems.

Holography is a technique for recording the wavefront of a 3D scene on a 2D image, according to Ting-Chung Poon, a professor in the Bradley Department of Electrical and Computer Engineering at Virginia Technical Institute. "Dating back a few decades, the process can only be conducted through optical means by mixing the object wave with a reference beam, and recording the resulting fringe pattern on a photographic film,'' Poon and colleague P.W.M. Tsang wrote in the 2013 paper *Review on Theory and Applications of Wavefront Recording Plane Framework in Generation and Processing of Digital Holograms*. "With the rapid advancement of computing technologies in recent years, the optical hologram formation mechanism can be simulated with numerical means."

A holographic image of Michael Jackson performs onstage during the 2014 Billboard Music Awards in Las Vegas, nearly five years after the pop star's death.

PHOTO BY KEVIN WINTER/BILLBOARD AWARDS 2014/GETTY IMAGES FOR DCP

The Role of Computers

This approach, commonly referred to as computer-generated holography (CGH), computes the diffraction pattern emitted from a 3D object, and adds a reference wave to produce a digital hologram that can be displayed on electronic devices, says Poon, also a fellow of The Optical Society. "Computer-generated holography deals with the methods of digitally generating holograms. The hologram can be subsequently printed on a film or loaded onto a spatial light modulator (SLM) for holographic reconstruction,'' Poon explains.

The phrase "digitally generating holograms" refers to the computer calculations of a hologram for a 3D object, he says. Digital holography (DH) has been made possible through ongoing advances in computing and in optical scanning, along with the growing availability of high-capacity digital storage and wide-band communication technologies.

Merging 19th-Century and Present-Day Technologies

Holography has been a boon for the entertainment and arts industries in particular, with technology that may be used to bring dead performers "back to life." Musion 3D, for example, a London-based company that develops, markets, produces, and broadcasts 3D holographic illusions, has utilized its Eyeliner technology to resurrect a number of deceased entertainers, including comedian Les Dawson for an ITV appearance in June 2013—an event planned prior to his death in June 1993.

Musion Eyeliner was created in 1995 when company founder Uwe Maass, a laser show engineer, was commissioned to produce a large Pepper's Ghost display for crystal manufacturer Swarovski. Pepper's Ghost refers to a 19th-century light projection system that produces lifelike stage effects using plate glass or transparent films and special lighting.

"The [Swarovski] display required an alternative to the use of glass sheets, which hitherto had been the material used to create the illusion, such as in Disney's Haunted House attraction in California,'' says Ian O'Connell, director of Musion 3D. "After researching various polymer compounds, Maass found wide film from Hoechst (now Mitsubishi), which provided the scale required for the image. The large scale achievable, coupled with the advances in HD video projection during this period, provided a renaissance for the Pepper's Ghost platform."

There are three key components to an Eyeliner system: a specially developed foil to reflect images from high-definition video projectors; the stage, and the content. The stage is configured in a manner similar to a live music set with audio and lighting, says O'Connell. The foil is configured at a 45-degree angle across the entire stage, "tensioned [taut] within truss framing so as to be invisible to the watching audience." The content is typically 3D computer-generated imagery (CGI), combined with 35mm film or high-definition video (1080p or 1080i) to complete the effect. "Computers are used in the creation of content, the automation of stage operations such as synchronizing lighting control with video, and finally, in playing back the video content,'' he says.

Eyeliner is used primarily for product visualization events such as trade shows, fashion shows, and musical concerts, according to O'Connell. "However, new uses are being created by different industries,'' he adds. In February, for example, Interblock S.A., which supplies gaming products to casinos, unveiled its Holographic Gaming Lounge using its own proprietary computer real-time number generator in combination with Musion technology, at the International Casino Exposition/ICE 2014.

"At times, we are streaming tens of millions of pixels of uncompressed video content for GPU. This not only has huge visual processing requirements, but also massive data implications."

AV Concepts, a provider of holographic and immersive technology, creates holographic illusions using specialized holographic foil, lighting effects, and a proprietary Liquid Scenic projection mapping media server. "The server takes completely uncompressed imagery and uses multiple high-brightness projectors that integrate seamlessly to create very powerful, life-like performances," explains creative design engineer Alok Wadhwanir. Highly tensioned foil is used to reflect the images.

The Tempe, AZ-based company has done a number of projects using holographic illusion technology in a variety of industries, Wadhwani says, including working with a major hotel chain to bring its CEO to the stage virtually, with a shoe retailer to launch a new sneaker and show the audience the shoe's internal technology holographically, and with entertainer Dolly Parton to bring the holographic ghosts of Christmas to her Dollywood theme park's rendition of *A Christmas Carol*.

Michael Jude, consumer communications services program manager at research firm Frost & Sullivan, cautions that what Musion and others are doing is not pure holographic projection. There are no actual holographic projection systems commercially available, he maintains, adding that these companies are simply creating special effects à la Pepper's Ghost. "It is a well-known optical illusion coupled with CGI ... it takes a laser to generate a real hologram."

Currently, the only way to do holographics is by looking through a holographic plate/image, which is an interference pattern, Jude says. "If you're looking to bring Elvis back, you have to generate him somewhere else before projecting [the image] ... and that is heavily computer-dependent, and it is not real-time."

Generating a hologram requires bouncing a laser off whatever it is you want to capture. "You illuminate that [object] with a laser beam and that light bounces off the [object] and strikes a photographic plate,'' Jude explains. "Then you shine the same laser beam at the thing at the same time you're illuminating the photographic plate, to create the interference." Once the flat interference pattern has been created on the plate, the user can shine the laser through it to generate a hologram.

"It looks very realistic, but transmitting that interference pattern in such a way that it can present a hologram is really, really hard to do,'' Jude says. "It has only been done in labs." Also, true holograms do not appear in full color, Jude says, because they are generated by an ultra-pure laser beam of a specific hue, so they are typically green or red.

So what is being called a holographic projection system commercially is really just "good marketing, and it sounds cool," Jude says.

Like Musion, Wadhwani acknowledges AV Concepts' high-definition holographic projection system is based on the Pepper's Ghost illusion. "We've updated it with 21st-century technology to create a three-dimensional, life-size illusion that moves and interacts within a live stage setting,'' he adds. "Creating holographic illusions involves specialized, highly tensioned foil, lighting effects and our ... Liquid Scenic projection mapping media server to create dynamic visual experiences."

Software and hardware play a crucial role in producing, designing, creating, and executing the illusions, he says. Visual processing power is supplemented by powerful CPUs and the growing popularity of parallel programming standards such as OpenCL, he says, which give AV Concepts the ability to create and render extremely large, immersive, and "incredibly captivating visual media that was never before possible." However, Wadhwani adds, there can be complications when trying to create this type of processor-intensive media and ensuring it is reliable, plays in real-time, is synchronized, and is of high quality.

"At times, we are streaming tens of millions of pixels of uncompressed video content per GPU. This not only has huge visual processing requirements, but also massive data implications, leading us to utilize high-volume RAID arrays of solid-state disks,'' Wadhwani says. "Networking these systems together creates a powerful server architecture that can be reliably integrated into extremely challenging environments."

Holography as Art

Ikuo Nakamura, founder of Hololab Studio in New York City, learned about holography during his study of physics at Tokyo University of Science. Nakamura uses a ruby pulse laser to shoot portraits or live objects, and has created holography installations (http://www.hololab.com/holographyart.php) for which he has received numerous awards.

Nakamura works with both digital and 3D film, and says computers play an integral role in holography, since they preserve memories. "For me, [the] computer is [an] extension tool,'' he says, which is used as an "interactive device."

He also says "I want to make it clear, projection on transparent screen[s] is not holography. Holography has true 3D volume."

Looking Ahead

In terms of creating computer-generated holographic images, it now is possible to generate what Jude calls "a synthetic interference pattern," but its resolution is very low. "Ultimately, it will be possible to generate fairly complex projections, but the amount of computing [capacity required] is fairly high,'' he says.

As the technology progresses, Wadhwani says, the possibilities are infinite. He says AV Concepts recently developed new scalable holographic displays able to quickly, easily, and cost-effectively create holographic illusions on a much smaller scale—something that historically has required expensive custom hardware, software, and engineering, he says.

Future plans for Eyeliner are "to make the images even more realistic, through a combination of 4K video and much brighter image projection using latest-generation LED technology."

Meanwhile, O'Connell says Musion is working on resurrecting the next dead entertainer with Eyeliner. "All I can say is that the entertainer is male, is still a huge star today and will likely play his first concert in the USA around May or June this year.''

Further Reading

Tsang, P. W. M., Liu, J. P., Cheung, K. W. K., Poon, T. C.
Modern method for fast generation of digital holograms. 3D Research Center and Springer, 2010.

Tsang, Peter, Cheung, W.K., Poon, T.C., Zhou, Z
Holographic video at 40 frames per second for 4-million object points. Department of Electronic Engineering, City University of Hong Kong, Hong Kong; Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA,; and Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Science, Shanghai, Optical Society of America, 2011.

Tsang, P.W.M., and Poon,T.C.
Review on theory and applications of wavefront recording plane framework in generation and processing of digital holograms. Department of Electronic Engineering, City University of Hong Kong, Hong Kong; Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA. Chinese Optics Letters, 2013.

Minas K. Balvan
Object image correction using an X-ray dynamical diffraction Fraunhofer hologram. *Journal of Synchotron Radiation*, Vol. 21, Part 2. Epub March, 2014.

Bergstrom, P., Khodadad,D., Hallstig, E., Sjodahl, M.
Dual-wavelength digital holography: Single-shot shape evaluation using speckle displacements and regularization. U.S. National Library of Medicine, National Institutes of Health. Applied Optics. January, 2014.

Smalley, D.E., Smithwick, Q.Y.J., Bove, V.M., Barabas, J., Jolly, S.
Anisotropic leaky-mode modulator for holographic video displays. *Nature*, 498, 313-317. June, 2013.

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

ACM Member News

ALEXANDER M. WOLF ELECTED ACM PRESIDENT

Alexander M. Wolf, professor and holder of a Chair in Computing in the Department of Computing of Imperial College London, was elected president of ACM in the May 2014 general election.

Previously ACM vice president, Wolf says, "As an organization we must confront the reality that what ACM contributed to the computing profession for more than 65 years might not sustain it in the future." He says his vision as president is to position ACM to better meet the challenges posed by the evolving computing community in the age of computer-mediated, cost-free, dynamic social networks.

The election results include:

PRESIDENT
Alexander M. Wolf,
Imperial College London
(July 1, 2014–June 30, 2016)

VICE PRESIDENT
Vicki L. Hanson,
University of Dundee
(July 1, 2014–June 30, 2016)

SECRETARY/TREASURER
Erik R. Altman,
IBM T.J. Watson Research Center
(July 1, 2014–June 30, 2016)

MEMBERS AT LARGE
Cherri M. Pancake,
Oregon State University
(July 1, 2014–June 30, 2018)
Per O. Stenström,
Chalmers University of Technology
(July 1, 2014–June 30, 2018)

The number of votes polled by each candidate:

President	
Alexander L. Wolf	6,195
Ronald Perrott	2,718
Vice President	
Vicki L. Hanson	5,304
Eugene H. Spafford	3,664
Secretary/Treasurer	
Erik R. Altman	5,093
David A. Wood	3,702
Members at Large	
Per O. Stenström	4,896
Cherri M. Pancake	4,869
Victor Bahl	4,175
Madhavan Munkund	1,711

DOI:10.1145/2618099

Pamela Samuelson

Legally Speaking
Watching TV on Internet-Connected Devices

The ABC vs. Aereo case has potentially far-reaching consequences.

SHOULD YOU BE able to watch your favorite broadcast television programs on your iPhone or PC with the aid of a third-party provider's equipment? Aereo thinks the answer is yes and has designed a system to accomplish this task.

ABC, several other networks, and some broadcast television stations think not. They have sued Aereo for copyright infringement, claiming that Aereo is commercially retransmitting their works to subscribers in violation of the public performance right of copyright law.

Although Aereo successfully defended its service in some lower courts, the Supreme Court agreed to hear ABC's appeal. Oral argument was held this past April, and by the end of June we should know whether ABC can stop Aereo from providing this service.

After describing Aereo's system, this column reviews ABC's and Aereo's core legal arguments. It then considers what is at stake not only for ABC and Aereo, but also for cloud computing and similar services that store or stream copies of digital content on behalf of consumers. While I often predict the outcome of appellate cases, *Aereo* seems too close to call at this point.

Aereo's System

One way to understand Aereo's system is to view it from the users' perspective. When subscribers log onto Aereo's system, they see a program guide for over-the-air broadcast television programs airing at that time or in the near future on local stations. Subscribers can select which programs they wish to watch or have recorded.

If a user presses the "watch" button, Aereo streams the program to the user's device, although there is a slight delay in transmission because Aereo records the program on servers so that if the user presses "record" during the program, the user will be able to capture the program from the point at which she first began watching it. A user can also press "record" to capture future programs, which Aereo automatically stores for later viewing.

From the users' perspective, the Aereo system functions like a TV set with a remote digital video recorder (DVR) and capabilities that technologies such as Slingbox provide to view television programs on Internet-connected devices. Only local over-the-public-airwaves television programming is available through Aereo's system.

Another way to understand Aereo's system is by considering its technical infrastructure. Aereo operates a facility in Brooklyn, NY, with large banks of antennas and servers. When a user selects a program to watch or record, Aereo's system sends a signal to an antenna that Aereo assigns to that customer and tunes that antenna to the broadcast frequency of the channel on which the desired program is showing or is slated to show.

Aereo transcodes the data for that program, buffers the data, and then sends it to an Aereo server where a copy of the program is saved to a directory

PHOTOGRAPH BY MONKEY BUSINESS IMAGES

on a hard drive reserved for that user. The program is then streamed to the user's Internet-connected device, either approximately as the program is showing or a later time.

ABC's Legal Argument

For the past 38 years, it has been beyond cavil that retransmitting broadcast television signals by services such as cable or satellite television providers is a public performance requiring copyright owner permissions. Broadcasters these days rely heavily on cable and satellite retransmission revenues to support their programming and operations.

ABC argues that Aereo's service is the functional equivalent to cable and satellite television services, so it too should be within the public performance right. Indeed, Aereo has advertised its service as an alternative to these paid services. Aereo can offer a lower price to its subscribers at least in part because, unlike cable and satellite providers, it does not pay anything for retransmitting broadcast programs.

ABC complains that Aereo's "entire business model is premised on massive for-profit unauthorized exploitation of copyrights where competitors' prices are undercut because they seek authorization and pay fees."

ABC contends that Aereo's activities cause several types of harm. Cable and satellite providers are using the existence of Aereo's services to pressure broadcasters to reduce retransmission fees they now pay. Some threaten to reconfigure their systems to be more like Aereo's so that they too can avoid paying retransmission fees. More subtle harm arises from the lack of Nielsen ratings for Aereo subscribers' viewing of programs, which affects the advertising fees the networks and stations can charge.

Aereo's Defense

Aereo's main defense is based on the "undisputed fact" that its users are the ones who transmit the programs and each user can transmit the programs only to themselves. This, Aereo argues, is a private performance of the broadcast programming, not a public one. Copyright law does not give owners the right to control private performances.

Aereo merely "suppl[ies] remote equipment that allows a consumer to tune an individual remotely located antenna to a publicly accessible over-the-air broadcast television signal, use a remote DVR to make a personal recording from that signal, and then watch that recording."

Since the Supreme Court's 1984 *Sony Betamax* ruling, it has been settled that consumers can lawfully make time-shift copies of broadcast television programs to view at a later time. That decision also affirms the right of third parties to supply consumers with technological tools through which to make time-shift copies.

(The Court held that Sony was not liable for contributory infringement because its Betamax machines had substantial non-infringing uses (to wit, enabling consumers to make time-shift copies of broadcast television programs), even though Sony knew that some consumers would use the machines to make infringing copies of the programs.)

Even if two Aereo users decide to watch the same program at the same time, Aereo notes that its system automatically makes a separate copy for each user. Each recording is unique "not only in the sense that it's personal, but also because owing to electronic interference, technical glitches, and occasional equipment failure, no two copies are identical." The only person able to access the recording supplied through use of Aereo's system is the subscriber who ordered it.

Aereo's system is a quite different in technical design and operation than cable and satellite systems. They receive broadcast programs through a single feed and continuously retransmit that content to their customers. The constancy of those transmissions to subscribers is why cable and satellite transmissions are public performances. Aereo, by contrast, enables its subscribers to transmit specific television programs only to themselves.

What's at Stake?

ABC and its co-plaintiffs contend that the future of over-the-air broadcast television is at stake. Without retransmission fees from those who commercially exploit broadcast programming, incentives to invest in broadcast programming will be undermined, along with the ability to recoup these investments. Some may stop broadcasting and move their programs behind paywalls. Broadcast television has played an important role in enriching civic life and it would be a tragedy to let it fail.

Several copyright industry organizations have filed friend-of-the-court briefs in support of ABC's appeal, asserting that upholding the ruling in favor of Aereo would have baleful consequences for many copyright owners.

But the stakes are high for many information technology companies if the Supreme Court rules in ABC's favor because it would make liability depend on the technical infrastructure design of their systems.

Cablevision

The main ruling supporting *Aereo* involved a company called Cablevision. Copyright owners that provided programs to Cablevision sued it for infringement because this cable company began to offer a remote DVR service to its customers. They claimed the remote DVR service infringed their reproduction and public performance rights. An appellate court ruled that Cablevision did not need a separate license from broadcasters and other content providers to offer the remote DVR service.

The technical infrastructure of the Cablevision system mattered. Cablevision designed its system so that each subscriber could direct that a particular program be saved on Cablevision's servers. The system would then store that program in a server directory assigned to that subscriber. The user could later watch the program at a time and place of his or her choosing, which in effect, caused the program to be transmitted to him or her.

ABC and its co-plaintiffs contend that the future of over-the-air broadcast television is at stake.

The court in *Cablevision* ruled the buffer copies made by the remote DVR system operations were too ephemeral to be infringing reproductions and the DVR program copies stored on Cablevision servers were made by individual subscribers, not by Cablevision.

That court also rejected claims of infringement of the public performance right because Cablevision's subscribers were the ones who transmitted DVR programs to themselves, and no one else could access that DVR copy but the subscriber who ordered it. Each transmission was, therefore, a private performance of the work. Cablevision engaged in no volitional conduct that caused DVR transmissions to occur.

Cablevision is factually distinguishable from *Aereo* because Cablevision had authorization from copyright owners to transmit programming to its customers and it was paying fees to copyright owners for these transmissions. The question was only whether Cablevision was obliged to get a separate license and pay a new round of fees to enable the remote DVR service.

Rube Goldberg Design?

ABC has characterized the Aereo system as a "Rube Goldberg-like contrivance" that was designed "to take advantage of [the] perceived loophole" in the statutory definition of the public performance right.

This may be true, but Aereo is only one of many information technology services that have taken advantage of the *Cablevision* ruling. A study by Harvard professor Josh Lerner reported that in the two-and-a-half years after the *Cablevision* ruling, capital investments in U.S. cloud computing grew somewhere between $728 million and $1.3 billion with a positive effect on job creation. This was, in Lerner's view, equivalent to $2.5–$5 billion in traditional R&D investments. Had the *Cablevision* case gone the other way, Lerner's study suggests these investments might not have been made. Many cloud service providers (for example, backup storage providers), after all, transmit copies on behalf of users.

Conclusion

It will be interesting to see what the Supreme Court does in *Aereo*. There are four possible outcomes.

First, the Court may say the current definition of the public performance right does not encompass transmissions enabled by Aereo's system, and if Congress wants to outlaw services such as Aereo's, it must amend the law.

Second, the Court could parse the public performance right to hold Aereo as an infringer, but affirm *Cablevision's* interpretation of the public performance right as to other systems. Some friend-of-the-court briefs on behalf of cloud computing and other technology providers may offer suggestions about this.

Third, the Court may overturn both *Cablevision* and *Aereo* so that technology developers can be held directly liable for infringement because their systems transmit copyrighted materials to users.

Fourth, the Court may affirm the right of technology providers to facilitate personal use copying and viewing of copyrighted content and affirm the ruling in favor of Aereo.

Aereo's system is the latest example of an innovative, inexpensive technology that enables consumers to enjoy copyrighted works in new ways. Will copyright knock this one and others out of the market? Stay tuned for the outcome of this important case.

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

DOI:10.1145/2618103

Chuck Huff and Almut Furchert

Computing Ethics
Toward a Pedagogy of Ethical Practice

Teaching computing ethics in a manner that allows students to address both abstract ethical knowledge and actual ethical practice.

There is a tension at the heart of our curriculum in computer ethics. We want to prepare students for the wide range of ethical concerns in the practice they will find upon graduation. But primarily, we teach knowledge, intermixed with such practice as fits in the spaces of the lecture/discussion course.

Our recent work on "moral exemplars" in computing[a] found a variety of ways that professionals approached doing good.[7] Some designed systems to help individuals or organizations. Alan Newell, for instance, has been designing software and hardware to help the disabled—his lab did some of the earliest work on predictive spelling systems. Others focused on changing social systems, like Stephanie (Steve) Shirley who brought many women into computing through her pioneering efforts in software consulting.

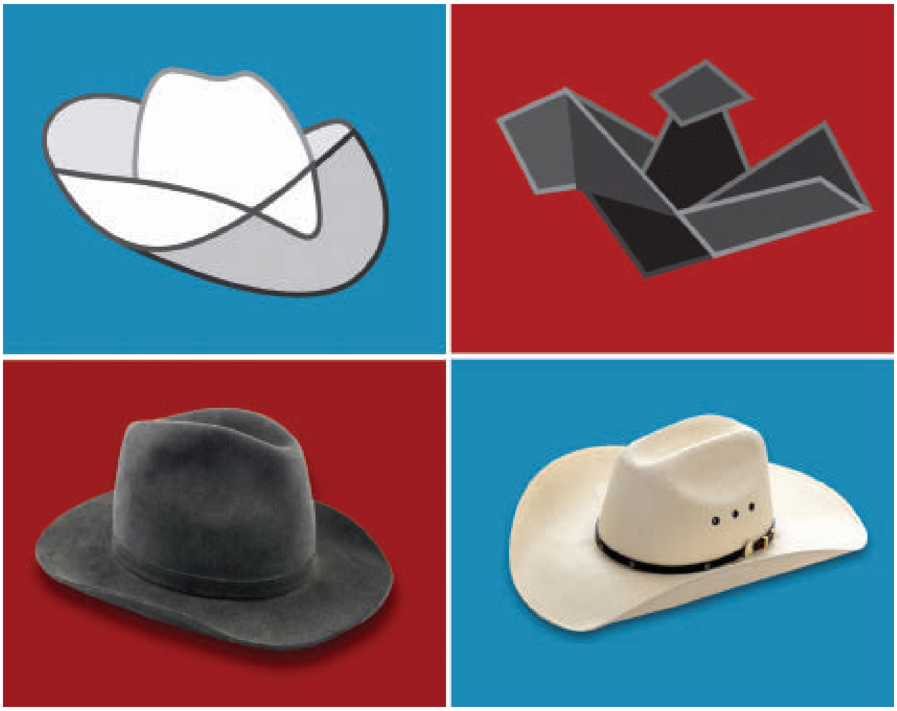


ILLUSTRATION BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

This variety of ethical practice is a normal finding in work on moral exemplars.[2] But among the variety, we also find similarity: the exemplars' language in the narrative interviews we conducted was filled with expressions implying significant ethical commitments (for example, "to look for the needs that people and organizations have;" "quality, fitness for purpose;" "openness, transparency"). These fit items one can find in the software engineering ethics code (see the accompanying table). Oddly, although some helped write ethics codes, *none of the interviewees ever mentioned any code of ethics*, even when asked specifically about principles.

So on what were they basing their moral action? Among those who were designing computing systems, the center of their craft was recognizing the needs of stakeholders and using their expertise to reframe those needs into things that computing could help them do. They used social expertise to listen to individuals and organize networks of individuals to get things done. This blended into technical expertise in socio-technical and requirements analysis. And it included deep technical skill in encryption, database design, and networking protocols, among other topics. Thus, our exemplars were exercising an expertise or skill to integrate their ethical commitments with their knowledge and

[a] We did in-depth interviews with a careful sample of 24 moral exemplars in computing, in the hope that understanding the lives of these experts in ethical computing practice we might learn how to teach that expertise. We are grateful to the exemplars who took time for our project, and to those whose names we use by permission.

know-how of technical and social systems.

Aristotle's early analysis of ethical action hints that ethical knowledge (the what) and ethical skill (the how) are intermingled in any practice. Alongside knowledge of the good, we need to gain practical wisdom (phronesis) that guides our ethical action. We acquire practical wisdom by practicing, we "become builders by building and lyre players by playing the lyre; so too we become just by doing just acts..."[1] Here lies what the German philosopher Hans-Georg Gadamer[5] has described in the context of medical practice as the application problem. The unavoidable difficulty for any medical professional is "applying [medical] knowledge in the concrete case." Gadamer also claims this application problem is "irreducible ... where ever knowledge in general needs to be applied." Bridging this gap is an act of constant evaluation, reasoning, and decision making to translate what we know into doing—and this translation unavoidably brings the ethical dimension into every aspect of professional practice. If we want to teach ethical practice in computing, we must therefore ask what competencies, skills, or virtues are needed to translate ethical knowledge into ethical practice in the field.

Herein lies an uncomfortable tension: While the ethics code is full of the obligation to design systems with ethical concern in mind, the typical computing ethics textbook does not help one learn how to do that. One can learn knowledge about legal, philosophical, and societal issues of privacy in a classroom, and even practice thinking about these issues with cases. Some of these skills (identifying stakeholders, making an ethical argument) are useful when one has the problem of designing a computing system, but one must remember them, and remember they apply, and know how to adapt them to the concrete case. The *Social Issues and Professional Practice* ACM curriculum[3] points us in the direction of balancing knowledge with practice by setting "usage" objectives (what one must be able to do with the knowledge).

But to implement these recommendations requires a pedagogy of ethical practice. How does one, for instance, "... address ethical ... issues related to work projects" per section 3.03 of the software engineering ethics code? It requires knowledge about ethical issues in general, but also know-how to identify them in a particular project. It requires knowledge about the socio-technical system, but also know-how about how to acquire such knowledge. It requires knowledge about best practices, but also know-how about selecting and adapting those practices for a specific social and organizational context.

Research suggests one best acquires expertise by long practice, informed by knowledge and theory of the domain and with coached feedback about performance that is immediate and explicit.[4] This fits with what we have heard from the philosophical approaches and the narratives told by our moral exemplars. All three lead us to think our moral exemplars have developed an expertise in the ethical practice of their profession; they have extensive, skilled practice guided by ethical commitments and knowledge.[b,6] This is perhaps why our ethical experts do not cite the code. They learned their ethical design skills by practicing them until they became automatic, thoughtful, goal-directed action. They did not reference the high-level principles of the code since they were integrated into their skilled practice and had become part of their expertise.

We have developed a computing ethics class that attempts to teach ethical expertise in practice. Students provide clients with consulting on ethical issues associated with their computing systems (see http://pages.stolaf.edu/csci-263-2014/). They construct a model of the socio-technical system based on interviews with the client and then look for social and ethical issues (including safety, privacy, property, justice) at the individual to the societal level.[8] They scope their project based on time, resources, and urgency of the issues. They then analyze those issues in that socio-technical system using human-computer interaction (HCI) approaches to data

[b] Moral expertise is a central aspect of a forthcoming book titled *Taking Moral Action*.

Items from the ACM/IEEE Software Engineering Code of Ethics.

These 16 items are selected because they are clearly relevant to having competence in the consideration of ethical aspects in the design of software.

1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy, or harm the environment. The ultimate effect of the work should be to the public good.

1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.

1.07. Consider issues of physical disabilities, allocation of resources, economic disadvantage, and other factors that can diminish access to the benefits of software.

2.07. Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.

3.01. Strive for high quality, acceptable cost, and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.

3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.

3.03. Identify, define, and address ethical, economic, cultural, legal, and environmental issues related to work projects.

3.04. Ensure that they are qualified for any project on which they work or propose to work by an appropriate combination of education and training, and experience.

3.05. Ensure an appropriate method is used for any project on which they work or propose to work.

3.06. Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified.

3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements, and have the appropriate approvals.

3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.

3.12. Work to develop software and related documents that respect the privacy of those who will be affected by that software.

3.13. Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.

3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrences.

4.01. Temper all technical judgments by the need to support and maintain human values.

These 16 statements represent 20% of the total of 80 statements in entire code, and over 70% of the items in the section on responsibilities having to do with the production of a product (section 3).

collection (for example, interviewing, focus groups, active and passive observation, think-aloud protocols and so forth). Finally, they construct proposed solutions and make design recommendations. The course is built on the dialogue among concern for ethical issues, knowledge about computers in context, and the practice of working for a client. They learn about ethical and social issues from a textbook[9] and from constructing solutions for their clients. They learn about socio-technical systems from the ethics text and also by using methods from an HCI text.[10] During the past decade, teams from the course have produced over 50 studies for non-profit, for-profit, and internal clients. They have made recommendations for Web design, for customizable privacy settings in social networks, for medical database systems, key-card access systems, productivity software for robotic manufacturing, cloud-based systems, and many more. The goal is that students leave the course having learned how to hold together their ethical concern, knowledge, and practice. This is the beginning of expertise in ethical issues in computing design.

There already exist a variety of methods for the incorporation of ethical concerns in systems design that might be used in courses (see Shilton[11] for an overview). We hope our suggestions here will encourage further experimentation to incorporate them into the curriculum. For instance, in those programs that have HCI labs, one might arrange for significant overlap between the lab and a supporting course on ethical issues. Programs with project courses in other areas (such as networking, information management, parallelism) could incorporate into those projects some of the ethical design approaches that might be taught in more detail in a computer ethics course.

As educators of those who will design future technology, we have a responsibility to prepare our students to practice their craft in a way that integrates their ethical concern into their work. We propose that it is necessary and possible to teach computing ethics know-how that helps students to navigate between abstract ethical knowledge and its actual ethical practice. By doing so, the students gain experience and expertise in applying what they know in the concrete case.

References

1. Aristotle. Nicomachean ethics. In *The Basic Works of Aristotle*, R. McKeon, Ed. Random House, New York, 1941, 927–1112.
2. Colby, A. and Damon, W. Some do care: Contemporary lives of moral commitment. Free Press, New York, 1992.
3. *Computer Science Curricula 2013.* ACM/IEEE Computer Society, 2013.
4. Dreyfus, H.L. and Dreyfus, S.E. The ethical implications of the five-stage skill-acquisition model. *Bulletin of Science, Technology, & Society 24* (2004), 251–264.
5. Gadamer, H.-G. *The Enigma of Health.* Stanford University Press, Stanford, CA, 1996.
6. Huff, C.W. From meaning well to doing well: Ethical expertise in the GIS domain. *Journal of Geography in Higher Education*, in press.
7. Huff, C.W. and Barnard, L.K. Good computing: Moral exemplars in the computing profession. *IEEE Technology and Society Magazine* (2009), 47–54.
8. Huff, C.W. and Martin, C.D. Computing consequences: A framework for teaching ethical computing. *Commun. ACM 38*, 12 (Dec. 1995), 75–84.
9. Johnson, D. *Computer Ethics, 4th Edition.* Pearson, New York, 2009.
10. Rogers, Y., Sharp, H., and Preece, H. *Interaction Design: Beyond Human-Computer Interaction.* Wiley, New York, 2011.
11. Shilton, K. This is an intervention: Foregrounding and operationalizing ethics during technology design. In *Emerging Pervasive Information and Communication Technologies (PICT): Ethical Challenges, Opportunities, and Safeguards, 11*, K.D. Pimple, Ed., Springer, New York, 2013, 177–192.

Chuck Huff (huff@stolaf.edu) is a professor of psychology at St. Olaf College, Northfield, MN.

Almut Furchert (furchert@stolaf.edu) is a senior visiting research fellow in the Hong Kierkegaard Library at St. Olaf College in Northfield, MN.

This work was partially supported by NSF grants DUE-9980786, DUE-9972280, SES-0217298, and SVS-0822640. We are grateful to Don Gotterbarn, Keith Miller, and Herman Tavani for conversations and advice about this column.

DOI:10.1145/2618105

Mari Sako

Technology Strategy and Management
The Business of the State

Considering the opportunities and challenges for commercial firms involved with government business process outsourcing.

OUTSOURCING IS BIG business, demand for which is generated in no small part by governments. In recent years, public sector outsourcing has outstripped private sector outsourcing, with public sector organizations around the world spending $10.3 billion in the third quarter of 2013 on IT outsourcing and business process outsourcing, compared with $6.4 billion in the private sector. The top three spending governments are in the U.S., Britain, and Australia.[2] Outsourcing creates business opportunities for commercial firms in a wide range of sectors including IT, defense, security, detention and prison service, healthcare, transport, and shared services.

But public-to-private transactions in government outsourcing are rife with reputational risks for corporations. We consider the following. What has motivated governments to outsource their operations and services? What is the logical limit to government outsourcing, that is, are there things that only governments can do? And how can commercial firms balance the opportunities and challenges of providing services that used to be delivered by the government? Specifically, what are the reputational risks for corporate providers, and how is the management of these risks central to the future of government outsourcing? Answering

PHOTOGRAPH BY DAVID REILLY/SHUTTERSTOCK.COM these questions, and understanding what is similar but also different between public sector and private sector outsourcing, is of enduring importance as many companies encounter governments as their biggest clients.

What Do Governments Outsource?

The range of things governments used to do themselves, which they now outsource, is very extensive. The U.S., for example, awarded more than $500 billion in outsourcing contracts in the 2012–2013 fiscal year. Likely most familiar to *Communications* readers is the outsourcing of IT services, including data management, infrastructure management, application development, and customer support. Over the past decades, many e-government initiatives at national and local levels have been digitizing citizens' records as taxpayers, benefits recipients, and medical patients.

Another significant area of contracting out is the financing, construction, and management of hospitals, prisons, and transportation infrastructure. The Private Finance Initiative (PFI), known as Public-Private Partnerships (PPP) in some countries, funds public infrastructure projects with private capital. In the last 15 years, the British government and local authorities signed some 600 PFI contracts with a capital value of over $100 billion, including approximately 100 hospital schemes, 100 educational projects, and 43 transportation-related projects.

National governments and international organizations such as the United Nations have also come to outsource peace-keeping and relief efforts in conflict zones, by making use of private military and security companies.[9] Personnel employed by these companies may be staffed as unarmed or armed guards, and engage in a variety of settings including demining, logistical mobile security, and the operation of armed vehicles and secure telecommunications.

Governments also outsource services to each other, giving rise to 'G2G' (government-to-government) trade. From German police overseeing security in the streets of Mumbai to the U.S. Federal Aviation Administration (FAA) training Chinese pilots, and the Australian-led Regional Assistance Mission to the Solomon Islands (RAMSI) taking over law enforcement on the islands, the notion of the "unbundling of the nation-state" is newsworthy, if not totally new.[5]

Economic reasons for government outsourcing are simple to understand, as they have parallels in the private sector.

Why Do Governments Outsource?

Government budget cuts have provided the most tangible impetus for recent waves of contracting out. However, it is wise to remember also the ideological underpinnings for various forms of government outsourcing, if we wish to predict how much demand for outsourcing might occur in the future.

Economic reasons for government outsourcing are simple to understand, as they have their parallels in the private sector. In this context, government organizations are not all that different from large corporations. Both want to save money and operate more efficiently. Governments commit to infrastructure investments by balancing the need for more efficient services with the need to rein in costs. Outsourcing provides a solution to this balancing act.

Ideological reasons for public sector outsourcing have a history going back to the Reagan-Thatcher era of the 1980s.[6] Neoliberalism—a belief in markets promising choice and competition—became articulated in policies to make governments smaller (via privatization) and more efficient (via new public management).

Since the 1980s, privatization involved the government selling off big chunks of assets, for example in railways, electricity, gas, water, and telecommunications. At the same time, competitive bidding was introduced to build and manage hospitals and prisons. Thereafter, further waves of privatization have become less visible as contracting out takes the form of slicing off a small part of public services at a time.

New public management (NPM) has been applied to what remains of the government machinery after privatization and contracting out. The idea here is to make government more "business-like," by subjecting public servants to the same sort of market testing as in commercial settings. In particular, NPM advocates introducing business management techniques and competition between public agencies and commercial firms to make public service delivery more efficient.[1]

In the U.S., the idea of "reinventing government" by Osborne and Gaebler became popular during Bill Clinton's 1992 presidential election.[7] According to Osborne and Gaebler, we should stop arguing about too much government or too little government. Instead, we should focus on better government that promotes entrepreneurial competition among service providers, competition that empowered citizens by giving them greater choice. Noteworthy, thereafter, is the "end of ideology" on the issue, as subsequent governments of different ideological leanings, at least in the three leading nations of Australia, the U.S., and Britain, have not attempted to reverse this trend toward further contracting out.

Corporate Responses to Government Outsourcing

Wherever there is demand, firms go to fulfill it. At least, this appears to be the logic behind the rapid growth of companies that provide services to governments. The main irony is that in key sectors, the government has relied on the competitive bidding process, but ended up creating a highly oligopolistic market structure—precisely the structure that it intended to circumvent. Moreover, major service providers operate transnationally, having developed a variety of expertise through acquisitions in areas that had been reserved for public sector agencies.

The market leader, G4S, best illustrates the challenges faced by an organization of enormous scale and scope. It employs 620,500 and manages over 6,000 contracts in 125 countries, rivaling McDonald's in ubiquity. G4S provides services in areas ranging from security, defense, justice and police. The latter includes offender management, rehabilitation, courtroom security, and police services. In policing, the act of arresting is still carried out by police officers, but G4S staff go to the scene of the arrest, drive offenders away, and process them for fingerprints and other paperwork in the company's own "custody suites." In defense, G4S provides logistics and technical support, and risk management consultancy on mine detection and clearance. Guarding and escorting in prisons and detention centers also bring them in direct contact with prisoners and detainees, leading to managing high-risk services in difficult conditions.

Large companies have a clear strategy, for example to provide "security solutions." In practice, they have expanded and adapted to local jurisdictions, by following a salami technique, slicing off a small part of public services to see how far they can go.[4] Thus, high-risk contracts remain financially rewarding. And testing out in uncharted waters how much they can push the boundary of the state inch-by-inch is unlikely to set limits on how far they would go.

Is There a Limit to Government Outsourcing?

As large corporations develop capabilities in areas formerly the exclusive preserve of governments, how should one define the essential functions of government? Addressing this question is important for the sustainability of service providing companies as much as of governments.

In this context, Francis Fukuyama's distinction between the scope and the strength of the state is useful.[3] In the post-Washington Consensus era, reducing the scope of the state by liberalizing the market and privatizing state-owned assets concentrates the minds of government to ensure it retains strength, namely the capacity to provide law and order, and other public goods. One could argue that only the state can legitimately exercise force on its citizens and other nations. However, such a minimalist state could go further, and outsource the use of force to private providers under proper supervision.

In short, as we push the boundary of what can be delegated to the private sector, government becomes less the provider of public services, and more "the supervisor of proxies who do the actual work."[6] Yet, government relies on private partners to do government work. A central question is no longer what only the state can do that private sector firms cannot do. Instead, the question is: How far can we make the supervisory role of the state on private actors sustainable in variable circumstances, by making use of formal contracts and informal contractual norms?

Commercial firms that have governments as major clients operate at their peril if they are not fully cognizant of the motivations for public sector outsourcing.

Managing Reputational Risks Is Central to the Future of Government Outsourcing

This column thus far argued that governments continue to be market makers in outsourcing. It also asserts that commercial firms that have governments as major clients operate at their peril if they are not fully cognizant of the motivations—economic, political, ideological—for public sector outsourcing.

Looking into the future, we should be clear-headed about the challenges for commercial firms acting on behalf of governments. The key challenges lie in managing reputational risks. First, for-profit companies are unlikely to have objectives and values that match government's mission. In order to deliver efficiency without being seen to be driving it too far for profit-making, providers might consider adopting corporate forms other than a public limited company. Second, the general public tends to use a different set of metrics to judge corporations and governments. For some, being 'for profit' may lead to expectations of higher quality of service. The fact that profits are made out of taxpayers' money raises such expectations even further.

Third, the corporate reputation of providers might be at risk in democracies where they are seen to be not subject to democratic control. Citizens as consumers are likely to demand greater transparency and accountability from service providers in their attempt to exercise such control. Although large providers like G4S have developed a visible brand, this may be necessary but not sufficient. At the same time, remaining invisible behind government departmental logos, analogous to no-brand contract manufacturers as "behind-the-scene champions," as I discussed in a previous column,[8] appears not to be a viable strategy in government contracting.

Public service companies are beginning to take account of these reputational risks, and might consider refraining from pushing the frontier of the state further, even as governments are willing to slice off more and more of their services. What is at stake here is not just a matter of governments finding the next areas of savings in an attempt to balance the budget. Practicing the art of the state in the 21st century is being put to test, and private sector corporations could play a part in articulating it.

References

1. Dunleavy, P. and Hood, C. From old public administration to new public management. *Public Money & Management* (July–Sept. 1994), 9–16.
2. Flinders, K. Global public sector outsourcing far outstrips private sector. ComputerWeekly.com (Nov. 25, 2013).
3. Fukuyama, F. *State-Building: Governance and World Order in the Twenty-First Century.* London, Profile Books, London, 2005.
4. G4S: The inside story. *Financial Times* (Nov. 14, 2013).
5. Government-to-government trade: Unbundling the nation state. *The Economist*, (Feb. 8, 2014).
6. Kettl, D.F. *Sharing Power: Public Governance and Private Markets.* The Brookings Institution, Washington D.C., 1993.
7. Osborne, D. Ten ways to turn D.C. around. *The Washington Post* (Dec. 9, 1990).
8. Sako, M. Driving power in global supply chains. *Commun. ACM 54*, 7 (July 2011), 23–25.
9. Singer, P.W. *Corporate Warriors: The Rise of the Privatized Military Industry.* Cornell University Press, Ithaca, NY, 2003.

Mari Sako (mari.sako@sbs.ox.ac.uk) is Professor of Management Studies at Saïd Business School, University of Oxford.

DOI:10.1145/2618107

Jane Margolis, Joanna Goode, Gail Chapman, and Jean J. Ryoo

Broadening Participation That Classroom 'Magic'

Effective teaching practices for broadening participation in computer science.

WHAT CREATES THAT classroom "magic" when the most discouraged students engage actively, critically, and creatively with the subject at hand? Specifically, what does this look like in a computer science classroom, a subject that has historically attracted only a narrow stratum of students, leaving the majority feeling that they don't belong? How can this "magic" be described, defined, and measured? And which parts of this "magic" are the most effective for broadening participation in computing?

While an engaging classroom may feel "magical," it really is not. Rather, it is the result of purposeful instructional practices and a curriculum intentionally designed for broadening participation in computing. Though computer science educators often scrutinize curricular efforts, more attention needs to be paid to the particular teacher proficiencies that are most impactful for reaching diverse learners. Our research team has been conducting research in Los Angeles Unified School District (LAUSD) Exploring Computer Science (ECS) classrooms investigating the question: "What characteristics of high school ECS teaching practices are most effective for broadening engagement and participation in computing for students traditionally underrepresented in the field?"

Analysis of pre- and post-student surveys shows increased interest in and motivation to learn computer science after taking ECS.

Exploring Computer Science Program

The Exploring Computer Science (ECS) program consists of a high school introductory computer science course combined with an accompanying teacher professional development program. ECS was developed in response to previous research, detailed in *Stuck in the Shallow End*, which

PHOTOGRAPH BY JEAN RYOO

Young women make up approximately half of the Exploring Computer Science student population in LAUSD.

identified disparities in CS learning opportunities that fall along race and socioeconomic lines.[3] In that research we found that schools with high numbers of low-income students of color were offering keyboarding and other basic rudimentary computing skills as "computer science." Most students were not being prepared for the only available college preparatory computer science course, AP Computer Science (AP CS). To fill this need and to carry out our mission of broadening participation in computing, the ECS curriculum was written by two of our team members, Joanna Goode and Gail Chapman.

ECS consists of six units of approximately six weeks each, covering Introduction to Human Computer Interaction, Problem Solving, Web Design, Introduction to Programming (Scratch), Computing and Data Analysis, and Robotics. The ECS curriculum is structured to facilitate inquiry and equity-based instructional practices so that *all* students, especially those in schools with high numbers of low-income students of color, are introduced to the problem solving, computational practices, and modes of inquiry associated with doing computer science.

One metric of our success has been the LAUSD ECS student demographics that stand in sharp contrast to most other computer science courses. In 2013–2014, approximately 2,500 LAUSD students were enrolled in ECS, and 75% of ECS students were Latino (72% of district population), 10% were African-American (10% of district population), 9% were Asian (6% of district population), and 5% were White (10% of school population). Girls represented 45% of enrolled ECS students.

These ECS enrollment statistics are dramatically different from the participation rates of girls and students of color in national and state AP CS statistics: out of nearly 5,000 exam-takers in California last year, only 8% were Latino and 1% African-American. Only 22% of exam-takers were girls.[4]

Yet, even with these promising LAUSD ECS enrollment statistics, we recognize the work is not done. Teachers must work to transform ECS classroom culture and teaching so that *all* students experience and engage with foundational computing concepts and develop essential computational practices. Thus far, analysis of pre- and post-student surveys show increased interest in and motivation to learn computer science after taking ECS. We have partnered with SRI International to design assessment measures that will capture student knowledge, skills, and active learning.

Research to Examine Effective Teaching Practices

For the past several years we have conducted intensive mixed-methods research to understand which teaching practices support broadened participation in computing. In 2011–2012, we conducted 219 weekly observations in nine ECS classrooms. Through this ethnographic field research, along with several years of pre- and post-student surveys, teacher surveys, and student/teacher interviews as data sources, we have identified three strands of teaching practice critical for supporting broadening participation in computing.

- Computer Science Disciplinary Practices
- Inquiry Practices
- Equity Practices

It is important to note these strands are interweaving and inseparable, and that no strand can exist alone.

Computer Science Disciplinary Practices. While the more traditional computer science curriculum commonly focuses on programming and the computer as a tool, ECS focuses on the underlying problem solving and critical thinking necessary to explore effectively the wide array of topics that comprise the field of computer science. The curriculum is purposefully structured so the first two units put the focus on problem-solving (often without any use of computers) as well as setting classroom norms of inquiry, collaboration, equitable practices, creativity, and cognitively demanding problem-solving. Though each of the six units has a particular computer science content area of focus, all of the units and lesson plans incorporate the following *computational practices*:

- Analyzing the effects of developments in computing.
- Designing and implementing creative solutions and artifacts.
- Applying abstractions and models.
- Analyzing students' own computational work and the work of others.
- Communicating computational thought processes, procedures, and results to others.
- Collaborating with peers on computing activities.

These practices parallel those in the guiding framework for the new Advanced Placement CS Principles course. The actual classroom integration of these disciplinary practices then depends on inquiry and equity-based teaching practices.

Inquiry Practices. Our research has found that ECS inquiry teaching is *guided* and its success depends on skillful teacher designing, facilitating, and assessing learning opportunities for *active student learning*. The inquiry practices that we identified in ECS classrooms are:

- Focusing on the problem-solving process instead of only emphasizing the "right" answer, recognizing that there can be multiple solutions to a problem.
- Posing initial questions and prompts that help facilitate cognitively challenging thinking and exploration opportunities.
- Engaging students with hands-on activities so students apply and test what they know and what they are discovering.
- Encouraging exploration, autonomy, risk-taking, and creativity by resisting "giving" students the answers and immediate solutions.
- Promoting collaboration through peer-to-peer learning, small group

We observed how changing from a direct instruction teaching philosophy to an inquiry- and equity-based teaching philosophy takes time.

work, and in-depth whole class discussions.

- Connecting computer science concepts to students' prior knowledge.
- Employing journal writing, sometimes as a tool for metacognitive reflection.

Equity Practices. Especially in computer science, and other fields that suffer from underrepresentation of females, students of color, and students with disabilities, an array of equitable teaching practices are necessary to make the classroom a welcoming and enriching learning environment for all students. The equity-based practices identified in the ECS classrooms include:

- Using culturally responsive and student-centered teaching that makes computer science learning relevant to students' personal experiences and out-of-school knowledge.
- Incorporating students' cultures and out-of-school knowledge as assets instead of deficits.
- Connecting classroom learning to the sociopolitical contexts and issues relevant to students and their communities.
- Developing caring and respectful relationships with students.
- Engaging in ongoing reflection about their own and students' belief systems about who can excel in computer science. It is not uncommon for students and teachers alike to enter a computer science classroom with stereotypical notions about who will enjoy and/or do well in the course.
- Maintaining high expectations for all students that counter stereotypes about who should excel in computer science.
- Differentiating learning for diverse learning styles, English language learners, and students with disabilities.
- Creating opportunities for students to broaden participation in computing outside of the classroom through internships, community college courses, and summer programs.

These findings support research on science learning for traditionally underrepresented students that shows how engagement with the material is facilitated and learning is deepened when the practices of the field are recreated in "locally meaningful ways" and the field is presented in a way that "allows youth to express who they are and want to be in ways that meaningfully blends their social worlds with the world of science."[3]

These empirical findings also contribute to the framework of culturally responsive computing education as outlined by Eglash, Gilbert, and Foster recently in *Communications*.[2] The instructional design and pedagogy of these ECS classrooms, which purposefully combined cultural and computational practices, led to increased engagement and interest in computer science for students who historically have not had access to computing knowledge.

The world desperately needs diverse perspectives to be present at the design tables.

What Are the Implications for CS Educational Reform?

As there is increased recognition of the importance of K–12 computer science education, the issues associated with implementation are no longer a distant concern. In particular, the CS10K community (supporting the mission of building 10,000 U.S. teachers to teach high school computer science) is considering such questions as: When we recruit teachers to teach computer science, what are the qualities we hope to recruit? How important is content and pedagogical knowledge? Likewise, these findings have implications for teacher education and professional development planning. How do we design course pathways for pre-service computer science teachers? What should be the focus of professional development? What knowledge and skill sets are needed for facilitators of professional development? How do we best support a teacher corps strong in the most effective practices for broadening participation in computing?

In addition to the teaching practices identified here, our research has found variation in implementation of these practices within and across classrooms. We observed how changing from a direct instruction teaching philosophy (for example, lectures, individual student learning, right or wrong answers) to an inquiry- and equity-based teaching philosophy takes time. The critical supports for teachers include: a curriculum and accompanying professional development program that have inquiry and equity practices as the foundation; in-classroom coaching and mentoring; and a strong and vibrant teacher professional development learning community. See http://www.exploringcs.org.

The world desperately needs diverse perspectives to be present at the design tables. Our mission of democratizing K–12 CS knowledge requires making this subject accessible for all students, especially those who have been traditionally underrepresented in the field, whether they intend to become a computer scientist or not. As most professions and fields are now being transformed by computer science, students who have this knowledge have a jump start into multiple careers or academic pathways. These opportunities and contributions must not be reserved for only a narrow band of students with "preparatory privilege" that includes family resources, parental knowledge, and learning venues. This is why we advocate for computer science learning for all students. C

References

1. Calabrese Barton, A. and Tan, E. We be burnin'! Agency, identity, and science learning. *The Journal of the Learning Sciences 19*, 2 (Feb. 2010), 187–229.
2. Eglash, R., Gilbert, J., and Foster, E. Toward culturally responsive computing education. *Commun. ACM 56*, 7 (July 2013), 33–36.
3. Margolis, J., Estrella, R., Goode, J., Holme, J., and Nao, K. *Stuck in the Shallow End: Education, Race, and Computing.* MIT Press, Cambridge, MA, 2008.
4. The College Board. *California Summary of AP Program Participation and Peformance.* (2013); http://research.collegeboard.org/programs/ap/data/participation/2013.

Jane Margolis (margolis@gseis.ucla.edu) is a senior researcher at UCLA's Graduate School of Education and Information Studies.

Joanna Goode (goodej@uoregon.edu) is an associate professor of education at the University of Oregon.

Gail Chapman (chapgail@gmail.com) is the Exploring Computer Science Director of National Outreach.

Jean J. Ryoo (jryoo@exploratorium.edu) is a postdoctoral researcher with ECS and is currently project director at the SF Exploratorium.

DOI:10.1145/2618109

Batya Friedman

Viewpoint Structural Challenges and the Need to Adapt

Broadening the conversation about scholars and scholarship in computing and information research.

PHOTOGRAPH BY ZHANG XIANGYANG

At key junctures in the course of a field's evolution adjustments may be needed to stimulate and sustain rich, vital scholarship. In this Viewpoint, I will argue that the field of computing and information research is at just such a juncture and that structural changes are needed to ensure the field's ongoing health. Recently, *Communications* contributors and others have engaged in a discussion around issues related to the publication culture in computing research and its effects on the field.[1,3,6] That discussion responds in part to a shift in the late 1990s within the main computing archival publication format away from journal publications with variable-page lengths, rolling submissions, and multiple review cycles to conference proceedings with typically 10–15 page limits, set deadlines, and minimal review cycles.[4] Here, I seek to broaden the conversation to one that foregrounds the question: How do we—as a field and as individual researchers—create robust new scientific and engineering knowledge? Related conversations that concern depth and rigor of scholarship, individual career trajectories, publication, authorship norms, reporting of primary data and results, creation and deployment of artifacts, evaluation criteria, and others, follow from this central question.

To place my comments in perspective, step back to consider a few general observations about the development of any young field in relation to the development of the intellectual lifespans of individual scholars within that field. A new field by definition emerges and takes shape out of one or more existing fields.[a] At the onset, there are likely a host of new research questions and opportunities. Early in a field's history, there may be relatively little prior work to build directly upon; and more work is likely to be the first of its kind. The first scholars in an emerging field, even as they seek to create the early canonical work, often bring an interdisciplinary orientation to their thought. In such cases, they came from and were trained in other fields and bring diverse ways of asking scientific questions and doing scientific work. Over time the infusion of interdisciplinary perspectives can dissipate as those who created the field build the first departments, train the next generation of researchers, and award Ph.D.'s to inddividuals who, in turn, train the second generation of young researchers, and so on. The movement often tends away from an interdisciplinary orientation

[a] For computing and information research this transition occurred during the 1950s and 1960s stemming from the fields of electrical engineering, information theory, mathematics, and so on.

and toward developing the new field's distinctive culture and norms.

As a field continues to mature and research accumulates, the need for synthetic, integrative activities emerges. Substantive contributions that are truly novel may be less frequent and may require even greater ingenuity. The sequence described here is not uncommon for young fields and computing and information research is no exception. These and other factors converge in important and complex ways in our field such that the time is right to revisit some of the processes and norms that have evolved and consider adjustments. Such adjustments shape and enable continued strong growth.

Toward that end, in this Viewpoint I articulate seven structural challenges to the field's vitality and capacity for knowledge creation, and point briefly to the potential for practices and incentives within the field to act as constructive forces while simultaneously carefully attending to managing the transition, particularly where the careers of young scholars could be at risk.

Seven Structural Challenges to Vibrant Scholarship

Each of the structural challenges hypothesized and discussed here bear in important ways on the kinds of scientific questions that we ask as well as on the kind of research we conduct and report in response to those questions.

1. Building on Prior Work. Knowledge advances in part by building upon, extending, or reacting against earlier ideas. The field profits when researchers acknowledge and make explicit how their new ideas and findings stand in relation to what previously was understood. Doing so typically requires time and space. Time to think through those connections and space in publication venues to report on them. A significant majority of subfields in computing and information science and engineering currently place an emphasis on conference proceedings with comparatively short papers as the predominant publication venue. One significant consequence of this emphasis: there simply may not be enough space to report substantively on how one built on prior work. Given the constraints of page length, priority reasonably must be for reporting the contributions and results of the new work. Furthermore, and in part because research results typically are reported in small(er) pieces, finding let alone exploring and developing the linkages across problem domains and subfields may be difficult if not impossible. At the individual level, researchers are no longer as readily accountable for the hard work of making and elaborating on explicit connections between prior scholarship and the ways in which their new contributions complement, extend, differ from, or challenge it. At the field level, continuity and coherence of research suffers. Over time, the larger development of knowledge may slip from view.

As a field continues to mature and research accumulates, the need for synthetic, integrative activities emerges.

2. Methodological Robustness. Research findings and outcomes have meaning in context: when we know how and under what conditions they were generated. Only then are we positioned as individual scholars and as a field to judge their novelty, rigor, and, of equal importance, their limitations. Here, too, shorter publication lengths may have taken their toll. The robust reporting of method requires space to substantively convey enough details so that reviewers can evaluate the work's appropriateness and quality; near-term readers of publications can understand the results in context in order to use, apply, replicate, extend, or refute them; and readers in a farther future can understand the historical development of methods and ideas within the field. All of the above are at risk. Moreover, and perhaps most troubling, as researchers have implicitly become less accountable for reporting method robustly, there can be a corresponding tendency for methodological approaches to become less rigorous.

3. Scope and Depth of Research Questions. A field defines itself by the questions that it asks. Here, again, the current publication culture can be felt. Whereas the relatively fast conference publication cycle of shorter papers could lead to more concise reporting of results, it may also lead researchers to pose questions that can be answered more readily within 15 pages or less.[b] Such papers may also fare better in the review process as they can be contained and dealt with more fully within the page limits. Granted, some large(r) meaningful questions and results can be expressed and defended concisely, particularly in those subfields that express ideas in precise mathematical notation. Still other meaningful questions are of smaller scope. These should be pursued. However, the undue broader influence of the current conference publication cycle on the scope and depth of research questions needs to be investigated and addressed.

4. Synthesis and Theory Building. Synthesis and theory building[c] are two tools for making sense of vast amounts of individual units of knowledge and, reciprocally, providing direction for subsequent investigations. They provide a means for taking up big(ger) ideas, working them out in large(r) ways, and sustained intellectual dialogue. While valuable at any point in a field's development, these tools become essential as a field matures, subdivides, and accumulates large numbers of seemingly unconnected research findings. For example, both in human-computer interaction and in computer security there are literally hundreds of studies that engage privacy on some dimension; yet we see few analyses that bring these literatures together and offer overarching interpretations and synthesis of the results. In some subfields of computing and information research there are few out-

[b] There is more to be said about the impact of publication pace on the scope and depth of research (including the conference publication cycle and the duration and expectations of industry summer internships), however, given space limitations that discussion is beyond the scope of this Viewpoint.

[c] I use the terms "theory" and "theory building" broadly to refer to a wide range of mathematical and social scientific activity.

lets for such work. Moreover and again depending on subfield, such contributions tend to carry limited weight in the tenure and promotion process. The skills to produce these sorts of quality components take sustained effort to develop, are rarely taught explicitly to younger scholars, and currently often go largely unrewarded.

5. Interdisciplinary Work: Authorship Norms and Reporting of Primary Results. Mature and maturing fields require a steady influx of new ideas to sustain innovation and vitality. Interdisciplinary work is one such wellspring. At intellectual boundaries, established constructs encounter alternative paradigms and familiar problems, new methods, and tools, from which, in turn, new problems (and solutions) emerge. More generally, diverse ways of thinking inspire creativity and imagination. That said, conducting interdisciplinary work is not without its challenges. Two, among the many, pertain to publishing interdisciplinary work and are notable for the ethical quandaries into which they can inadvertently thrust researchers: authorship norms and publication of primary results. With regard to authorship, diverse fields (and even subfields) may have distinct and at times conflicting norms for assigning credit for intellectual contribution.[d] Given such conflicts, restructuring authorship norms for interdisciplinary work is critical if interdisciplinary work and the scholars who undertake it are to be positioned to thrive.

A similar challenge concerns the publication of primary results. Current ethical conventions around scientific publishing prohibit publishing primary results more than once. Those engaged in interdisciplinary work must choose either between violating that ethical norm or inadequately publishing to all the relevant research communities the primary results. Moreover, the latter choice means that over time, inadvertent and unintentional biases could emerge such that some fields are not directly made aware or given access to primary results within the expected communication (for example, publication) venues of their field. From the perspective of those research communities, it can be as if the research was not conducted at all. There, too, lies a certain sort of ethical failing. Underlying both of these challenges are our understandings of ethical scientific practice and the fact that with interdisciplinary work done well there is no meaningful way to speak of a primary discipline. To be sure, solutions to these challenges must reach beyond the computing and information research community into the other sciences and potentially further.

6. Solid Scholarship and Rarity of Innovation. By and large, most research in mature fields contributes incremental new knowledge that fills in gaps and tests the boundaries of established ideas. What we might call interesting solid work. Every now and then, truly new ideas are advanced that result in paradigm shifts, question previously accepted foundational knowledge, or provide radically new ways of thinking. This work is truly novel and innovative. And, if we are honest, quite rare. Both types of activities are important and complementary. That said, hiring and tenure committees at first tier universities by and large privilege novelty. In effect, these committees ask young scholars: "What new thing have you done? What new field (or subfield or sub-subfield) have you created?" Such criteria have at least two unintended consequences: they both encourage or even reward researchers who take familiar ideas adjusted slightly and rename them as new ideas, and they undervalue or even penalize researchers who build in deep ways on prior work and appropriately acknowledge that intellectual legacy. Occasionally, of course, there will be entirely new inventions and significant breakthroughs, and these will need to be recognized as such.

7. Growth of a Scholar. Not only does a field mature but so, too, do the individual scholars whose efforts lead

[d] To clarify the dilemma, consider this real example: in one field the authorship expectation is "students first, followed by faculty" (as is common in some sub-fields in computing and information research) and in another field "the order of authorship credit should accurately reflect the relative contributions of persons involved" (as is the rule in psychology[5]); given a faculty member who is the intellectual lead for an interdisciplinary team comprised of students and faculty from different disciplines, appropriately following one field's norm for authorship order necessarily would violate that of the other, and vice versa.

As with any dynamic ecosystem, the computing and information research field does not have the possibility of remaining static.

to that field's ongoing vitality and development. Moreover, experience matters. Yet, I observe that—through complex forces and pressures within the field's ecosystem—less experienced researchers often prematurely take on key intellectual leadership roles (for example, graduate students now typically present the majority of papers at conferences and in some subfields act as reviewers for major conferences and journals), and increasingly frame research questions and even major research programs. My point is not to suggest disallowing any of the activities mentioned here for less-experienced researchers but that currently the balance is skewed. In short, we fail to adequately nurture the less-experienced researcher or to provide time and space for intellectual maturation; and correspondingly we erode opportunities for the type of substantive intellectual contributions that can come from more seasoned researchers.

Cultivating Adaptations: Toward a (More) Vibrant Scholarly Ecosystem

Exactly how to adapt current practices and incentives in the field to address the structural issues identified in this Viewpoint remains an open question. At a minimum, we can expect to engage the norms and expectations that underlie research and scholarship. For example, the current trend to produce many small(er) publications could be reversed with policies and incentives that reward a smaller number of stronger publications. Indeed, and in part in response to these and related issues, in the U.S. we have seen a recent policy change at the National Science Foundation that now limits the number of proposal submissions for some programs to two per year for each Principal Investigator. This policy and others like it could serve to incentivize writing a smaller number of stronger proposals. Other practices and incentives will need to value building substantively on prior work; recognize solid intellectual development (without requiring all or even most researchers to invent new subfields or coin new terms); and reward synthesis and theory building. For interdisciplinary work, we will need to rethink and clarify norms for crediting intellectual contribution and authorship as well as for the primary publication of results. Critical to all of this is the need to revisit the balance and distribution of activities among more and less experienced researchers. Of course, each of these will need to be carefully thought through and debated within the community (and for those aspects tied to interdisciplinary research within the broader scientific community).

Biologists warn us that in shifting ecosystems, those who were well adapted to one environment may be at risk in another.[2] So, too, when there are significant shifts in a social ecosystem, such as those of the scope advocated for here. Even as we adapt norms, expectations, and practices to sustain and continue to evolve the field, we will need to attend to the careers of talented young researchers who will need to navigate that transition. In particular, the kind of changes discussed here will have teeth when hiring and promotion and tenure committees correspondingly shift their evaluation criteria to, for example, emphasize a smaller coherent body of more substantive publications. Thus, faculty mentoring Ph.D. students and new Ph.D.'s as well as hiring committees and tenure and promotion committees will need to be alert to the transition and its implications for the scholars they are mentoring and evaluating.

As with any dynamic ecosystem, the computing and information research field does not have the possibility of remaining static. In the normal course of events, as some aspects of the field change—with the accumulation of new knowledge, training of younger researchers, and shifts in publication modes—others will need to be adjusted in response. Moreover, at key junctures dynamic systems may require critical, intentional adjustments to ensure their ongoing viability and vibrancy. I have argued that this is just such a moment for the field, if we are to ensure the field's ongoing ability to generate new, transformative knowledge; ensure deep scholarship; and sustain impact. That said, perturbing any functioning ecosystem is risky business. A small adjustment in one area may have far reaching effects, some or many of which may be unanticipated. Thus, changes will need to be considered carefully, managed over time, and refined (readjusted) as they unfold. As a community, we must proceed both boldly to ensure great scholarship and continuing impact and with alertness so as to minimize harm to the next generation that will carry the field forward.

Admittedly, this Viewpoint is just that. Each of the challenges I have articulated could be the subject of a serious, deep analysis. That would be an excellent next step. Moreover, I have refrained from suggesting specific solutions, as I believe those need to come from the community as a result of thoughtful process and debate. It is my hope this Viewpoint continues and deepens the conversation about these and related issues.

References

1. Birman, K. and Schneider, F.B. Program committee overload in systems. *Commun. ACM 52*, 5 (May 2009), 34–37.
2. Dubos, R. *Man Adapting* (enlarged edition). Yale University Press, New Haven, CT and London, England. 1965, 1980.
3. Grudin, J. Technology, conferences, and community. *Commun. ACM 54*, 2 (Feb. 2011), 41–43.
4. Patterson, D., Snyder, L., and Ullman, J. Best practices memo: Evaluating computer scientists and engineers for promotion and tenure. *Computing Research News* (Aug. 1999); http://cra.org/resources/bp-view/evaluating_computer_scientists_and_engineers_for_promotion_and_tenure/.
5. *Publication Manual of the American Psychological Association* (Sixth Edition). American Psychological Association, Washington, D.C., 2010.
6. Vardi, M.Y. Revisiting the publication culture in computing research. *Commun. ACM 53*, 5 (May 2010).

Batya Friedman (batya@uw.edu) is a professor in the Information School, an adjunct professor in the Department of Computer Science and Engineering, and an adjunct professor in the Department of Human-Centered Design and Engineering at the University of Washington where she co-directs the Value Sensitive Design Research Lab and co-directs the UW Tech Policy Lab.

Thanks to Lynette Millett at the National Research Council and to the participants of the Dagstuhl Workshop on the Publication Culture in Computing Research for extended conversations on this topic. Thanks, too, to Jon Eisenberg, Jonathan Grudin, Fred Schneider, Jacob Wobbrock, Ellen Zegura, and three anonymous reviewers for comments on this Viewpoint.

DOI:10.1145/2618111

Phillip A. Laplante

Viewpoint
Licensing Professional Software Engineers: Seize the Opportunity

Professional organizations should be in the forefront of the ongoing discussion about licensing professional software engineers.

In his July 2013 *Communications* column, ACM President Vint Cerf revisited the controversy of licensing professional software engineers in the U.S. This issue has been one that divides the profession since reasonable cases can be made both pro and con. Officially, ACM has opposed and IEEE has supported such licensure. Even within both organizations, though, there is substantial support and opposition.[1–3,5] I think President Cerf was right in suggesting that, because of the dramatic growth in interacting software in both devices and in applications that significantly impact peoples' lives, and more importantly, because licensing is happening, we have reached a tipping point.

I am a longtime (more than 25 years) member of both the ACM and IEEE and I chair the committee that developed and maintains the licensing exam for use by the states. I have also been involved in helping states' licensing boards deal with the issues of operationalizing the licensing process. Even so, I have been no longtime supporter of licensing software engineers—I wrote against it as recently as 2005, noting the many challenges.[5] But two things changed my mind. The first was a book about the tumultuous history of implementing medical licensing in the U.S.[8]—a history that in many ways parallels that for licensing software engineers. The second involved deep conversations with Dennis Frailey and Don Bagert, who have written in support of licensing certain software engineers for many years. I became convinced it was necessary to move forward with the licensing process and to be involved in helping address these challenges from within the community of licensed professional engineers.

I will not revisit the pros and cons of licensure—as noted, there is ample literature on that—and whatever your feelings, licensing of certain software engineers is now mandatory in 40 states in the U.S. and other states will

likely follow. Instead, I would like to briefly recap the current situation, describe some of the implementation challenges, and then suggest ACM and its members should be more involved.

The opinions expressed here are my own and represent no other entity with which I am affiliated.

Licensing of certain software engineers is now mandatory in 40 states in the U.S. and other states will likely follow.

Current Situation

States license engineers who work on systems in which failure could adversely affect the health, safety, and welfare of the public and who offer their services directly to the public.

Generally, the requirements for licensure as a professional software engineer are:

- Holding a bachelor's degree in software engineering from an ABET-accredited program.
- Passing the Fundamentals of Engineering (FE) exam.
- Having applicable work experience (typically, at least four years) under the supervision of a licensed professional engineer (PE).
- Passing the Principles and Practices of Software Engineering (P&P) exam.

In addition, evidence of good moral character is needed and most states require continuing education to retain licensure. These requirements vary between states—usually pertaining to years of education, the nature of work experience, industrial exemption rules, and alternative paths to licensure.

My volunteer committee developed the P&P exam. The development effort was supported by a consortium of non-profit entities including The National Society of Professional Engineers, The Texas Board of Professional Engineers, and IEEE (through its U.S. Board and Computer Society). The National Council of Examiners of Engineers and Surveyors (NCEES) is the non-profit organization that oversees all exam development and administration for the state boards. Several exam development committee members are ACM members, though they were acting privately. A discussion of the exam development process can be found in Laplante.[6]

In April 2013, the first Principles and Practices exam was offered. Twelve individuals took that exam and six passed the exam. It is likely the low number of examinees was because the FE exam must be passed before the P&P exam can be taken.

The FE exam comprises 180 multiple-choice questions and it is this exam that seems to cause the most resistance and fear when I discuss licensing. The morning session is the same for all engineering disciplines—120 questions in mathematics, probability and statistics, chemistry, computers, ethics, business practice, economics, mechanics, strength of materials, material properties, fluid mechanics, electricity and magnetism, and thermodynamics. Software engineers take the same afternoon exam as electrical and computer engineers. This exam comprises 60 questions covering circuits, power, electromagnetics, control systems, communications, signal processing, electronics, digital systems, and computer systems.

Most of the afternoon topics would be learned by a student in an ABET-accredited software engineering program (there are 21 in the U.S.) or computer science program. Some of the morning session topics, however, would not normally be seen by such students. Still, one can think of circumstances where the concepts of material properties, fluid mechanics, and thermodynamics would be relevant to a software engineer working on water treatment, power generation and distribution, or road and railway systems. Besides, these areas represent a small fraction of the test and can be studied independently through review courses. The FE exam was recently revised by NCEES and will continue to evolve to more accurately reflect the widening differences in the ABET engineer-

Calendar of Events

July 21–26
International Symposium on
Software Testing and Analysis,
San Jose, CA,
Sponsored: SIGSOFT,
Contact: Corina S. Pasareanu,
Phone: 650-604-4767
Email: corina.s.pasareanu@nasa.gov

July 23–25
7th ACM Conference on
Security & Privacy in
Wireless and Mobile Networks,
Oxford, UK,
Sponsored: SIGSAC,
Contact: Ivan Martinovic,
Email: ivan.martinovic@cs.ox.ac.uk

August 8–9
ACM Symposium on Applied
Perception 2014,
Vancouver, BC, Canada,
Sponsored: SIGGRAPH,
Contact: Reynold Bailey,
Email: rjb@cs.rit.edu

August 8–10
The Joint Symposium on
Computational Aesthetics and
Sketch Based Interface and
Non-Photorealistic Animation
and Rendering,
Vancouver, BC, Canada,
Sponsored: SIGGRAPH,
Contact: David Robert Mould,
Email: mould@scs.carleton.ca

August 8–10
The 19th International
Conference on
Web3D Technology,
Vancouver, BC, Canada,
Sponsored: SIGGRAPH,
Contact: Nicholas Fearing Polys,
Email: npolys@vt.edu

August 10–13
The 20th ACM SIGKDD
International Conference on
Knowledge Discovery and Data
Mining,
New York, NY,
Sponsored: SIGKDD, SIGMOD,
Contact: Claudia Perlich,
Phone: 914-409-5609
Email: Claudia.perlich@gmail.com

August 10–14
Special Interest Group on
Computer Graphics and
Interactive Techniques
Conference,
Vancouver, BC, Canada,
Sponsored: SIGGRAPH,
Contact: Dave Shreiner,
Email: shreiner@yahoo.com

ing programs. Here is an opportunity for software professionals and faculty to seek licensure and work to reform the FE exam and the ABET software engineering degree.

Alternate Paths and Industrial Exemption

Some state boards will waive certain requirements, such as those related to education and experience, or provide for alternative satisfaction. For example, many board policies allow for trading education for experience and vice versa, say, holding a relevant master's degree could count for one year of experience; a Ph.D. could count for two years of experience. Additional years of experience could mitigate holding a non-ABET accredited degree in software engineering or a related degree (such as computer science), an unrelated degree, an associate's degree, or no degree at all. In certain cases "recognized standing" can be used to petition boards for licensure or exemption from certain criteria. An opportunity exists for ACM to lobby state boards to accept a broad set of degrees (for example, computer science, mathematics, information systems) as relevant, award credit for certain professional certifications, and to introduce or expand grandfathering criteria.

Many states have "industrial exemptions" that allow the practice of engineering without licensure in such areas as electrical or telecommunications utilities, and businesses that manufacture a product. Recently, NCEES convened a group to study how states deal with these exemptions. The group found that "few states actually exempt many categories of engineers from licensure. For example, engineering faculty are specifically exempt in just a handful of jurisdictions. State and local government agencies are exempt from engineering licensure in only one jurisdiction. Public utilities are specified in only 11 jurisdictions."[7] The group further recommended that state boards be encouraged to reduce the number of exemptions. The NCEES recommendation presents a threat—narrowing the list of exemptions will increase the number of professionals who need to be licensed. But this is also an opportunity for the ACM and its members to be involved in the discussion to encourage state boards to intelligently expand the exemptions.

Ironically, I have discovered that some people who object to licensing probably do not need to be licensed.

Implementation Issues

Complex implementation issues still need to be resolved. One important issue is in defining the "penumbra," that is, to which systems should licensure statutes apply? Other questions include: how do we treat the chain of interactions, custody and responsibility for interacting systems; what roles can licensed professional engineers play; what artifacts should be sealed by the PE; and what roles can non-PE software professionals play? These questions must be answered in a comprehensive and consistent way across states. These issues existed in other licensed engineering disciplines and the answers have emerged with time and experience.

Another unresolved issue, which is common to licensing of all engineering disciplines, is international recognition. There are some agreements in place that can act as building blocks, but there is no uniformity in how state boards treat licensed professionals from other countries and how other countries recognize U.S. licensed engineers.[4] ACM and its members can help in addressing these challenges.

Going Forward

Wherever there are challenges there are opportunities. Several companies are seeking licensure of their software engineers as a competitive advantage; individuals can do the same. Given the way licensing statutes are currently interpreted, I believe a very small percentage of software professionals will have to be licensed. Most electrical and mechanical engineers are not licensed, nor need to be, yet the path to licensure exists for them.

For those who are working on systems that will be within the penumbra, licensing will be mandatory. In these cases, whatever their career starting point, I can suggest a path to licensure. It might mean taking a review course or two, or it might mean completing a degree. In some cases it might mean petitioning a board for an exemption. Ironically, I have discovered that some people who object to licensing software engineers probably do not need to be licensed, or would be able to be licensed if only they would take the tests or petition their state board for an accommodation.

State professional engineer licensing boards, which are generally populated by civil engineers, need help understanding software implementation, understanding the penumbra, developing alternative paths to licensure, creating grandfathering criteria, and refining the set of industrial exemptions. It is the difficulties in addressing these issues that are the usual basis of the case against licensing. But licensing is a reality and professional organizations such as ACM, ASQ, and IEEE, should take the lead in helping state boards answer these questions. Otherwise lawyers, lobbyists, and engineers of other disciplines will influence the rulemaking. C

References

1. Bagert, D.J. Taking the lead in licensing software engineers. *Commun. ACM 42*, 4 (Apr. 1999), 27–29.
2. Frailey, D.J. Licensing software engineers. *Commun. ACM 42*, 12 (Dec. 1999), 29–30.
3. Knight, J.C. and Leveson, N.G. Should software engineers be licensed? *Commun. ACM 45*, 11 (Nov. 2002), 87–90.
4. Laplante, P.A. An international perspective on U.S. licensure of software engineers. *Technology and Society 32*, 1 (Jan. 2013), 28, 30.
5. Laplante, P.A. Professional licensing and the social transformation of software engineers. *Technology and Society 24*, 2 (Feb. 2005), 40, 45.
6. Laplante, P.A., Kalinowski, B., and Thornton, M. A principles and practices exam specification to support software engineering licensure in the United States of America. *Software Quality Professional 15*, 1 (Jan. 2013), 4–15.
7. Launey, A.J.P. NCEES task force studies licensure exemptions. *Today's Engineer* (July 2013); http://www.todaysengineer.org/2013/Jul/licensure-July.
8. Starr, P. *The Social Transformation of American Medicine.* Basic, New York, 1982.

Phillip A. Laplante (plaplante@psu.edu) is a professor of software engineering at Penn State University.

ACM's Career & Job Center

Are you looking for your next IT job?

Do you need Career Advice?

The **ACM Career & Job Center** offers ACM members a host of career-enhancing benefits:

- A **highly targeted focus** on job opportunities in the computing industry
- **Access to hundreds** of industry job postings
- Resume posting **keeping you connected** to the employment market while letting you maintain full control over your confidential information
- **Job Alert system** that notifies you of new opportunities matching your criteria
- **Career coaching** and guidance available from trained experts dedicated to your success
- **Free access** to a content library of the best career articles compiled from hundreds of sources, and much more!

Visit ***ACM's Career & Job Center*** *at:*
http://jobs.acm.org

acm
Association for
Computing Machinery

Advancing Computing as a Science & Profession

The **ACM Career & Job Center** is the perfect place to begin searching for your next employment opportunity!

Visit today at http://jobs.acm.org

DOI:10.1145/2619238

Article development led by acmqueue
queue.acm.org

You must have some trust if you want to get anything done.

BY THOMAS WADLOW

Who Must You Trust?

IN HIS NOVEL *The Diamond Age*,[5] Neal Stephenson describes a constructed society (called a *phyle*) based on extreme trust in one's fellow members. One of the membership requirements is that, from time to time, each member is called upon to undertake tasks to reinforce that trust. For example, a phyle member might be told to go to a particular location at the top of a cliff at a specific time, where he will find bungee cords with ankle harnesses attached. The other ends of the cords trail off into the bushes. At the appointed time he is to fasten the harnesses to his ankles and jump off the cliff. He has to trust the unseen fellow phyle member who was assigned the job of securing the other end of the bungee to a stout tree actually did his job; otherwise, he will plummet to his death. A third member secretly watches to make sure the first two do not communicate in any way, relying only on trust to keep tragedy at bay.

Whom you trust, what you trust them with, and how much you trust them are at the center of the Internet today, as well as every other aspect of your technological life.

Here is an experiment to try. Take walks in various mixed-use neighborhoods with a variety of residences and businesses. Walk in the daytime, before and after lunch. Walk in the nighttime, at the height of the evening activities. Walk late at night, after most things have shut down. With each outing, put yourself in a security mindset—which is to say, look with the eyes of a thief and notice what you see.

During the day, for example, at busy sidewalk cafes, do people reserve outdoor tables by placing their possessions on the table and then going inside to order? Do they use their grocery bags for this? Their car and house keys? Their wallets?

Late at night, are those same tables and chairs stacked outside or inside? Are they chained together? Are the chains lightweight or substantial?

Do the neighborhood home have porch furniture or lawn tools visible from the street? Are they locked up? Do you see bars on the windows of the homes? Are the family cars parked outside? Do they have steering-wheel locks?

Do postal workers or delivery services leave packages unattended by the front doors of houses? Are bundles of newspapers and magazines left in front of newsstands before they open?

These observations, and many more, are flags for the implicit levels of trust that people have in their neighbors and neighborhoods. The people themselves may not even think of these things. They may leave things on their porches, perhaps accidentally, and nothing bad happens, so they do not worry if it happens again. After a while, it becomes something they do not even notice they do.

ILLUSTRATION BY MICHAEL GLENWOOD

m.glenwood

Heartbleed

In spring 2014, a bug in the open source package OpenSSL became widely known. Known as Heartbleed (http://heartbleed.com), the bug had been present for some time, and may have been known by some, but the full disclosure of the problem in the OpenSSL package came to the public's attention only recently. OpenSSL had been reviewed by many experts and had been a well-used and trusted part of the Internet ecosystem until that point. As of this writing, there is no evidence suggesting any cause other than a programming error on the part of an OpenSSL contributor.

On the morning before the Heartbleed bug was made public, few people were familiar with OpenSSL and hardly gave the functions it provided a second thought. Those who knew of it often had a strong level of trust in it. By the end of the day, that had all changed. Systems administrators and companies of all sizes were scrambling to contain the problem. In just a few days, this obscure piece of specialized software was at the top of the news cycle, and strangers—sitting in outdoor cafes at tables they had reserved with their house and car keys—were discussing it in the same tones with which they might have discussed other catastrophes.

Systems Administrators

At the heart of everything that works on the Internet are systems administrators. Sometimes they are skilled experts, sometimes low paid and poorly trained, sometimes volunteers of known or unknown provenance. Often they work long, unappreciated hours fixing problems behind the scenes or ones that are all too visible. They have access to systems that goes beyond that of regular users.

One such systems administrator worked for the National Security Agency. His name is Edward Snowden. You probably know more about him now than you ever expected to know about any sys admin, even if you are one yourself.

Another less familiar name is Terry Childs,[3,7] a network administrator for the city of San Francisco, who was arrested in 2008 for refusing to divulge the administrative passwords for the city's FiberWAN network.

Whatever components or services you choose, consider how they have been tested for trustworthiness.

This network formed the core of many city services. According to reports, Childs, a highly qualified and certified network engineer, was very possessive of the city's network, having designed and implemented much of it himself—perhaps too possessive, as he became the sole administrator of the network, claiming not to trust his colleagues' abilities. He allowed himself to be on-call 24/7, year-round, rather than delegate access to those he considered less qualified.

After an argument with a new boss who wanted to audit the network against Childs' wishes, the city's CIO demanded that Childs provide the administrative credentials to the FiberWAN. Childs refused, which led to his arrest. His supervisors claimed he was crazy and wanted to damage the network. Childs claimed he did not want to provide sensitive access credentials to unqualified individuals who might damage "his" network.

In 2010, Childs was found guilty of felony network tampering and sentenced to four years in prison and $1.5 million in restitution for the costs the city incurred in regaining control of the network. An appeals court upheld the verdict.

Was Childs a fanatic, holding on too tight for his own good, or a highly responsible network admin who would not allow his network to be mismanaged by people he considered to be incompetent? His case brings up these questions:

- Could something like this happen at your enterprise? How would you know this problem was developing, *before* it became a serious problem?
- What safeguards do you have in place to prevent a single-point concentration of power like this?
- What would you do if your organization found itself in this situation?

Who Must You Trust?

Some people dream of going back to nature and living apart from the rest of humanity. They will build their own cabins, grow or raise their own food, and live entirely off infrastructure they have built with their own two hands and a trusty axe. But who made that axe? Even if you can make a hand-chipped flint axe from local materials, it is far from "trusty," and the amount of wood you can cut with a flint axe pales in comparison to what you can cut with a modern steel axe. So if you go into the woods with a modern axe, can you truly say you are independent of the world?

If you work on the Internet, or provide some service to the Internet, you have a similar problem. You cannot write all of the code if you intend to provide a modern and useful network service. Network stacks, disk drivers, Web servers, schedulers, interrupt handlers, operating systems, compilers, software-development environments, and all the other layers needed to run even a simple Web server have evolved over many years. To reinvent it all from the specifications, without using other people's code anywhere in the process, is not a task for the fainthearted. More importantly, you could not trust it completely even if you did write it all. You would be forever testing and fixing bugs before you were able to serve a single packet, let alone a simple Web page.

Neither can you build all of the hardware you run that service on. The layers of tools needed to build even a simple transistor are daunting, let alone the layers on top of that needed to build a microprocessor. Nor can you can build your own Internet to host it. You have to trust some of the infrastructure necessary to provide that service. But which pieces?

How Much Do You Need to Trust?

To determine how far your trust needs to extend, start with an evaluation of your service and the consequences of compromise. Any interesting service will provide some value to its users. Many services provide some value to their providers. What is valuable about your service, and how could that value be compromised?

Once you have a handle on these questions, you can begin to think about the minimum of components and services needed to provide such a service and which components you have to trust.

Writing your own software can be part of this exercise, but consider the bulk of security derived from that is what is known as "security through obscurity." Attacks will fail because attackers do not understand the code you have built—or so some think. If you choose the path of obscurity as a strategy, you are betting that no one will show interest in attacking your service, that your programmers are better than others at writing obscure code in a novel way, and that even if the code is obscure, it will still be secure enough that someone determined to break through it will be thwarted. History has shown these are not good bets to make.

Who Is Everybody Else Trusting?

A better approach might be to survey the field to see what others in similar positions are doing. After all, if most of your competitors trust a particular software package to be secure, then you are all in the same situation if it fails. There are variables, of course, because any software, even the best, can be untrustworthy if it is badly installed or configured. Furthermore, your competitors might be mistaken.

A variation on this approach is to find out which software all of your competitors *wish* they could use. Moving to what they use now could leave you one generation behind by the time you get it operational. On the other hand, moving to one generation ahead could leave you open to yet-undetected flaws. The skill is in choosing wisely.

How Are Services Evaluated for Security?

Whatever components or services you choose, consider how they have been tested for trustworthiness. Consider these principles attributed to Auguste Kerckhoffs, a Dutch linguist and cryptographer, in the 19th century:

- The system should be, if not theoretically unbreakable, then unbreakable in practice.
- The design of a system should not require secrecy, and compromise of the system design should not inconvenience the correspondents.

Kerckhoffs was speaking of cipher design in cryptosystems, but his two principles listed here can be applied to many security issues.

When considering components for your enterprise, you should ask if they live up to Kerckhoffs' principles. If they seem to, who says that they do? This is one of the strongest cases for open-source software. When done properly, the quality and security of open-source code can rival that of proprietary code.[2]

For services you wish to subscribe to, consider how often and how thoroughly they are audited, and who conducts the audit. Do the service providers publish the results? Do they allow prospective customers to see the results? Do the results show their flaws and describe how they were fixed or remediated, or do they just give an overall thumbs-up?

Thinking About the Bad Cases First

The legendary Fred Brooks, he of *The Mythical Man Month*,[1] famously said: "All programmers are optimists." Brooks meant this in terms of the tendency of programmers to think they can complete a project faster than it will actually take them to do so. But as *Communications*' own Kode Vicious is wont to point out, there is a security implication here as well. Developers often code the cases they want to work first and, if there is enough time, fill in the error-handling code later, if at all.

When you are worried about security issues, however, reversing the order of those operations makes a lot of sense. If, for example, your application requires a cryptographic certificate to operate, one of the first issues a security programmer should think about is how that certificate can be revoked and replaced. Selecting certificate vendors from that perspective may be a very different proposition from the usual criteria (which is almost always cost). Building agile infrastructure from the start, in which the replacement of a crypto cert is straightforward, easy to do, and of minimal consequence to the end user, points the way toward a process for minimizing trust in any one vendor.

Developing an infrastructure that makes it easy to swap out certificates leads to the next interesting question: How will you know when to swap out that bad certificate? Perhaps the question can be turned around: How expensive is it to swap out a certificate—in money, effort, and customer displeasure? If it can be done cheaply, quickly, easily, and with no customer notice, perhaps it should be done frequently, just in case. If done properly, a frequent certificate change would help limit the scope of any damage, even if a problem is not noticed at first.

But there be dragons here! Some might read the previous paragraph and think that having certificates that expire weekly, for example, eliminates the need to monitor the infrastructure for problems, or the need to revoke a bad certificate. Far from it! All of those steps are necessary as well. Security is a belt-and-suspenders world.

An infrastructure that is well monitored for known threats is another part of the trust equation. If you are confident your infrastructure and personnel will make you aware of certain types of problems (or potential problems), then you can develop and practice procedures for handling those problems.

That covers the "known unknowns," as former U.S. Secretary of Defense Donald Rumsfeld[4] said, but what about the "unknown unknowns"? For several years Heartbleed was one of these. The fault in OpenSSL was present and exploitable for those who knew of it and knew how to do so. As of this writing, we do not know for certain if anybody did exploit it, but had someone done so, the nature of the flaw is such that an exploit would have left little or no trace, so it is very difficult to know for sure.

There are two major kinds of "unknown unknowns" to be aware of when providing a network service. The first are those unknowns you do not know about, but somebody else might know about and have disclosed or discussed publicly. Let's call them "discoverable unknowns." You do not know about them now, but you can learn about them, either from experience or from the experiences of others.

Discoverable unknowns are discoverable if and only if you make the effort to discover them. The pragmatic way to do this is to create an "intelligence service" of your own. The Internet is full of security resources if you care to use them. It is also full of misdirection, exaggeration, and egotism about security issues. The trick is learning which resources are gold and which are fool's gold. That comes with practice and, sadly, often at the cost of mistakes both big and small.

A prudent, proactive organization has staff and budget devoted to acquiring and cultivating security resources. These include someone to evaluate likely websites, as well as read them regularly; subscriptions to information services; membership in security organizations; travel to conferences; and general cultivation of good contacts. It also includes doing favors for other organizations in similar situations and, if possible, becoming a good citizen and participant in the open-source world. If you help your friends, they will often help you when you need it.

The second type of unknowns can be called "unexpected unknowns." You do not know what they are, you do not even know for sure that they exist, and you are not on the lookout for them specifically. But you can be on the lookout for them in general, by watching the behavior of your network. If you have a way of learning the baseline behavior of your network, system, or application, you can compare that baseline to what the system is doing now. This could include monitoring servers for unexpected processes, unexpected checksums of key software, files being created in unusual places, unexpected load changes, unexpected network or disk activity, failed attempts to execute privileged programs, or successful attempts that are out of the ordinary. For a network, you might look for unusual protocols, unexpected source or destination IP addresses, or unusually high- or low-traffic profiles. The better you can characterize what your system is supposed to be doing, the more easily you can detect when it is doing something else.

Detecting an anomaly is one thing, but following up on what you have detected is at least as important. In the early days of the Internet, Clifford Stoll,[6] then a graduate student at Lawrence Berkeley Laboratories in California, noticed a 75-cent accounting error on some computer systems he was managing. Many would have ignored it, but it bothered him enough to track it down. That investigation led, step by step, to the discovery of an attacker named Markus Hess, who was arrested, tried, and convicted of espionage and selling information to the Soviet KGB.

Unexpected unknowns might be found, if they can be found at all, by reactive means. Anomalies must be noticed, tracked down, and explained. Logs must be read and understood. But defenses against known attacks can also prevent surprises from unknown ones. Minimizing the "attack surface" of a network also minimizes the opportunities an attacker has for compromise. Compartmentalization of networks and close characterization of regular traffic patterns can help detect something out of the ordinary.

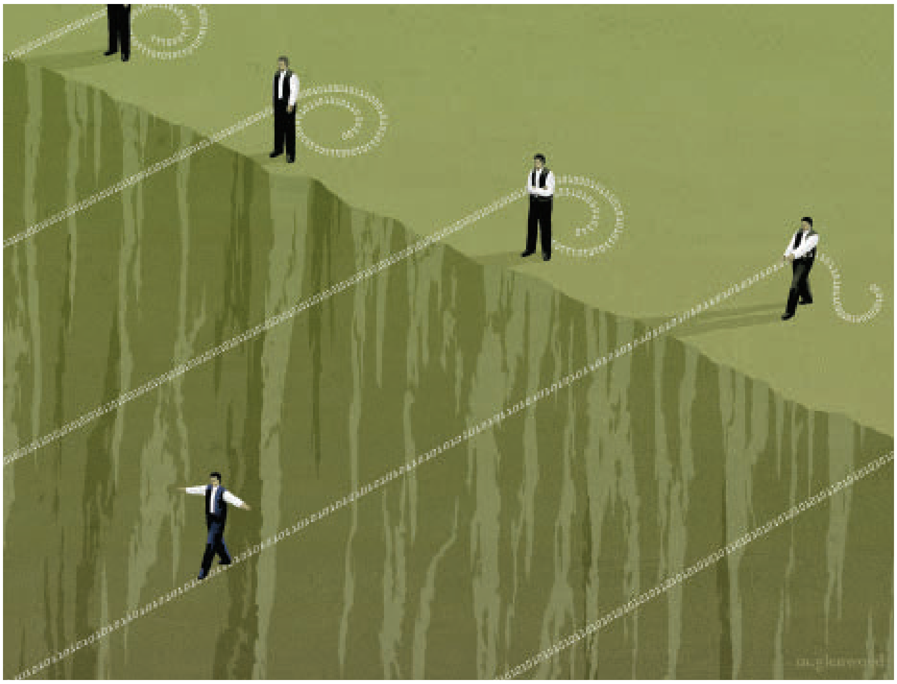


ILLUSTRATION BY MICHAEL GLENWOOD

What Can You Do?

How can issues of trust be managed in a commercial, academic, or industrial computing environment?

The single most important thing a practitioner can do is to give up the idea that this task will ever be completed. There is no device to buy, no software to install, and no protocol to implement that will be a universal answer for all of your trust and security requirements. There will never come a time when you will be done with it and can move on to something else.

Security is a process. It is a martial art you can learn to apply by study, thought, and constant practice. If you do not drill and practice regularly, you will get rusty at it, and it will not serve you when you need it. Even if you do become expert at it, an attacker may sometimes overpower you. The better you get at the process, however, the smaller the number of opponents that can do you harm, the less damage they can do, and the quicker you can recover.

Here are some basic areas where you can apply your efforts.

Know whom you trust and what you trust them to do. Though it is an overused term, "Web of Trust" is descriptive of what you are building. Like any sophisticated construction, you should have a plan, a diagram, or some other form of enumeration of which trust mechanisms are needed to support your enterprise. The following entities might be on such a plan: datacenter provider (power, A/C, LAN); telecommunications link vendors; hardware vendors; paid software vendors; open source software providers; cryptographic certificate suppliers; time-source suppliers; systems administrators; database administrators; applications administrators; applications programmers; applications designers; security engineers.

Of course, mileage may vary, and there may be many more entities as well. Whatever is on the list you generate, perform the following exercise for each entry:

- Determine whom this entity trusts to do the job and who trusts this entity.
- Estimate the consequences if this entity were to fail to do the job properly.
- Estimate the consequences if this entity were a bad actor trying to compromise the enterprise in some way (extract information without authorization, deny service, provide bad information to your customers or yourself, and so on).
- Rate each consequence for severity.

Know what you would do if any of those entities lost your trust. Now that you have a collection of possible ways that your enterprise can be affected, sorted by severity, you can figure out what you would do for each item. This can be as simple or complicated as you are comfortable with, but remember that you are creating a key part of your operations handbook, so if your plans cannot be turned into actions when these circumstances occur, they will not be worth much.

Developing an infrastructure that makes it easy to swap out certificates leads to the next interesting question: How will you know when to swap out that bad certificate?

Here are some examples of the kinds of consequences and actions that might be needed:

- A key open-source package is discovered to have a serious bug and must be: replaced with a newer, bug-fixed version; replaced with a different package with the same API; replaced with a different package with a different API; or mitigated until a fix can be developed. Your plan should be a good guide to handling any of those situations.
- A key systems administrator has been providing network access to a potentially unfriendly third party. You must: determine the extent of information lost (or was your information modified?); determine if any systems were compromised with backdoor access; determine which other systems under the sys admin might be affected; figure out the best way of handling the personnel issues (firing, transfer, legal action).
- A key data center is rendered unusable by a disaster or attack. You must: shift to a standby reserve location; or improvise a backup datacenter.

Practice, practice, practice. Having a plan is all very nice, but if it is in a dusty file cabinet, or worse yet, on a storage volume in a machine that is made unavailable by the very circumstances you are planning for, then it does not help anybody. Even if the plan is readily available, carrying it out for the first time during a crisis is a good way to ensure it will not work.

The best way to make sure that your plan is actionable is to practice. That means every plan needs to have a method of simulation of cause and evaluation of result. Sometimes that can be as easy as turning off a redundant server and verifying that service continues. Others are more complex to simulate. Even a tabletop exercise, in which people just talk about what is needed, is better than never practicing your contingency plan.

Practice can also take the form of regular operations. For example, Heartbleed required many service providers to revoke and reissue certificates. If that is a critical recovery operation for your enterprise, then find a way to work that procedure into your regular course of business, perhaps by revoking and reissuing a certificate once a month.

Other operations can also benefit from practice, such as restoring a file from backups; rebuilding an important server; transferring operations to a backup datacenter; or verifying the availability of backup power and your ability to switch over to it.

Set mousetraps. The most important step in defending against attackers (or Murphy's Law) is acquiring the knowledge that you have a problem. If you understand your trust relationships—who is trusted with what and who is not trusted—then watching for violations of those relationships will be very instructive. Every violation will probably fall into one of these categories:

- An undocumented but legitimate trust relationship. This might be sys admins doing their assigned work, for example, but that work was improperly overlooked when building the trust map.
- A potentially reasonable but unconsidered potential trust relationship that must be evaluated and either added to the trust map or explicitly prohibited—for example, a sys admin doing unassigned but necessary work to keep a system operational.
- An unreasonable or illegitimate use.

The only way to know which case it is will be to investigate each one and modify your trust map accordingly. As with all things of this nature, mousetraps must be periodically tested to see if they still work.

Vet your key people. Trusting a systems administrator often takes the form of management saying to sys admins, "Here are the keys to everything," followed by more-or-less blind trust that those keys would not be abused. Or to quote science fiction author Robert Heinlein: "It's amazing how much mature wisdom resembles being too tired." That sort of blind trust is asking for trouble.

On the other hand, tracking sys admins closely and forcing them to ask permission for every privileged operation they wish to perform can hobble an organization. Chances are good that both the sys admins and the granters of permission will grow tired of this and the organization will move back toward blind trust.

Having a plan is all very nice, but if it is in a dusty file cabinet or on a storage volume in a machine made unavailable by the very circumstances you are planning for, then it does not help anybody.

A good way to navigate between these two rocky shoals is to hire good people and treat them well. Almost as important is communicating with them to reinforce the security and trust goals of your organization. If they know what must and must not be done and, at least in general principle, why those constraints are good, then the chances are greater they will act appropriately in a crunch.

Log what they do. Have somebody else review those logs regularly. Good people can make mistakes and sometimes even go astray. A regular nonprivileged (in the security sense) employee should still have a reasonable expectation of workplace privacy, but a systems administrator should know he or she is being watched when performing sensitive tasks or accessing sensitive resources. In addition, encourage sys admins to perform extremely sensitive tasks with at least one other person of equal or higher clearance present. That way, someone else can attest the action taken was necessary and reasonable.

Wherever possible, log what the sys admins do with their privileges and have a third party review those logs regularly for anomalies. The third party should be distant enough from the systems administrators or other employees given trusted access so that no personal or professional relationships will obscure the interpretation of the logs.

Investigate what you suspect and act on what you find. Let your trusted people know in advance that is what you will do. Let them know their positions of responsibility make them the first suspects on the list if trust is violated.

Minimize your windows of vulnerability. Once you know ways in which you can be vulnerable, develop plans to minimize and mitigate those vulnerabilities. If you can close the hole, then close it. If you cannot close it, then limit what can be done through the hole. If you cannot limit what can be done, then limit who can do it and when it can be exploited. If you cannot limit anything, then at least measure whether an exploit is taking place. You may not have a perfect solution, but the more limits you put on a potential problem, the less likely it is that it will become a real problem.

Layer your security. When it comes to trust, you should not depend on any one entity for security. This is known as "defense in depth." If you can have multiple layers of encryption, for example, each implemented differently (one depending on OpenSSL, for example, and the other using a different package), then a single vulnerability will not leave you completely exposed.

This is a good reason to look at every component of your enterprise and ask: What if these components were to be compromised?

Practice being agile. If a component were compromised, how would you replace it, and with what? How long would it take to switch over? Theories do not count here. You need to be prepared to switch packages or vendors or hardware in order to be adequately safe. How long will it take your purchasing department to cut paperwork for a new license, for example? How long to get that purchase order signed off? How long for the vendor to deliver?

This is not work you can do once and think you are ready. You need to revisit all components regularly and perform this kind of analysis for each of them as circumstances change.

Look at your network as an attacker would. Know the "as-built" configuration of your network, not just the "as-specified." Remember the as-built configuration can change every day. This means you have to have people to measure the network, and tools to examine it. What network services does each component provide? Are those services needed? Are they available only to the places they are needed? Are all of the components fully patched? Are they instrumented to detect and report attack attempts? Does someone read those logs? What is the longest period of time between when an attack happens and when somebody notices it? Are there any events (such as holidays) when the length of time an attack goes unnoticed might increase?

The Internet abounds with free or inexpensive software for security analysis. These are tools often used by attackers and defenders. There is something to be learned by looking at your network through the same tools your attackers might use.

Track security issues and confirm they get fixed. If you find a problem, how is it tracked? Who is responsible for getting it into the tracking system, getting it to someone who can fix it, and getting it fixed? How do you measure the problem is present? Do you measure again after the fix is applied to ensure it worked?

Develop your own security intelligence resources. Does your organization have personnel who track the technology used for potential security issues? How often do they check? Are they listened to when they report a problem?

Any equipment, software, vendors, or people you depend on should be researched on a regular basis. Quality security-focused websites exist, but they are often surrounded and outnumbered by those with products to sell or misinformation to distribute. Having staff gain the expertise to distinguish the good from the bad is extremely valuable.

Plan for big-ticket problems. If you run a networked enterprise, whether you provide a public, private, or internal suite of services, you will find that trusted services *will* fail you, sooner or later. Repeatedly. How you respond to those failures of trust will become a big part of your company's reputation. If you select your vendors, partners, and components wisely, seriously plan for responses to trouble situations, and act on your plans when the time comes, then you will fare much better in the long run than those whose crisis planning is filed under "Luck."

Conclusion

The problem of trust is not new. If anything, the only *new* part is the mistaken impression that things can be trusted, because so many new things seem to be trustworthy. It is a sometimes-comforting illusion, but an illusion nonetheless. To build anything of value, you will have to place your trust in some people, products, and services. Placing that trust wisely is a skill that is best learned over time. Mistakes will abound along the way. Planning for your mistakes and the mistakes of others is essential to trusting.

It is generally better, faster, and safer to take something that meets good standards of trustworthiness and add value to it—by auditing it, layering on top of it, or adding to the open source—than it is to roll your own. Be prepared to keep a wary eye on the components you select, the system you include them in, and the people who build and maintain that system. Always plan for trouble, because trouble will surely come your way.

You must have some trust if you want to get anything done, but you cannot allow yourself to be complacent. Thomas Jefferson said, "Eternal vigilance is the price of liberty." It is the price of security as well.

Acknowledgments

Thanks to Jim Maurer and George Neville-Neil of the *ACM Queue* Editorial Board for encouraging and supporting this article. An extended version of this article is available at http://queue.acm.org/detail.cfm?id=2630691.

Related articles on queue.acm.org

The Answer is 42 of Course
Thomas Wadlow
http://queue.acm.org/detail.cfm?id=1071727

Weapons of Mass Assignment
Patrick McKenzie
ttp://queue.acm.org/detail.cfm?id=1964843

LinkedIn Password Leak: Salt Their Hide
Poul-Henning Kamp
http://queue.acm.org/detail.cfm?id=2254400

References

1. Brooks Jr., F.P. *The Mythical Man-Month.* Addison-Wesley, 1975.
2. Coverity. Coverity Scan report finds open source software quality outpaces proprietary code for the first time, 2013; http://www.coverity.com/press-releases/coverity-scan-report-finds-open-source-software-quality-outpaces-proprietary-code-for-the-first-time/.
3. McMillan, R. IT admin locks up San Francisco's network. PCWorld, 2008; http://www.pcworld.com/article/148469/article.html.
4. Rumsfeld, D. Press conference (Feb. 12, 2002); http://www.c-span.org/video/?168646-1/DefenseDepartmentBriefing102.
5. Stephenson, N. *The Diamond Age.* Bantam Spectra, 1995.
6. Stoll, C. *The Cuckoo's Egg.* Doubleday, 1989.
7. Venizia, P. Sorting out the facts in the Terry Childs case. *CIO* (July 31, 2008); http://www.cio.com.au/article/255165/sorting_facts_terry_childs_case/?pf=1.

Thomas Wadlow is a network and computer security consultant based in San Francisco, where he enjoys the cafes. but *never* uses his keys to save an outdoor table.

DOI:10.1145/2617754

Article development led by acmqueue
queue.acm.org

A discussion with Michael Donat, Jafar Husain, and Terry Coatta

Automated QA Testing at Electronic Arts

TO MILLIONS OF game geeks, the position of quality assurance (QA) tester at Electronic Arts must seem like a dream job. But from the company's perspective, the overhead associated with QA can look downright frightening, particularly in an era of massively multiplayer games.

Hence the appeal of automated QA testing, which has the potential to be faster, more cost-effective, more efficient and more scalable than manual testing. While automation cannot mimic everything human testers can do, it can be very useful for many types of basic testing. Still, it turns out the transition to automated testing is not nearly as straightforward as it might at first appear. Some of the thorniest challenges are considered here.

At Electronic Arts (EA) in Vancouver, British Columbia, **Michael Donat** is an advocate of automation. His current focus is process improvement on the Player and Business Analysis team. He was previously the manager of QA at Silicon Chalk and ActiveState Corp., and has worked at Microsoft as a software design engineer.

Joining the discussion is **Jafar Husain**, a lead software developer for Netflix. Previously he worked at Microsoft, where one of his tasks involved creating the test environment for the Silverlight development platform. There he was introduced to Model View ViewModel (MVVM); he is a convert, he says, and now likes to spread the gospel of MVVM where applicable.

Terry Coatta, a member of the *ACM Queue* board, brought this group together to discuss the potential for automated QA testing. He and Donat once worked together at Silicon Chalk, where creating a sophisticated test environment was among their challenges. Coatta is now the CTO of Marine Learning Systems developing a learning management system for marine workers.

TERRY COATTA: In terms of your efforts so far to apply automated QA testing at EA, I gather you've found the going a little bumpy.

MICHAEL DONAT: We started the journey thinking automation was a good idea, but then we tried it, and it failed. Still, we figured out what was wrong, and we fixed it. But, while we made it to a nice plateau, we realized there was still a long way to go. Our solution clearly wasn't going to get us everything we wanted—which was a way to broadly apply automated testing. To get there, and for some other reasons, several of us have concluded that what we really need is a new architecture along the lines of MVVM.

JAFAR HUSAIN: What exactly was your driver for automating in the first place?

DONAT: Our primary motivation had to do purely with the cost of manual testing, which has become quite significant given the complexity of our

MICHAEL DONAT

We're trying to determine how we can specify these things in such a way they will be understandable, maintainable, and robust in the face of change.

games. Basically, code changes that require us to retest everything manually can be incredibly expensive. By reducing those costs, we felt we would have an opportunity to redirect our testers away from what I call "stability testing"—which is something automation is capable of handling—so they can start focusing more on the authenticity and fun of our game experience.

COATTA: In terms of stability testing, what did you see as your first opportunities for automation?

DONAT: We started looking at this seriously when we were working on EA Sports' FIFA 10 [soccer game]. Initially, that involved 10 vs. 10 gameplay, which then became 11 vs. 11 with the addition of goalies. So we needed that many testers—either 20 or 22. But that's not all, since we also needed to test for interactions between different matches to make sure the server wasn't getting confused about what information to send to which match. So, in addition to the testers required to cover one match, we needed to have at least one other match in play at the same time—meaning we actually needed to have 40 or so testers involved at the same time.

Then, even after we'd managed to get everyone organized, we might end up running into some trivial bug just seconds into the match that would bring the whole thing down. Besides being wasteful, that was extremely frustrating for a lot of people who could have been doing something more productive during that time. All that came together to make a pretty strong argument for automation.

COATTA: What were some of the problems you encountered as you worked toward that end?

DONAT: First, setting up an OTP (online team play) match in FIFA 10 required the user to go through a few screens. There were 20 consoles and the script was time-based, meaning it sent commands to the consoles and then waited for some prescribed amount of time for all of them to get into the right state. Then it would send out the next batch of commands. The goal was to move the consoles in lockstep through a set of screen transitions

ALL PHOTO TREATMENTS BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

> JAFAR HUSAIN
>
> **It's one thing to have two different but similar libraries in a code base, while it's quite another to have two different paradigms within the same code base. When you're in that situation, it can be very difficult for onboarding developers to figure out exactly what to do.**

in order to set things up for gameplay: participants chose which side they wanted to play, what jersey they wanted to wear, what position they wanted to play, and various other parameters. All those things needed to happen in concert just to keep the programming for the game as simple as possible.

At the time, our primitive test-automation system made navigating the front end problematic. Timing had to be just right, or tests would fail. As a result, I began advocating for a means of skipping the front end altogether, but I was forced to change my point of view. During manual testing of FIFA 10 OTP, a number of issues came up—so many, in fact, that the budget for manual testing had to be increased significantly. The question around the organization was, "How can we stop this from happening in the future?"

That led me to analyze roughly 300 crash bugs for which we had obtained data in the QA cycle. Part of my goal was to see whether there was any significant ROI to be realized by continuing to pursue automation. I found that slightly more than half of our crash bugs were actually coming up in those initial screen transitions. It turned out I'd been telling the games developers exactly the wrong thing. That is, we really did need to do testing on the front end, as well as on the back end. We couldn't make automation simpler just by getting rid of the front end.

COATTA: That's interesting. It seems like all that's happening on the front end is that people are choosing things from menus, so how likely are you to find bugs there? In reality, what looks like a simple process of choosing menu items actually amounts to a distributed computation. You've got 20 different things going on, with input coming from all these different places, and now all of that has to be coordinated.

DONAT: Exactly. It became clear we needed a different mechanism altogether. Just sending control inputs wasn't going to be enough. We needed the test program to be aware of where it was on a particular console and then be able to move that forward in an error-correctable way.

The guys who had originally put together the test-automation framework for FIFA had realized this would be necessary, but the capability for handling it had rotted over the years and didn't really exist by the time we were ready to tackle FIFA 11. So, one of the things we had to do was get the details we needed to see coming out of the UI so we'd be able to tell where things actually were.

HUSAIN: I guess that instead of driving things from the view layer—that is, going through the controller and the views—you needed to bypass that view and go directly to the model itself.

DONAT: Believe it or not, we were not at that stage yet. At that point, we were just happy to have scripts that were far more reliable, simply because they knew where they were in the state of the program.

COATTA: That way, you could actually close the feedback loop. Before that, you would send a command and then have to wait and trust in God that something wasn't going to happen in the meantime, whereas now you don't need to have that trust since you can verify instead.

DONAT: Right. We got to where we had more of a controlled state transition. Another big QA improvement we made on FIFA 11 was the addition of Auto Assist, whereby automation could be left to run the game itself while one or two manual testers drove the actual gameplay by supplying controller inputs for selected consoles. They didn't need to have 20 people on hand. That represented a huge improvement.

COATTA: Some people might have just rested on their laurels at that point.

DONAT: Maybe, but it was just one step for me. While introducing some test automation to specific applications like FIFA OTP is a wonderful thing, what I really want is a much broader application for stability purposes because that's what will make it possible for us to focus our testers on the overall game experience. That's the way to go about building a superior product.

The work on FIFA 11 helped convince EA of the potential benefits of automated testing, but accomplishing that end was clearly going to require a different architecture. The answer seemed to lie with the MVVM paradigm, an architectural pattern largely based on MVC. MVVM facilitates a clear separation between the development of the graphical user interface and the development of the back-end logic, meaning it should allow EA to separate OTP gameplay testing from UI testing.

COATTA: Looking back on where things stood once you'd finished with FIFA 11 test automation, what did you see as your next steps?

DONAT: As encouraging as FIFA 11 proved to be, the problem was that we had to spend a ton of time coding. Mostly that's because during game development, changes frequently would be made to certain screens, and then we would have to make corresponding changes in our test-automation script. That wasn't always properly coordinated, so things would end up breaking. As a result, we had a very fragile test-automation script that required a software engineer virtually dedicated to working on maintenance.

In the case of FIFA 11 OTP, that expense was justified, but I couldn't make the case for applying a similar level of test-automation effort across every other area of the game. We had to continue relying on a large number of manual testers to cover the full breadth of testing. Which made it pretty obvious we needed to figure out a way to encode our tests so that ongoing maintenance could be performed less often, using less expensive resources.

COATTA: And that led you where exactly?

DONAT: Basically, it meant the architecture would need to change. It should be easy to see how the game is laid out in terms of its screen transitions, but there should also be ready access to the data those screens act upon. In general, things should just be more accessible at a higher level of abstraction than is currently the case.

HUSAIN: Is it fair to say you would like to focus on workflows independent of the actual UI controls?

DONAT: That's absolutely right. Once that became clear, we realized we needed a different architecture—something more like MVVM. That isn't to say it has to be MVVM; it just needs to be something that can provide that sort of capability.

COATTA: What is it about the MVVM paradigm that's important?

DONAT: Essentially, it allows us to separate the data used by the screens from the screens themselves. We need that separation so automation systems can gain access to the things they need to tie into.

HUSAIN: It might be useful to contrast the MVVM approach with other patterns many developers might be more familiar with—MVC, for example. In MVC architecture both the controller and the view know about each other and send messages to each other directly. In MVVM architecture, instead of a controller, you have a view model, which is just that—a model of the view. The view model stores the state of the view, and the view object decides how the state of the view model ought to be presented.

Unlike in the MVC pattern, the view model has no direct knowledge of the view. Instead of sending messages to the view directly, the view model updates the view indirectly via the observer pattern. When the view model is changed, it broadcasts those changes, and the view responds by updating itself. The main advantage of this is that it's possible to test that the view models are in the correct state without even instantiating the view objects, which would add many asynchronous operations (usually related to rendering) that in turn would have to be coordinated.

Testing new models this way is easy since your models expose methods that can be directly invoked. Testing logic through the view layer is much more prone to error since it requires waiting for buttons to load and relies on the delivery of brittle messages such as simulated mouse clicks and key events.

Anyway, as you've moved beyond FIFA 11, what additional steps have you taken toward an MVVM sort of architecture?

DONAT: I should point out that improved test automation is only one benefit of MVVM. Several other groups at EA are also moving this way for a variety of reasons. The steps we've taken so far have mostly been to make the separation of the data from the screens more apparent. Unfortunately, FIFA has so many screens that we can't just go in and rewrite everything. What we can do, however, is to work the new paradigm into new features.

HUSAIN: It's interesting that, in the face of so many challenges, you've chosen to evolve your architecture in this stepwise manner toward MVVM. It seems you've found it easier just to add new events or extra components that follow this new pattern and then start using those as you can. I presume at some point the plan is to make a more wholesale transition to MVVM—or something like it—as that opportunity presents itself.

DONAT: That is the plan because it's the only way we can actually go about it. It's going to be a while before we can achieve the full breadth of automation I'm pushing for, but at least we're moving in the right direction.

Our next challenge is figuring out how to specify our tests, since we now have an architecture that lets us access that stuff. But we still don't know what those tests ought to look like, how they should be packaged, or how to contain the information such that it's easy to maintain and makes sense to the people who maintain it.

COATTA: What's the pushback on arguments for an MVVM-like environment? Are people afraid the transition would be too hard?

DONAT: There's no doubt it would be hard. What makes it worse is the software engineers would have to make those changes in lieu of adding some new features, which can be a very difficult sacrifice to justify. I can't even say exactly how much they would be able to save as a consequence of automation. The truth is they probably wouldn't save all that much since we're just talking about moving manual resources from one kind of testing to another.

COATTA: Do you think it would actually be more expensive to build in MVVM? Or is this really more about resistance on the part of the software engineers to making any changes to the way they're accustomed to working?

DONAT: That depends on the underlying code involved. Also, we sometimes make incremental changes to existing features. That is, we sometimes need to rewrite features because they need to evolve beyond the original design. If we're about to rewrite a feature anyway, that certainly presents an opportunity to take the newer approach.

On the other hand, if we're putting in a new twist for an existing game mode or adding a small feature to something that's already there, it would be very difficult to do that the new way while all that old stuff is still around. That would serve only to make those incremental changes all the more expensive.

HUSAIN: It seems that, in order to get to a place where you've actually got something useful, you're going to need to move an entire workflow to MVVM. I suppose that's going to be difficult to accomplish incrementally.

DONAT: That's right.

HUSAIN: We've run into this at Netflix. So I think you've touched on something that's worth pointing out—namely, it's one thing to have two different but similar libraries in a code base, while it's quite another to have two different paradigms within the same code base. When you're in that situation, it can be very difficult for onboarding developers to figure out exactly what to do. Have you found this to be a stumbling block? And has that caused any friction?

DONAT: Absolutely. There are many FIFA developers all over the world, so the idea of unifying all of them in support of moving in more of an MVVM direction is pretty hard to imagine.

HUSAIN: I wonder if the current attitude of those developers toward MVVM reflects the fact that the benefits you're touting will only be realized downstream. Beyond that, though, are they also aware that MVVM might be a better architecture in general for development, quite apart from any testing benefits?

DONAT: Actually, I've been really impressed with the software engineers I've worked with here. They all seem to know what the right thing to do is. But time is also an issue.

HUSAIN: Is it fair to say the developers don't have any objection to MVVM, and might even be very much in favor of making the necessary changes to use MVVM?

DONAT: Often, I'll be talking to a group of game developers about some idea and they'll say, "Oh yeah, we already know we should go that route," but when it comes to implementation, they aren't able to follow through because of time constraints.

HUSAIN: In terms of how you move forward, I gather you still have some questions regarding the architecture and that you're also still trying to figure out what the API for your testers ought to look like.

DONAT: That's right, although I'd put the emphasis on specification rather than API, because programming is expensive. We're trying to determine how we can specify these things in such a way that they'll be understandable, maintainable, and robust in the face of change. That is, in its purest form, you'd like to run an OTP test where you have 22 consoles, with 11 being assigned to one team and the other 11 going to the other side, along with the ability to associate all appropriate parameters with each.

Then the question becomes: how can you specify that in such a way as to cover a broad range of tests? And that's, of course, because each time you run a test, you would like to be able to do different things. If you've got a multiple-match situation, for example, you might want to roll through all the different teams, stadiums, and jerseys so that over the course of many weeks of testing, you would wind up cycling through as many different combinations as possible—and all of that by essentially specifying only one test. That's our goal, anyway, but it's not entirely clear at this point how we're going to manage that.

HUSAIN: There really are two questions here: (1) Is it possible? (2) Does it scale? There are also some more advanced approaches you could use to build asynchronous tests, but would those then be accessible to junior developers or test engineers?

DONAT: Right. There's no point in doing this unless we can do it in a low-cost manner.

The transition to automated testing has a significant cost dimension when it comes to human resources. First, software engineers accustomed to doing things one way must be convinced to change. They must learn a new paradigm, move from synchronous to asynchronous programming, and perhaps even learn a domain-specific language (DSL) for writing event-based tests.

Second, it's essential to strike the right balance between the work done by lower-cost QA testers and that which is reserved for higher-paid specialists. This means taking advantage of the asynchronous nature of the game by emphasizing declarative tests that are started and gated by events, while designing tests orchestrated to play off those events. This could allow for large numbers of inexpensive coders to write the declarative tests, while a much more select set of expensive coders are left to focus on the more sophisticated orchestration issues.

HUSAIN: Have you explored different languages that might make it easier for lower-skilled developers to write event-based tests?

DONAT: I've been considering the possibility of using a DSL. What worries me, however, is that there was a time when we had to encode game information in the test code, and I'm afraid we might end up going back to encoding information in some other type of code if we were to choose the wrong DSL.

One of the DSL properties we'd be looking to use is a container for the game information that must be transparent enough so people can easily access that information. It's important the information can be accessed using vocabulary that both the QA people and the game producers are familiar with.

HUSAIN: Understood. The line between where a DSL begins and a library ends can be somewhat blurry. But a DSL can also be embedded as part of the general-purpose programming language you already use.

DONAT: At the moment I don't think we're really going to be looking at any if-then-else loop coding. We're probably talking only about tests at the level of stimulus and response—that is, "When the program responds in this particular way, then provide this sort of stimulus."

COATTA: Jafar, have you had any experience with DSLs at Netflix?

HUSAIN: We're actually currently struggling to make a similar transition—not so much for testing but more for finding a better way to coordinate the concurrency in our application. What we're using for that now is the Rx (Reactive Extensions) library. The interesting thing about this library is that it has actually been integrated into C#, meaning you can use it at a much higher level than if-then-else to coordinate asynchronous processes. There's also a JavaScript version of Reactive Extensions, which is what we're using now.

While this should make things very easy, in practice we're finding that it's a very new way of thinking for developers—particularly those who have come from a background of if-then-else, imperative, top-down programming. And this is despite the fact that the Rx abstraction is at a much higher level and is, in fact, quite declarative and obviously flexible, powerful, and capable enough to handle all sorts of complex asynchronous operations. It's not so much a matter of this new language being any more or any less difficult to work with; it's just that when you come from a synchronous way of thinking, making the transition to programming asynchronously can be very challenging.

> “
> TERRY COATTA
> **My group has been developing for asynchronous environments, and finite-state machines have worked really well in that regard. We've found them to be a stunningly good way to capture information about events and transitions and stuff like that.**
> ”

Asynchronous programming requires a significant investment in terms of learning something new and a whole different way of thinking about your code. Which is to say I'm skeptical you'll manage to find a DSL out there that can transform a synchronous programmer into an asynchronous programmer in a few weeks, or even over the course of a product cycle.

DONAT: That's my fear as well. There's going to be a need for people in the loop who are skilled in asynchronous programming. Whoever is coding up these exotic OTP tests where we have two or three matches going on at the same time is definitely going to need those skills.

But the open question for me is: How can you do that and still have the QA people specify most of the tests? It would be fantastic if we could just get to the point where 80% of the game code could be covered by tests written by the QA people. And then if the other 20% of the OTP tests had to be written by highly paid specialists, so be it. I would be cool with that just so long as we could get a large proportion of the code covered in a lower-cost manner.

HUSAIN: Those specialized developers might be expensive, but if they're using the right set of tools or languages or frameworks or paradigms, then you have the potential to squeeze a lot more out of them. There's real value in identifying those individuals who are naturally inclined toward asynchronous programming and intensively training them. Beyond that, I think we're starting to see more frameworks and tools that have the potential to yield some tremendous savings once you start leveraging them such that those specialists can produce six or seven or even eight tests a day instead of just two.

COATTA: Initially I got the sense, Michael, that you were hoping to find a DSL that would let you take better advantage of QA personnel by enabling them to execute a reasonably broad set of tests. Meanwhile, Jafar, it sounds like your experience so far is that the asynchronous stuff is sufficiently complex that the real win lies in finding those people who have some natural talent for it and then making them super-efficient.

How is this going to play out long-term? Is asynchronous programming just so difficult that it's always going to be the province of power people? Or is there anything to suggest this can be made more accessible to less-sophisticated programmers?

DONAT: I think we're going to see a mix of the two. There's going to be some significant portion of any product that will remain fairly straightforward, where the coding is likely to be the kind that can be handled by lower-cost individuals once you've got the right framework in place. But that framework is going to need to be set up by someone who understands asynchrony and who has the training and experience to deal with other reasonably complex requirements. There's definitely going to be a role for some highly trained and talented individuals, but you also want to make sure you can leverage those efforts to make their contributions broadly applicable.

HUSAIN: I'm a little pessimistic about that. We recently were looking to build some asynchronous frameworks on the server at Netflix, and I think some of our developers started out with a similar attitude, based on the assumption that maybe 80% of our asynchronous problems could be easily solved with a few helper methods. The idea was to provide some simple APIs for a few of the more common concurrency patterns so junior developers would be able to tackle most of the asynchronous problems. We discovered that simple APIs solved only about 10%–15% of our end-user cases—not 80%. That's because it was very easy to fall off a cliff, at which point it became necessary to revert to dealing with primitives such as semaphores or event subscriptions.

It turns out that even seemingly trivial async problems are actually quite complicated. For example, if you're making a remote request, it will invariably require some error handling like a retry. If two operations are executing concurrently, you'll need a way to specify different error-handling policies for each operation. What's more, each of these operations might be made up of several other sequential and concurrent operations. In reality, you need a compositional system to be able to express such rich semantics.

I admit it's possible that some simple helper APIs might prove more useful for building tests since the requirements are less stringent than for app development. So maybe you're right, Michael, to think you can mix low-skilled programmers with highly skilled ones. What exactly that mix looks like remains to be seen, though.

DONAT: I couldn't agree more. I think that's the big question.

COATTA: On a somewhat different note, my group has been developing for asynchronous environments, and finite-state machines have worked really well in that regard. We've found them to be a stunningly good way to capture information about events and transitions and stuff like that. So what are your thoughts about using state machines and some kind of language built around that? Are state machines simple enough for less-skilled developers, like QA people, to use them effectively?

DONAT: I certainly think state machines describe the mathematics well enough. A transition effectively amounts to a stimulus-response pair. So, yes, you can describe what we're talking about as hierarchical-state machines. And yes, that's the perfect mathematical paradigm to use for discussing this. But you can't present that to low-cost personnel and expect them to be able to do anything with it. What you can do, though, is to use those same mathematics to create the tools and the machinery that drives all this stuff. In terms of what you put in front of the QA people, however, that can't be anything more than what they already recognize as stimuli to the responses they're looking for.

HUSAIN: I completely agree. It's true that the primitives are simple enough that everyone can understand how to hook up to an event, set a variable, and then move from state to state. In practice, however, those simple primitives don't mean the overall program itself is going to be simple. In fact, it's going to be quite complex because there are so many different moving parts.

Another approach to asynchronous programming that's gaining a following in the JavaScript world is around an abstraction called Promises that's being integrated into common JS and F#. That gives you an async type that provides for composability while letting you build asynchronous programs in a stateless way. That might be the model you end up having to embrace—a declarative stateless way of describing an asynchronous computation—since that's a level of abstraction that lower-skilled developers might be able to take advantage of.

COATTA: Could you provide an example of that?

HUSAIN: What it comes down to is that there's a new way of thinking about asynchronous programs. The move away from GOTOs to structured programs raised the level of abstraction. Today, we build asynchronous programs with callbacks and state machines, and these programs suffer many of the same disadvantages of the old GOTO-based programs: logic tends to be fragmented into many different pieces. We can resolve this problem the same way we resolved it earlier—by raising the level of abstraction. Instead of using callbacks and state machines to build asynchronous programs, we can model them as sequences of data. An event, for example, can be seen as a sequence of data—one, in fact, that has no end. There's no way a mouse-move event is going to be able to say, "Hey, I'm done." It just goes on and on forever.

It's interesting to note we already have a means for modeling sequences in synchronous programming: the familiar iterator is a synchronous way of moving through a data structure, from left to right, simply by continuing to request the next item until the iterator finally reports there's no more data. Erik Meijer, when he was at Microsoft, turned the iterator pattern inside out and found the observer pattern fell out. In mathematical terms, the observer pattern is the dual of the iterator pattern. This is a very important unification since it means anything we can do to an iterator can also be done to observers such as event listeners.

The significance here is that we have several high-level languages for manipulating data structures that can be expressed as iterators. The most relevant example is SQL, which I would argue is a very successful high-level language because it allows developers to create complex queries that are both easy to understand and powerful to use. Now, based on the discovery that the observer and iterator patterns are dual, Erik has managed to build a framework that allows an SQL-like language to be used to create asynchronous programs.

The idea is that events and asynchronous requests for data are collections, just like arrays. The only difference is that asynchronous collections arrive over time. Most operations that can be performed on a collection in memory can also be performed on collections that arrive over time. Hence, we find that a DSL originally built into C# to compose synchronous sequences of data can also be used to compose asynchronous sequences. The result is a high-level language for building asynchronous programs that has the expressiveness and readability of SQL.

DONAT: That's a step in the right direction. I'm definitely going to look into this further.

HUSAIN: We're using this technology on our Xbox platform right now. It seems to be just what you're looking for, Michael.

COATTA: Can you describe how Erik Meijer's Reactive Extensions work applies in a QA environment? Say you've got a bunch of consoles you need to drive through some sequences so you can verify that certain things are happening in the game you're testing. Where does Rx fit into that? What would you be querying in that circumstance and how would you be able to turn that into a test result?

HUSAIN: That's a great question, since some people have difficulty seeing the connection between querying a database and creating a test. A test in its own way is actually a query along the lines of: "Did this stream fire and did that stream fire before some particular event fired, which then led to some other thing happening?" That's really no different from querying a table to see whether a certain condition is true.

COATTA: We would still need some mechanism to drive the system through different states. Perhaps Rx could even be used for that. At each stage the query is going to come back as either true or false. If it comes back false, then we'll know the test didn't pass since the sequence of events we had been expecting didn't match the query we issued.

HUSAIN: Exactly right. But this can be partitioned into two steps. The first is the one Michael already mentioned: transitioning the system so as to make it more observable, and by and large that's simply a matter of adding events that fire when interesting things occur. The second step involves building queries over those events. Those queries would be very, very declarative—they wouldn't be state machines at all—so you would be able to confirm that certain conditions are met as you drive through the system.

COATTA: It sounds like you're applying this approach to a product now. Has that experience proved to be positive? Are you finding that the Rx syntax or the query syntax is something nonexperts might be able to use to capture information about the system?

HUSAIN: Thus far, I don't think the syntax has really helped as much as I'd anticipated. The real challenge is in making the leap to thinking about events as collections. Most people have spent their careers thinking about events very mechanistically. Although thinking about events as collections might be conceptually simpler, it may also prove difficult to make the transition at the organizational level, if only because it's so hard to break bad old habits. My sense, however, is that if you can find some developers who are already inclined toward functional programming, then when you give them these powerful new tools for asynchronous programming, you're going to be able to realize the sorts of economies we're talking about.

Related articles on queue.acm.org

Orchestrating an Automated Test Lab
Michael Donat
http://queue.acm.org/detail.cfm?id=1046946

Finding Usability Bugs with Automated Tests
Julian Harty
http://queue.acm.org/detail.cfm?id=1925091

Adopting DevOps Practices in Quality Assurance
James Roche
http://queue.acm.org/detail.cfm?id=2540984

DOI:10.1145/2622628.2622630

Article development led by acmqueue
queue.acm.org

If you see something, say something.

BY MIKE BLAND

Finding More Than One Worm in the Apple

IN FEBRUARY, APPLE revealed and fixed a Secure Sockets Layer (SSL) vulnerability that had gone undiscovered since the release of iOS 6.0 in September 2012. It left users vulnerable to man-in-the-middle attacks thanks to a short circuit in the SSL/TLS (Transport Layer Security) handshake algorithm introduced by the duplication of a `goto` statement. Since the discovery of this very serious bug, many people have written about potential causes. A close inspection of the code, however, reveals not only how a unit test could have been written to catch the bug, but also how to refactor the existing code to make the algorithm testable—as well as more clues to the nature of the error and the environment that produced it.

This article addresses five big questions about the SSL vulnerability: What was the bug (and why was it bad)? How did it happen (and how didn't it)? How could a test have caught it? Why didn't a test catch it? How can we fix the root cause?

The Apple SSL vulnerability, formally labeled CVE-2014-1266, was produced by the inclusion of a spurious, unconditional `goto` statement that bypassed the final step in the SSL/TLS handshake algorithm. According to the National Vulnerability Database (http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-1266) and the Common Vulnerabilities and Exposure (CVE) Standard Vulnerability Entry (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1266), the bug existed in the versions of iOS, OS X, and the Apple TV operating system shown in the accompanying table.

These formal reports describe the bug as follows: "The `SSLVerifySignedServerKeyExchange` function in libsecurity_ssl/lib/sslKeyExchange.c in the Secure Transport feature in the Data Security component...does not check the signature in a TLS Server Key Exchange message, which allows man-in-the-middle attackers to spoof SSL servers by using an arbitrary private key for the signing step or omitting the

IMAGE FROM SHUTTERSTOCK.COM

signing step." This error is visible by searching for the function name within Apple's published open-source code (http://opensource.apple.com/source/Security/Security-55471/libsecurity_ssl/lib/sslKeyExchange.c) and looking for this series of statements:

```
if ((err = SSLHashSHA1.update(
    &hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
```

Those familiar with the C programming language will recognize that the first `goto fail` is bound to the `if` statement immediately preceding it; the second is executed unconditionally. This is because whitespace, used to nest conditional statements for human readability, is ignored by the C compiler; curly braces must enclose all statements bound to an if statement when more than one statement applies.

The other `goto fail` statements appearing throughout the algorithm are a common idiom in C for releasing resources when a function has encountered a fatal error prior to completion. In the flawed code, a successful `update()` call will result in an unconditional jump to the end of the function, before the final step of the algorithm; and the return value will indicate the handshake was successful. In essence, the algorithm gets short-circuited.

For users of Apple's Safari and other Secure Transport-based applications on the affected platforms, "secure" connections were vulnerable to man-in-the-middle attacks, whereby an attacker in a position to relay messages from a client to a "secure" server across the Internet can impersonate the server and intercept all communications after the bogus handshake. (Users of products incorporating their own SSL/TLS implementations, such as Google Chrome and Mozilla Firefox, were not affected.) Though it is unknown whether this vulnerability was ever exploited, it rendered hundreds of millions of devices (and users) vulnerable over the course of 17 months.

Apple was criticized for patching the vulnerability for iOS devices and Apple TV on Friday, February 21, 2014, making knowledge of the vulnerability widespread, but delaying the patch for OS X Mavericks until the following Tuesday. This four-day window left users who were not aware of the iOS patch vulnerable to a now very public exploit.

How Did It Happen (And How Didn't It)?

Many have noted apparently missing factors that could have caught the bug. Coding standards—specifically those enforcing the use of indentation and curly braces—combined with automated style-checking tools and code reviews, may have drawn attention to the repeated statement. An automated merge may have produced the offending extra line, and the developer may have lacked sufficient experience to detect it. Had coverage data been collected, it would have highlighted unreachable code. Compiler and static-analysis warnings also could have detected the unreachable code.

Others noted the code appears to lack unit tests, which likely would have caught the bug. While many of the other tools and practices might have been sufficient to prevent this specific vulnerability, a deeper problem, which ultimately produced the repeated `goto` statement, would have been prevented by proper unit-testing discipline.

Some question whether adequate

system testing took place, while others argue that because system testing cannot find every bug, this was merely an unfortunate case of one that happened to slip through. Others claim use of the goto statement and/or deliberate sabotage is to blame. None of these claims stands up to scrutiny.

GO TO not "considered harmful." Since it is one of the more popular theories, let's dispatch the argument that the use of goto is to blame for this vulnerability. Many have referred to the popular notion that goto is "considered harmful," based on Edsger Dijkstra's letter published in the March 1968 *Communications of the ACM*. This is what Dijkstra actually said in "A Case against the GO TO Statement": "I do not claim that the clauses mentioned are exhaustive in the sense that they will satisfy all needs; but whatever clauses are suggested (for example, abortion clauses) they should satisfy the requirement that a programmer-independent coordinate system can be maintained to describe the process in a helpful and manageable way."[9] In other words, "abortion clauses" to release a function's resources may still rely on goto, absent other direct language support.

This C language "abortion clause" idiom is legitimate and well understood, and is directly supported by other languages. For example, in C++, automatic destructors implement the Resource Acquisition Is Initialization (RAII) idiom; Java employs try/catch/finally blocks (http://docs.oracle.com/javase/tutorial/essential/exceptions/handling.html); Go provides the defer(), panic(), and recover() mechanisms (http://blog.golang.org/defer-panic-and-recover); and Python has try/except/finally blocks (http://docs.python.org/3/reference/compound_stmts.html#try) and the with statement, which is used to implement RAII (http://docs.python.org/3/reference/compound_stmts.html#the-with-statement). Absent these mechanisms, in C this remains a legitimate application of the goto statement, lest the code become bloated with repetitive statements or the control structure become nested so deeply as to hinder readability and maintainability.

Schedule of affected systems and security updates.

System	Vulnerable Versions	Vulnerable Since	Fixed Versions	Patch Date
iOS	6.x - 6.1.5	2012-09-19	6.1.6	2014-02-21
	7.x - 7.0.5	2013-09-18	7.0.6[a]	2014-02-21
OS X	10.9 - 10.9.1	2013-10-22	10.9.2[b]	2014-02-25
Apple TV	6.x - 6.0.1	2012-09-24	6.0.2	2014-02-21

[a] iOS 7.0.6 release notes; http://support.apple.com/kb/HT6147.
[b] OS X 10.9.2 release notes; http://support.apple.com/kb/HT6114.

Figure 1. The handshake algorithm containing the goto fail bug.

```
if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
```

Figure 2. The duplicate handshake algorithm appearing immediately before the buggy block.

```
if(isRsa) {
  /* … */
  if ((err = ReadyHash(&SSLHashMD5, &hashCtx)) != 0)
    goto fail;
  if ((err = SSLHashMD5.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
  if ((err = SSLHashMD5.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
  if ((err = SSLHashMD5.update(&hashCtx, &signedParams)) != 0)
    goto fail;
  if ((err = SSLHashMD5.final(&hashCtx, &hashOut)) != 0)
    goto fail;
}
```

In fact, a misplaced return statement could have produced the same effect. Imagine a macro such as the following had been defined:

```
#define ERROR _ EXIT {\
  SSLFreeBuffer(&hashCtx);\
  return err; }
```

Then the bug might have appeared in this incarnation:

```
if ((err = SSLHashSHA1.update(
    &hashCtx, &signedParams)) != 0)
    ERROR _ EXIT
    ERROR _ EXIT
```

Even enforcing the use of curly braces might not have prevented the error, as they could be mismatched:

```
if ((err = SSLHashSHA1.update(
    &hashCtx, &signedParams)) != 0)
{
    goto fail;
    goto fail;
if ((err = SSLHashSHA1.final(
    &hashCtx, &hashOut)) != 0)
    goto fail;
}
```

The blame for this vulnerability does not lie with the goto statement. A proper unit test would have caught the error regardless of how it was written.

Code duplication. The handshake algorithm in which the extra goto statement appears is duplicated six times throughout the code. Figure 1 shows the algorithm containing the repeated `goto fail` line as it appears in the `SSLVerifySignedServerKeyExchange()` function. Figure 2 shows the block immediately preceding this algorithm. *This duplication is the critical factor leading to the manifestation of the vulnerability*, and it can be traced directly to a lack of unit testing discipline—because of the absence of the craftsmanship and design sense that testable code requires. Someone writing code with unit testing in mind would have ensured only one copy of the algorithm existed—not only because it is theoretically more proper, but because it would have been easier to test.

The coder could not "smell" (http://blog.codinghorror.com/code-smells/) the duplicate code as he or she was writing it—or copying it for the second, third, fourth, fifth, or sixth time! This indicates a pattern of poor habits over time, not a single careless mistake. Ultimately, this speaks to a cultural issue, not a moment of individual error.

How Could a Test Have Caught It?

Landon Fuller published a proof-of-concept unit test implemented in Objective-C,[10] using the Xcode Testing Framework.[2] Fuller notes that "there's no reason or excuse for [the `SSLVerifySignedServerKeyExchange()` function] not being fully tested for" all of the potential error conditions. This proof of concept, however, misses the opportunity to look deeper into the code and provide full test coverage of this particularly critical algorithm—so critical that it appears six times in the same file.

Step one in properly testing the algorithm is to extract it into a separate function—which in itself might have prevented the duplicate `goto fail` that caused the bug, since a single block of code is less susceptible to editing or automated merge errors than six blocks of nearly identical code (Figure 3).

The two earlier blocks of code from `SSLVerifySignedServerKeyExchange()` now appear as in Figure 4.

Figure 3. The handshake algorithm extracted into its own function.

```
static OSStatus
HashHandshake(const HashReference* hashRef, SSLBuffer* clientRandom,
    SSLBuffer* serverRandom, SSLBuffer* exchangeParams,
    SSLBuffer* hashOut) {
  SSLBuffer hashCtx;
  OSStatus err = 0;
  hashCtx.data = 0;
  if ((err = ReadyHash(hashRef, &hashCtx)) != 0)
    goto fail;
  if ((err = hashRef->update(&hashCtx, clientRandom)) != 0)
    goto fail;
  if ((err = hashRef->update(&hashCtx, serverRandom)) != 0)
    goto fail;
  if ((err = hashRef->update(&hashCtx, exchangeParams)) != 0)
    goto fail;
  err = hashRef->final(&hashCtx, hashOut);
fail:
  SSLFreeBuffer(&hashCtx);
  return err;
}
```

Figure 4. `SSLVerifySignedServerKeyExchange()` **after extracting the handshake algorithm.**

```
if(isRsa) {
  /* … */
  if ((err = HashHandshake(&SSLHashMD5,&clientRandom,&serverRandom,
      &signedParams, &hashOut))!= 0){
  goto fail;
  }
} else {...}
...
if ((err = HashHandshake(&SSLHashSHA1,&clientRandom,&serverRandom,
     &signedParams, &hashOut)) != 0){
  goto fail;
}
```

This works because the `HashReference` is a "jump table" structure, and SSLHashMD5 and SSLHashSHA1 are instances of `HashReference`, which point to specific hash algorithm implementations. The `HashReference` interface makes it straightforward to write a small test exercising every path through the isolated `HashHandshake()` algorithm using a `HashReference` stub, and to verify that it would have caught this particular error:

```
+ build/libsecurity_ssl.build/.../
  x86_64/tls_digest_test
TestHandshakeFinalFailure failed:
  expected FINAL_FAILURE,
  received SUCCESS
1 test failed
```

The code for `tls_digest_test.c` is viewable at http://goo.gl/PBt9S7. Security-55471-bugfix-and-test.tar.gz (http://goo.gl/tnvIUm) contains all of my proof-of-concept changes; `build.sh` automates downloading the code, applying the patch, and building and running the test with a single command. The test and the patch are very quick efforts but work as a stand-alone demonstration without requiring the full set of dependencies needed to build the entire library. The demonstration admittedly does not address further duplication or other issues present in the code.

The point of all this is, if an ex-programmer who has been out of the industry for 2.5 years can successfully refactor and test this code within a couple of hours, never having seen it before, why didn't the engineer or team responsible for the code properly test it 17 months earlier?

Why Didn't a Test Catch It?

Several articles have attempted to explain why the Apple SSL vulnerability made it past whatever tests, tools, and processes Apple may have had in place, but these explanations are not sound, especially given the above demonstration to the contrary *in working code*. The ultimate responsibility for the failure to detect this vulnerability prior to release lies not with any individual programmer but with the culture in which the code was produced. Let's review a sample of the most prominent explanations and specify why they fall short.

Adam Langley's oft-quoted blog post[13] discusses the exact technical ramifications of the bug but pulls back on asserting that automated testing would have caught it: "A test case could have caught this, but it is difficult because it is so deep into the handshake. One needs to write a completely separate TLS stack, with lots of options for sending invalid handshakes."

This "too hard to test" resignation complements the "I don't have time to test" excuse Google's Test Mercenaries, of which I was a member, often heard (though, by the time we disbanded, testing was well ingrained at Google, and the excuse was rarely heard anymore).[11] As already demonstrated, however, a unit test absolutely would have caught this, without an excess of difficulty. Effectively testing the algorithm does not require "a completely separate TLS stack;" a well-crafted test exercising well-crafted code would have caught the error—indeed, the *thought* of testing likely would have prevented it altogether.

Unfortunately, some adopted Langley's stance without considering that the infeasibility of testing everything at the system level is why the small, medium, and large test size schema exists (that is *unit*, *integration*, and *system* to most of the world outside Google).[8] Automated tests of different sizes running under a continuous integration system (for example, Google's TAP, Solano CI) are becoming standard practice throughout the industry. One would expect this to be a core feature of a major software-development operation such as Apple's, especially as it pertains to the security-critical components of its products.

Writing for *Slate*, David Auerbach breaks down the flaw for nonprogrammers and hypothesizes that the bug might have been caused by a merge error (based on this diff: https://gist.github.com/alexyakoubian/9151610/revisions; look for green line 631), but then concludes: "I think the code is reasonably good by today's standards. Apple would not have released the code as open source if it were not good, and even if they had, there would have been quite an outcry from the open source community if they had looked it over and found it to be garbage."[3]

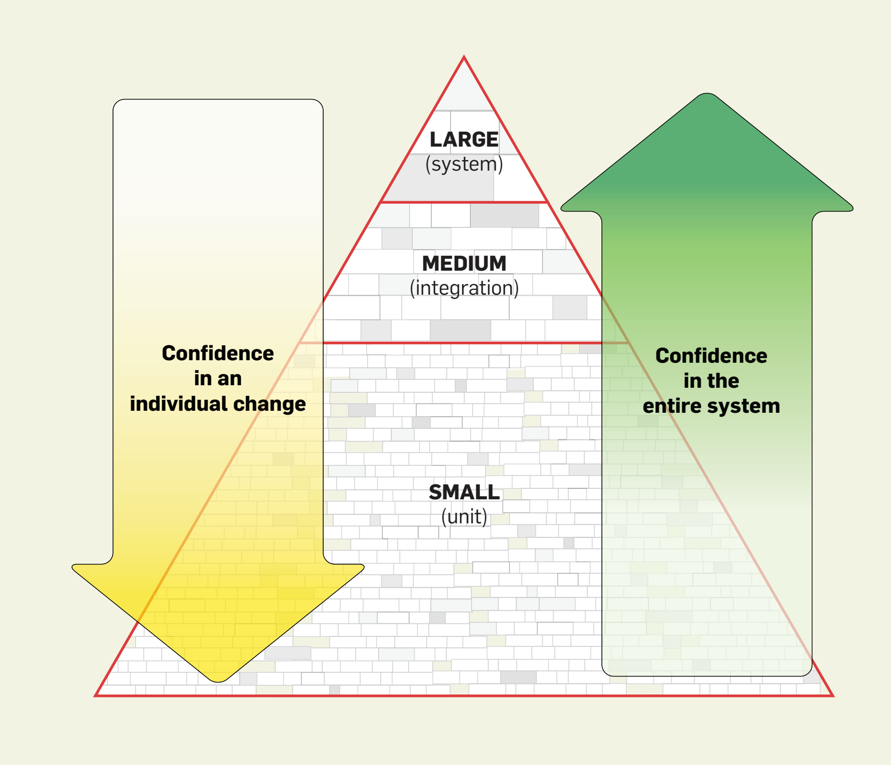
Auerbach's conclusion assumes that everything Apple releases is high quality by definition, that it has every reasonable control in place to assure such high quality, and that all open-source code receives the focused scrutiny of large numbers of programmers (thanks to Stephen Vance for pointing this out specifically in his comments on my presentation)—at least, programmers motivated to report security flaws. The actual code, however, suggests a lack of automated testing discipline and the craftsmanship that accompanies it, as well as the absence of other quality controls, not the fallibility of the existing discipline that Auerbach imagines Apple already applies.

Security guru Bruce Schneier notes, "The flaw is subtle, and hard to spot while scanning the code. It's easy to imagine how this could have happened by error.... Was this done on purpose? I have no idea. But if I wanted to do something like this on purpose, this is exactly how I would do it."[15] Schneier's focus is security, not code quality, so his perspective is understandable; but the evidence tips heavily in favor of programmer error and a lack of quality controls.

Delft University computer science professor Arie van Deursen notes many industry-standard tools and practices that could have caught the bug; but despite self-identifying as a strong unit-testing advocate, he demurs from asserting the practice should have been applied: "In the current code, functions are long, and they cover many cases in different conditional branches. This makes it hard to invoke specific behavior.... Thus, given the current code structure, unit testing will be difficult."[16] As already demonstrated, however, this one particular, critical algorithm was easy to extract and test. Software structure can be changed to facilitate many purposes, including improved testability. Promoting such changes was the job of the Test Mercenaries at Google.

My former Test Mercenary colleague C. Keith Ray noted both in his comments on van Deursen's post and in his own blog: "Most developers who try to use TDD [test-driven development] in a badly designed, not-unit-tested project

Figure 5. The small/medium/large test strategy.


LARGE
(system)
MEDIUM
(integration)
SMALL
(unit)
Confidence in an individual change
Confidence in the entire system

will find TDD is hard to do in this environment, and will give up. If they try to do 'test-after' (the opposite of TDD's test-first practice), they will also find it hard to do in this environment and give up. And this creates a vicious cycle: untested bad code encourages more untested bad code."[14]

I largely agree with Ray's statement but had hoped he might seize the opportunity to mention the obvious duplicate code smell and how to eliminate it. Again, that was our stock-in-trade as Test Mercenaries. Absence of TDD in the past does not preclude making code more testable now, and we have a responsibility to demonstrate how to do so.

Columbia University computer science professor Steven M. Bellovin provides another thorough explanation of the bug and its ramifications, but when he asks "why they didn't catch the bug in the first place," his focus remains on the infeasibility of exhaustive system-level testing: "No matter how much you test, you can't possibly test for all possible combinations of inputs that can result to try to find a failure; it's combinatorially impossible."[4]

As demonstrated, this vulnerability was not a result of insufficient system testing; it was because of insufficient unit testing. Keith Ray himself wrote a "Testing on the Toilet"[8] article, "Too Many Tests,"[11] explaining how to break complex logic into small, testable functions to avoid a combinatorial explosion of inputs and still achieve coverage of critical corner cases ("equivalence class partitioning"). Given the complexity of the TLS algorithm, unit testing should be the first line of defense, not system testing. When six copies of the same algorithm exist, system testers are primed for failure.

Such evidence of a lack of developer testing discipline, especially for security-critical code, speaks to a failure of engineering and/or corporate culture to recognize the importance and impact of unit testing and code quality, and the real-world costs of easily preventable failures—and to incentivize well-tested code over untested code. Comments by an anonymous ex-Apple employee quoted by Charles Arthur in *The Guardian*[2] support this claim:

"Why didn't Apple spot the bug sooner?

Given the complexity of the TLS algorithm, unit testing should be the first line of defense, not system testing. When six copies of the same algorithm exist, system testers are primed for failure.

"The former programmer there says, 'Apple does not have a strong culture of testing or test-driven development. Apple relies overly on *dogfooding* [using its own products] for quality processes, which in security situations is not appropriate....

"What lessons are there from this?

"But the former staffer at Apple says that unless the company introduces better testing regimes—static code analysis, unit testing, regression testing—'I'm not surprised by this... it will only be a matter of time until another bomb like this hits.' The only—minimal —comfort: 'I doubt it is malicious.'"

Reviewer Antoine Picard, commenting on the similarity between this security vulnerability and reported problems with Apple's MacBook power cords, noted: "When all that matters is the design, everything else suffers."[12]

How Can We Fix the Root Cause?

Those with unit-testing experience understand its productivity benefits above and beyond risk prevention; but when the inexperienced remain stubbornly unconvinced, high-visibility bugs such as this can demonstrate the concrete value of unit testing—*in working code*.

Seize the teachable moments! Write articles, blog posts, flyers, give talks, start conversations; contribute working unit tests when possible; and hold developers, teams, and companies responsible for code quality.

Over time, through incremental effort, culture can change. The Apple flaw, and the Heartbleed bug discovered in OpenSSL in April 2014—after this article was originally drafted—could have been prevented by the same unit-testing approach that my Testing Grouplet (http://mike-bland.com/tags/testing-grouplet.html), Test Certified,[5] Testing on the Toilet, and Test Mercenary partners in crime worked so hard to demonstrate to Google engineering over the course of several years. By the time we finished, thorough unit testing had become the expected cultural norm. (My commentary on Heartbleed, with working code, is available at http://mike-bland.com/tags/heartbleed.html.)

Culture change isn't easy, but it's possible. If like-minded developers band together across teams, across companies, even across the industry—such as is beginning to happen with the Automated Testing Boston Meetup (http://www.meetup.com/Automated-Testing-Boston/), its sister groups in New York, San Francisco, and Philadelphia, and the AutoTest Central community blog (http://autotestcentral.com/)—and engage in creative pursuits to raise awareness of such issues and their solutions, change will come over time.

The goal is to have this and other articles—including my "Goto Fail, Heartbleed, and Unit Testing Culture" article published by Martin Fowler (http://martinfowler.com/articles/testing-culture.html)—drive discussion around the Apple SSL and Heartbleed bugs, spreading awareness and improving the quality of discourse; not just around these specific bugs, but around the topics of unit testing and code quality in general. These bugs are a perfect storm of factors that make them ideal for such a discussion:

- The actual flaw is very obvious in the case of the Apple bug, and the Heartbleed flaw requires only a small amount of technical explanation.
- The unit-testing approaches that could have prevented them are straightforward.
- User awareness of the flaws and their severity is even broader than other well-known software defects, generating popular as well as technical press.
- The existing explanations that either dismiss the ability of unit testing to find such bugs or otherwise excuse the flaw are demonstrably unsound.

If we do not seize upon these opportunities to make a strong case for the importance and impact of automated testing, code quality, and engineering culture, and hold companies and colleagues accountable for avoidable flaws, how many more preventable, massively widespread vulnerabilities and failures will occur? What fate awaits us if we do not take appropriate corrective measures in the wake of `goto fail` and Heartbleed? How long will the excuses last, and what will they ultimately buy us?

And what good is the oft-quoted bedrock principle of open-source software, Linus's Law—"Given enough eyeballs, all bugs are shallow"—if people refuse to address the real issues that lead to easily preventable, catastrophic defects?

I have worked to produce artifacts of sound reasoning based on years of experience and hard evidence—working code in the form of the Apple patch-and-test tarball and heartbeat_test.c, contributed to OpenSSL (http://goo.gl/1F7SKs)—to back up my rather straightforward claim: a unit-testing culture most likely could have prevented the catastrophic `goto fail` and Heartbleed security vulnerabilities.

High-profile failures such as the Apple SSL/TLS vulnerability and the Heartbleed bug are prime opportunities to show the benefits of automated testing in concrete terms; to demonstrate technical approaches people can apply to existing code; and to illustrate the larger, often cultural, root causes that produce poor habits and bugs. Given the extent to which modern society has come to depend on software, the community of software practitioners *must* hold its members accountable, however informally, for failing to adhere to fundamental best practices designed to reduce the occurrence of preventable defects—and must step forward not to punish mistakes but to help address root causes leading to such defects. *If you see something, say something!*

Further reading. This article is based on my presentation, "Finding More than One of the Same Worm in the Apple" (http://goo.gl/F0URUR), corresponding material,[5–8] and excerpts from my blog post (http://mike-bland.com/2014/04/15/goto-fail-tott.html).

Figure 5 is by Catherine Laplace, based on the author's sketch of an image from the Testing Grouplet/EngEDU Noogler Unit Testing lecture slides for new Google engineers.

Acknowledgments

My deepest gratitude to former Google colleagues, new associates from the Automated Testing Boston Meetup, and generous acquaintances whom I've met only online: David Plass, Isaac Truett, Stephen Vance, RT Carpenter, Gleb Bahmutov, Col Willis, Chris Lopez, and Antoine Picard. Thanks to Sarah Foster of the AutoTest Meetups and the AutoTest Central blog for providing a forum to discuss this issue. Thanks to Guido van Rossum, George Neville-Neil, and Martin Fowler for supporting my efforts.

Related articles on queue.acm.org

Security is Harder than You Think
John Viega and Matt Messier
http://queue.acm.org/detail.cfm?id=1017004

Nine IM Accounts and Counting
Joe Hildebrand
http://queue.acm.org/detail.cfm?id=966720

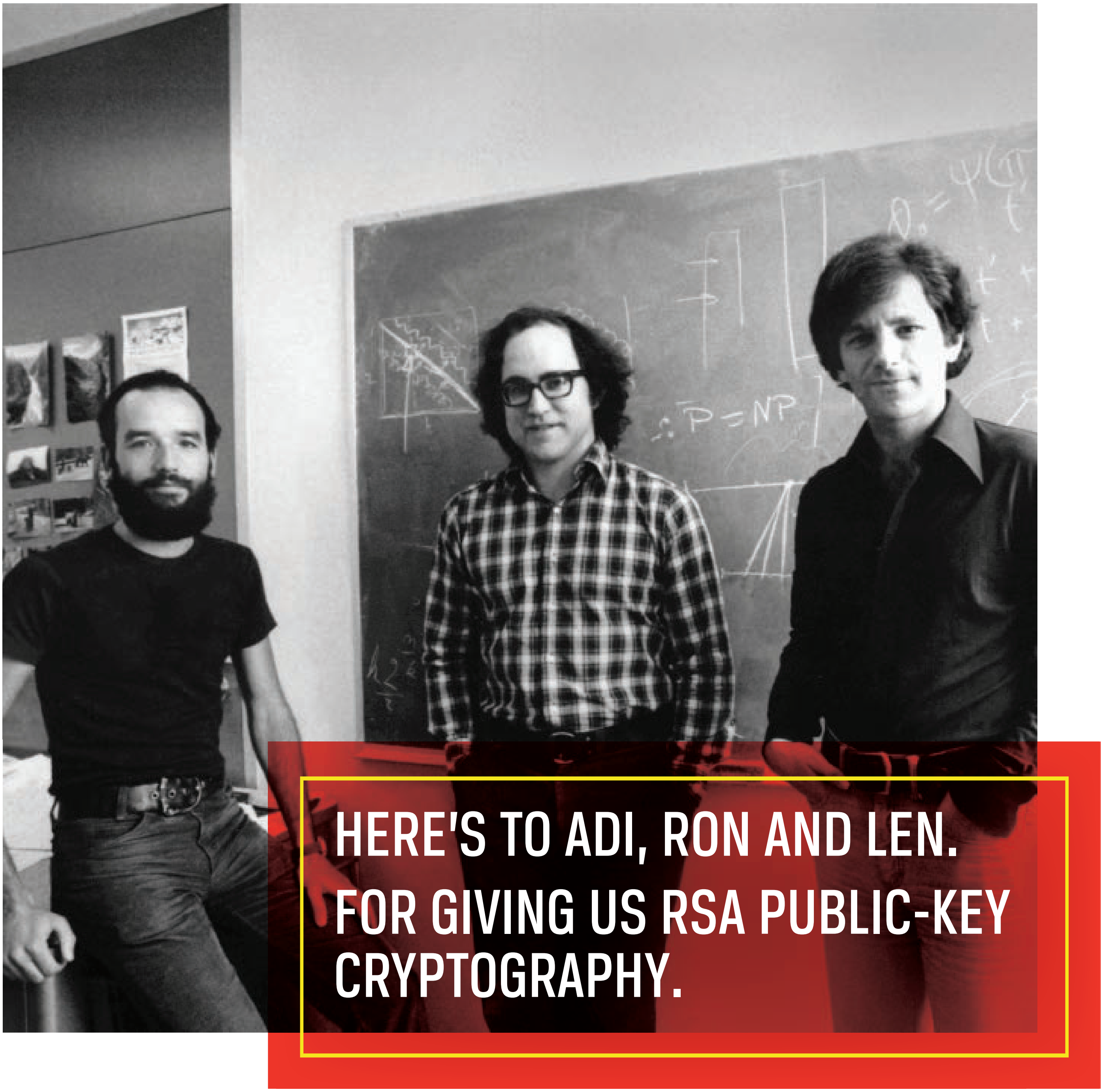
Browser Security Case Study: Appearances Can Be Deceiving
Jeremiah Grossman, Ben Livshits, Rebecca Bace and George Neville-Neil
http://queue.acm.org/detail.cfm?id=2399757

References

1. Apple Inc. Xcode overview, 2014; http://bit.ly/1kXUAzD
2. Arthur, C. Apple's SSL iPhone vulnerability: How did it happen, and what next? *The Guardian*, (Feb. 25, 2014); http://www.theguardian.com/technology/2014/feb/25/apples-ssl-iphone-vulnerability-how-did-it-happen-and-what-next.
3. Auerbach, D. An extraordinary kind of stupid. *Slate* (Feb. 25, 2014); http://slate.me/1o75yGs
4. Bellovin, S.M. Goto Fail. *SMBlog* (Feb. 23, 2014); https://www.cs.columbia.edu/~smb/blog/2014-02/2014-02-23.html.
5. Bland, M. Test Certified, 2011; http://mike-bland.com/2011/10/18/test-certified.html.
6. Bland, M. Testing on the Toilet, 2011; http://mike-bland.com/2011/10/25/testing-on-the-toilet.html.
7. Bland, M. Test Mercenaries, 2012; http://mike-bland.com/2012/07/10/test-mercenaries.html.
8. Bland, M. AutoTest Central, 2014; http://autotestcentral.com/small-medium-and-large-test-sizes
9. Dijkstra, E. A case against the GO TO statement. *Commun. ACM 11*, 3 (Nov. 1968), 147–148; http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD215.PDF.
10. Fuller, L. TestableSecurity: demonstrating that `SSLVerifySignedServerKeyExchange()` is trivially testable, 2014; https://github.com/landonf/Testability-CVE-2014-1266.
11. Google, Inc. Too many tests. *Google Testing Blog* (Feb. 21, 2008); http://googletesting.blogspot.com/2008/02/in-movie-amadeus-austrian-emperor.html.
12. Greenfield, R. Why Apple's power cords keep breaking. *The Wire* (July 30, 2012); http://www.thewire.com/technology/2012/07/why-apples-power-cords-keep-breaking/55202/.
13. Langley, A. Apple's SSL/TLS bug. *Imperial Violet* (Feb. 22, 2014); https://www.imperialviolet.org/2014/02/22/applebug.html.
14. Ray, C.K. TDD and signed SSLVerifySignedServerKeyExchange. *Exploring Agile Solutions: Software Development with Agile Practices* (Feb. 23, 2014); http://agilesolutionspace.blogspot.com/2014/02/tdd-and-signed-sslverifysignedserverkey.html.
15. Schneier, B. Was the iOS SSL flaw deliberate? *Schneier on Security: A Blog Covering Security and Security Technology* (Feb. 2014); https://www.schneier.com/blog/archives/2014/02/was_the_ios_ssl.html.
16. van Deursen, A. Learning from Apple's #gotofail security bug. *Arie van Deursen: Software Engineering in Theory and Practice* (Feb. 22, 2014); http://avandeursen.com/2014/02/22/gotofail-security/.

Mike Bland was a software engineer at Google from 2005 to 2011. Prior to working on Web-search infrastructure, he led the Testing and Fixit Grouplets; was a member of the Test Mercenaries, Testing Tech, and Build Tools teams; and was instrumental in bringing about the engineering culture changes that made thorough developer testing the accepted cultural norm. He does not represent Google in any capacity and is a student at Berklee College of Music. http://mike-bland.com/.

Credit: Ron Rivest


P=NP
HERE'S TO ADI, RON AND LEN.
FOR GIVING US RSA PUBLIC-KEY CRYPTOGRAPHY.

DOI:10.1145/2622628.2622633

The Caltech CSN project collects sensor data from thousands of personal devices for real-time response to dangerous earthquakes.

BY MATTHEW FAULKNER, ROBERT CLAYTON, THOMAS HEATON, K. MANI CHANDY, MONICA KOHLER, JULIAN BUNN, RICHARD GUY, ANNIE LIU, MICHAEL OLSON, MINGHEI CHENG, AND ANDREAS KRAUSE

Community Sense and Response Systems: Your Phone as Quake Detector

THE PROLIFERATION OF smartphones and other powerful sensor-equipped consumer devices enables a new class of Web application: community sense and response (CSR) systems, distinguished from standard Web applications by their use of community-owned commercial sensor hardware. Just as social networks connect and share human-generated content, CSR systems gather, share, and act on sensory data from users' Internet-enabled devices. Here, we discuss the Caltech Community Seismic Network (CSN) as a prototypical CSR system harnessing accelerometers in smartphones and consumer electronics, including the systems and algorithmic challenges of designing, building, and evaluating a scalable network for real-time awareness of dangerous earthquakes.

Worldwide, approximately two million Android and iOS devices have been activated every day since 2012, each carrying numerous sensors and

» key insights

- Sensors in smartphones and consumer hardware allow scientists to sense the physical world at unprecedented scale and detail.
- CSR systems partner with volunteers to understand and respond to events like natural disasters that could significantly affect even entire cities.
- CSR systems must provide fast, reliable response despite their reliance on large numbers of (potentially unreliable) consumer sensors.

high-speed Internet connection. Several recent sensing projects have sought to partner with the owners of these and other consumer devices to collect, share, and act on sensor data about phenomena that could affect millions of lives. Coupled with cloud computing platforms, these networks can achieve vast coverage previously beyond the reach of sensor networks.[6] Boulos et al.[5] includes an excellent overview of how the social and mobile Web facilitate crowdsourcing data from individuals and their sensor devices. Additional applications of community and participatory sensing include understanding traffic flows,[4,14,16,20] identifying sources of environmental pollution,[1,2] monitoring public health,[18] and responding to natural disasters like hurricanes, floods, and earthquakes.[8,9,11,15] These systems are made possible through volunteer sensors and low-cost Web solutions for data collection and storage. However, as the systems mature, they will undoubtedly extend beyond data collection and take real-time action on behalf of the community; for example, traffic networks can reroute traffic around a crash, and a seismic network can automatically slow trains to prevent derailment.

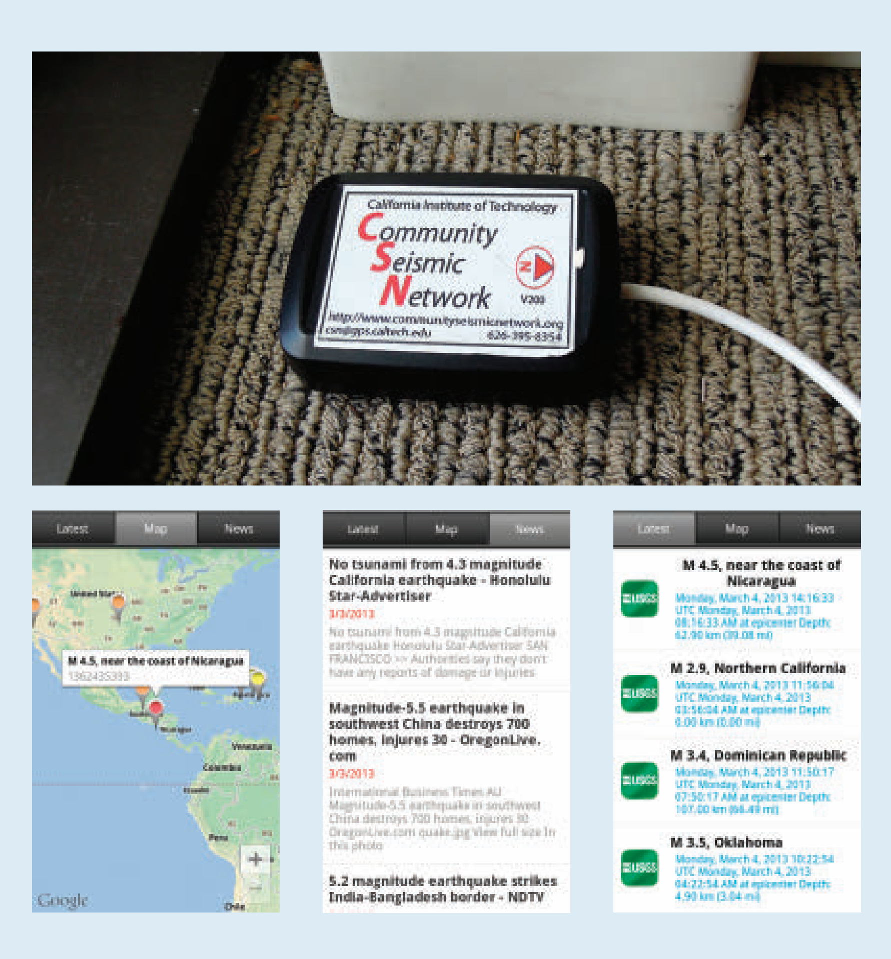
ILLUSTRATION BY HITANDRUN

From Collection to Action

Acting on community sensor data is fundamentally different from acting on data in standard Web applications or scientific sensors. The potential volume of raw data is vast, even by the standards of large Web applications. Data recorded by community sensors often includes signals produced by the people operating them. And many of the desired applications involve understanding physical phenomena at a finer scale than that of previous scientific models.

A CSR network can produce an enormous volume of raw data. Smartphones and other consumer devices often have multiple sensors and can produce continuous streams of GPS position, acceleration, rotation, audio, and video data. Even if events of interest like traffic accidents, earthquakes, and disease outbreaks are rare, devices must still monitor continuously to detect them. Beyond obvious data heavyweights like video, rapidly monitoring even a single accelerometer or microphone produces hundreds of megabytes per day. Community sensing makes possible networks with tens of thousands or even millions of devices; for example, equipping taxis with GPS devices or air-quality sensors can easily yield a network of 50,000 sensors in a big city like Beijing. At this scale, collecting even a small set of summary statistics is daunting; if 50,000 sensors report a brief status update once per minute, the total number of messages

Figure 1. CSN volunteers contribute data from low-cost accelerometers (above) and from Android smartphones via a CSN app (below).


California Institute of Technology
Community Seismic Network
V200
http://www.communityseismicnetwork.org
csn@gps.caltech.edu
626-395-8354
Latest
Map
News
Google
M 4.5, near the coast of Nicaragua
1362435393
Venezuela
Colombia
Peru
Chile
USGS

No tsunami from 4.3 magnitude California earthquake - Honolulu Star-Advertiser
3/3/2013
No tsunami from 4.3 magnitude California earthquake Honolulu Star-Advertiser SAN FRANCISCO >> Authorities say they don't have any reports of damage or injuries

Magnitude-5.5 earthquake in southwest China destroys 700 homes, injures 30 - OregonLive.com
3/3/2013
International Business Times AU Magnitude-5.5 earthquake in southwest China destroys 700 homes, injures 30 OregonLive.com quake.jpg View full size In this photo

5.2 magnitude earthquake strikes India-Bangladesh border - NDTV

M 4.5, near the coast of Nicaragua
Monday, March 4, 2013 14:16:33 UTC Monday, March 4, 2013 08:16:33 AM at epicenter Depth: 42.90 km (39.08 mi)

M 2.9, Northern California
Monday, March 4, 2013 11:56:04 UTC Monday, March 4, 2013 03:56:04 AM at epicenter Depth: 0.00 km (0.00 mi)

M 3.4, Dominican Republic
Monday, March 4, 2013 11:50:17 UTC Monday, March 4, 2013 07:50:17 AM at epicenter Depth: 107.00 km (66.49 mi)

M 3.5, Oklahoma
Monday, March 4, 2013 10:22:54 UTC Monday, March 4, 2013 04:22:54 AM at epicenter Depth: 4.90 km (3.04 mi)

would rival the daily load in the Twitter network.

Community devices also differ from their counterparts in traditional scientific and industrial applications. Beyond simply being less accurate than "professional" sensors, community sensors may be mobile, intermittently available, and affected by the unique environment of an individual user's home or workplace; for example, the accelerometer in a smartphone could measure earthquakes but user motion as well.

By enabling sensor networks that densely cover cities, community sensors make it possible to measure and act on a range of important phenomena, including traffic patterns, pollution, and natural disasters. However, due to the existing lack of fine-grain data about these phenomena, CSR systems must simultaneously learn about the phenomena they are built to act on; for example, a community seismic network may need models learned from frequent, smaller quakes to estimate damage during rare, larger quakes.

Such challenges are complicated by the need to make reliable decisions in real time with performance guarantees; for example, choosing the best emergency-response strategies following a natural disaster could be aided by real-time sensor data. However, false alarms and inaccurate data can be costly. Rigorous performance estimates and system evaluations are prerequisites for automating real-world responses.

Caltech Community Seismic Network

The CSN project at Caltech (http://csn.caltech.edu/) aims to quickly detect earthquakes and provide real-time estimates of their effects through community-operated sensors. Large earthquakes are among the few scenarios that can threaten an entire city. CSN is built on a vision of people sharing accelerometer data from their personal devices to collectively produce the information needed for real-time and post-event responses to dangerous earthquakes. To that end, it has partnered with more than 1,000 volunteers in the Los Angeles area and others in cities around the world contributing real-time acceleration data from their Android smartphones and low-cost USB-connected sensors (see Figure 1).

Following an earthquake, firefighters, medical teams, and other first responders must build situational awareness before they are able to deploy their resources effectively. Due to variations in ground structure, two points that may be only, say, a kilometer apart can experience significantly different levels of shaking and damage (see Figure 2). Likewise, different buildings may receive different degrees of damage due to the types of motion they experience. If communication is lost in a city, it can take up to an hour for helicopter surveillance to provide the first complete picture of the damage it has sustained. Fortunately, as sensors can detect the moderate P-wave shaking that precedes damaging S-wave shaking, they are likely to report initial quake measurements before the communication or power networks are compromised. These measurements can provide localized estimates of shaking intensity and damage to emergency responders immediately after a quake strikes.

Another intriguing application is early warning of strong shaking. Early warning follows the principle that accelerometers near the origin of an earthquake detect initial shaking before locations further from the origin experience strong shaking. While the duration of warning people receive depends on the speed of detection and their distance from the epicenter, warning times of tens of seconds to a minute have been produced by early-warning systems in Japan, Mexico, and Taiwan. These warnings give time needed to evacuate elevators, stop trains, or halt medical procedures. Following the 1989 Loma Prieta earthquake in Northern California, emergency workers involved in clearing debris received advance warning of aftershocks.

Community participation is ideal for seismic sensing for several reasons: First, community participation makes possible the densely distributed sensors needed to accurately measure shaking throughout a city; for example, instrumenting the greater Los Angeles area at a spatial resolution of one sensor per square kilometer would require more than 10,000 sensors. While traditional seismometer stations cost thousands of dollars per sensor to install and operate, the same number of sensors would be possible if only 0.5% of the area's population volunteered data from their smartphones. This way, community sensors can provide fine spatial coverage and complement existing networks of sparsely deployed, high-quality sensors.

Community sensors are also ideally situated for assisting the population through an emergency. In addition to collecting accelerometer data, community-sensing software on smartphones can report the last known location of family members or give instructions on where to gather for help from emergency teams; that is, community-sensing applications represent a new way for people to stay informed about the areas and people they care about.

CSN makes it easy for volunteers to participate through low-cost accelerometers and the sensors already in their Android phones. A free Android app called CrowdShake (http://csn.caltech.edu/crowdshake) makes volunteering data as easy as installing a new app. CSN also partners with Los Angeles-area schools and civic organizations to freely distribute 3,000 low-cost accelerometers from Phidgets, Inc. (http://www.phidgets.com) that interface through USB to a host PC, tablet, or other Internet-connected device. Phidgets sensors have also been installed in several high-rise buildings in the Los Angeles area to measure their structural responses to earthquakes, as in Figure 1.

Reliable, real-time inference of spatial events is a core task of seismic monitoring and a prototypical challenge for any application using physical sensors. Here, we outline a methodology developed by the CSN team to quickly detect quakes from thousands of community sensors, harnessing the computational power of community devices to overcome the noise in community-operated hardware and demonstrating that on-device learning yields a decentralized architecture scalable and heterogeneous even as it provides rigorous performance guarantees.

Decentralized Event Detection

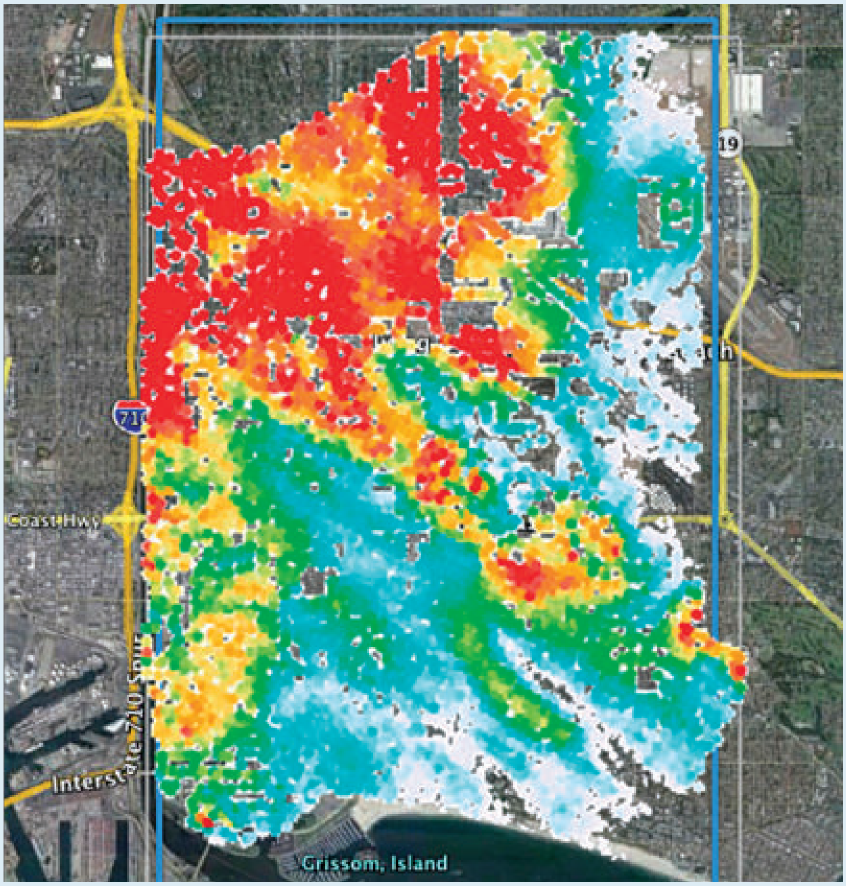
Suppose a strong earthquake begins near a metropolitan area and that 0.1% of the population contributes accelerometer data from a personally owned Internet-enabled device. In Los Angeles County, this would mean data from 10,000 noisy sensors located on a coastal basin of rock and sediment criss-crossed with fault lines and overlapped with vibration-producing freeways. How could a sensor network detect the quake and estimate its location and magnitude as quickly as possible? The "classic" approach is to collect all data centrally and declare an event has occurred when the following likelihood ratio test is true:

$$\frac{P(all\ measurements \mid quake)}{P(all\ measurements \mid quake)} > \tau \qquad (1)$$

The test would declare a detection if the ratio exceeds a predetermined threshold τ. Not surprisingly, this involves transmitting a daunting amount of data; a global network of one million phones would be transmitting 30TB of acceleration data per day. Additionally, the likelihood-ratio test requires distribution of all sensor data, conditioned on the occurrence or nonoccurrence of a strong earthquake. Each community sensor is unique, so modeling these distributions requires modeling each sensor individually.

A natural next step is a decentralized approach. Suppose each device transmits only a finite summary of its current data, or a "pick message." The central server again performs a hypothesis test but now using the received pick messages instead of the entire raw data. Results from decentralized hypothesis testing theory say if the sensors' measurements are independent, depending on whether an

Figure 2. Differences in soil conditions and subsurface structures cause significant variations in ground shaking; data recorded by the Long Beach, CA, network.



event has or has not occurred, and if the probability of the measurements is known in each case, then the asymptotically optimal strategy is to perform a hierarchical hypothesis test;[21] each sensor individually performs a hypothesis test, for some threshold τ, picking only when

$$\frac{P(\mathit{sensor\ measurement}|\mathit{quake})}{P(\mathit{sensor\ measurement}|\mathit{no\ quake})} > \tau \qquad (2)$$

Similarly, the cloud server performs a hypothesis test on the number of picks S received at a given time and declares a detection when this condition is met

$$\frac{Bin(S; r_T, N)}{Bin(S; r_F, N)} > \tau' \qquad (3)$$

The parameters r_T and r_F are the true positive and false positive pick rates for a single sensor, and $\mathrm{Bin}(\cdot, p, N)$ is the probability mass function of the binomial distribution. Decision rules (2) and (3) are asymptotically optimal for proper choice of the thresholds τ and τ'.[21] Additionally, collecting picks instead of raw data helps preserve user privacy.

Detecting rare events from community sensors presents three main challenges to this classical, decentralized detection approach:

Likelihood ratio tests. How can likelihood ratio tests be performed on each sensor's data when the data needed to accurately model sensor behavior during an event (such as measurements of large, rare quakes) is difficult to obtain?;

Modeling each sensor. How might each sensor be modeled? Serverside modeling scales poorly, and on-device learning involves computational and storage limits; and

Spatial dependencies. How can the (strong) assumption of conditionally independent sensors be overcome and spatial dependencies incorporated?

We next consider how the abundance of normal data can be leveraged to detect rare events for which training data is limited; then that new tools from computational geometry make it possible to compute the needed probabilistic models on resource-constrained devices; and finally that learning on the serverside adapts data aggregation according to spatial dependencies.

Leveraging "normal" data. The sensor-level hypothesis test in (2) requires two conditional probability distributions: The numerator models a particular device's acceleration during a strong quake that is impractical to obtain due to the rarity of large quakes. In contrast, the denominator can be estimated from abundantly available "normal" data. Is reliable quake detection still possible?

It turns out that mild assumptions, a simple approach to anomaly detection using only the probability of an acceleration time series in the absence of a quake, can obtain the same asymptotically optimal performance. A given sensor now picks when

$$\mathbb{P}(\mathit{sensor\ measurement}|\mathit{no\ quake}) < \tau \qquad (4)$$

For an appropriate choice of threshold, this technique can be shown to produce the same picks as the full hypothesis test, without requiring a model of sensor data during future unknown quakes; for more, see Faulkner et al.[11]

The anomaly-detection scheme makes use of the abundant "normal" data but still involves the challenge of computing the conditional distribution. In principle, each sensor could maintain a history of its observations, periodically estimating a probabilistic model describing that data. On a mobile device, this means logging approximately 3GB of acceleration data per month. Storing and estimating models on this much data is a burden on volunteers' smartphone resources. Is it possible to accurately model a sensor's data with (much) less storage?

The CSN system models accelerometer data as a Gaussian mixture model (GMM) over a feature vector of acceleration statistics from short-duration time windows, as in, say, phonemes in speech recognition. GMMs are a flexible family of multimodal distributions that can be estimated from data using the simple EM algorithm.[3] Unlike a single Gaussian, which is specified by the mean and variance of the data, learning the optimal GMM requires access to all the data; GMMs do not admit finite sufficient statistics. Thus, a device must store its entire sensor history to produce the optimal GMM. Fortunately, it turns out the picture is drastically different for approximating a GMM, which can be fit to an arbitrary amount of data, with an arbitrary approximation guarantee, using a finite amount of storage.

A tool from computational geometry, called a "coreset," makes such approximations possible. A coreset for an algorithm is roughly a (weighted) subset of the input, such that running the algorithm on the coreset gives a constant-factor approximation to running the algorithm on the full input. Coresets have been used to obtain approximations for a variety of geometric problems, including k-means and k-medians clustering.

It turns out that many geometric coreset techniques also provide approximations for statistical problems. Given an input dataset D, the challenge is to find the maximum likelihood estimate for the means and variances of a GMM, collectively denoted θ. A weighted set C is a (k, ϵ) coreset for GMMs if with high probability the log likelihood on $\mathcal{L}(C|\theta)$ is an ϵ approximation to the log likelihood on the full data $\mathcal{L}(C|\theta)$ for any mixture of k Gaussians:

$$(1\text{-}\epsilon)\mathcal{L}(D|\theta) \leq \mathcal{L}(C|\theta) \leq \mathcal{L}(D|\theta)(1+\epsilon) .$$

Feldman et al.[12] showed that given input D, it is possible to sample such a coreset C with size independent of the size of input D or depends polynomially on the dimension of the input, the number of Gaussians k, and parameters ϵ, δ, with probability at least $1 - \delta$ for all (nondegenerate) mixtures θ of k Gaussians. This result implies that the mixture model learned from a constant-size coreset C can obtain approximately the same likelihood as a model learned from the entire arbitrarily large D.

But where does C come from? Feldman et al.[12] showed efficient algorithms to compute coresets for projective clustering problems (such as k-means and generalizations) can provide coresets for GMMs. A key insight is that while uniformly subsampling, the input can miss "important" regions of data, an adaptive-sampling approach is likely to sample from "enough" regions to reliably estimate a mixture of k Gaussians; weighting the samples accounts for the sampling bias. Har-Peled and Mazumdar[13] identified that coresets for many optimization problems can be computed efficiently in the parallel or streaming model, with several such results applying here. In particular, a stream of input data can be buffered to some constant size, then compressed into a coreset. Careful merging and compressing of these coresets provides an approximation of the entire stream so far while using space and update time polynomial in all the parameters and logarithmic in n.

Quake detection in community networks requires finding a complex spatiotemporal pattern in a large set of noisy sensor measurements. The start of a quake may affect only a small fraction of a network, so the event can be concealed in single-sensor measurements and in networkwide statistics. Recent high-density seismic studies, as in Figure 2, found localized variations in ground structure significantly affect the magnitude of shaking at locations only one kilometer apart. Consequently, effective quake detection requires algorithms that are able to learn subtle dependencies among sensor data and detect changes within groups of dependent sensors.

The classical approach outlined earlier assumes sensors provide independent, identically distributed measurements conditioned on the occurrence or nonoccurrence of an event. In this case, the system would declare an event has occurred if a sufficiently large number of sensors, regardless of location, report picks. However, in many practical applications, the particular spatial configuration of the sensors matters, and the independence assumption is violated. How can (qualitative) knowledge about the nature of the event be exploited to improve detection performance?

How can (qualitative) knowledge about the nature of the event be exploited to improve detection performance?

The start of an event (such as an earthquake, fire, or epidemic) may first be observed by small groups of sensors that are close to the event or are most sensitive to its effects. Viewed as transmitting a vector $x \in \mathbb{R}^p$ through a noisy channel, the signal is mostly zeros (sparse), but many bits in the received vector y (picks) are flipped due to noise. Intuitively, the signal observed by these small groups will be lost among the environmental noise unless the system is aware of dependencies among the sensors. This intuition (and some desirable analytic properties) can be captured by learning an orthonormal change-of-basis matrix that projects the picks received by the server onto a coordinate system that, roughly, aggregates groups of strongly correlated sensors. Given such a matrix B with columns $b_i, \ldots, b_p$, the server declares an event when

$$\max_i b_i^T y > \tau$$

To obtain reliable detection when the signal is weak (measured by the ℓ_0 pseudo-norm, $\|x\|_0 < \sqrt{p}$), traditional hypothesis testing requires the error rate of each sensor (each element of x) to decrease as the number of sensors p increases. This requirement is in stark contrast to the intuition that more sensors are better and incompatible with the "numerous but noisy" approach of community sensing. However, Faulkner et al.[10] found that if the matrix B is "sparsifying," or $\|B^T x\|_0 = p^\beta$, $\|x\|_0 < p^\alpha$, $0 < \beta < \alpha < \frac{1}{2}$, then the test $\max_i b_i^T y > \tau$ gives probability of miss and false alarm that decays to zero exponentially as a function of the "sparsification ratio" $\|\mathrm{x}\|_0 \big/ \|B^T\|_0$, for any rate $r_F < \frac{1}{2}$ of pick errors. Effectively, a change of basis allows large numbers of noisy sensors to contribute to reliable detection of signals that are observed by only a small fraction ($\|x\|_0$) of sensors.

Learning to sparsify. These results depend on B's ability to concentrate weak signals.

A direct approach for learning B is to optimize

$$\min_B \|B^T X\|_0 \text{ subject to } BB^T = I \qquad (5)$$

where X is a matrix containing binary per-sensor event occurrences as its columns and $BB^T = I$ is the sum of non-zero elements in the matrix. The constraint ensures B remains orthonormal. Computing equation (5) can be impractical, as well as sensitive to noise or outliers in the data. It may thus be more practical to find a basis that sparsely represents "most of" the observations. More formally, let Z be a latent matrix that can be viewed as the "cause" in the transform domain of the noise-free signals X, or $X = BZ$. Z should be sparse and BZ should be close to the observed signal Y. These conditions suggest the next optimization, originally intro-

Figure 3. The CSN cloud maintains the persistent state of the network in datastore, performs real-time processing of pick data via Memcache, and generates notifications and quake statistics.

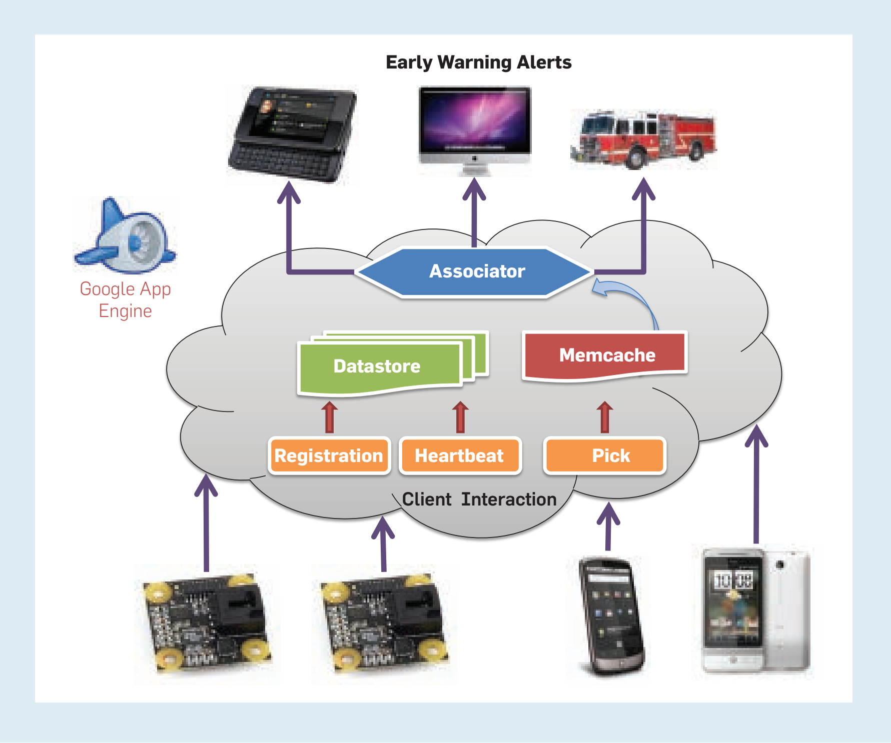
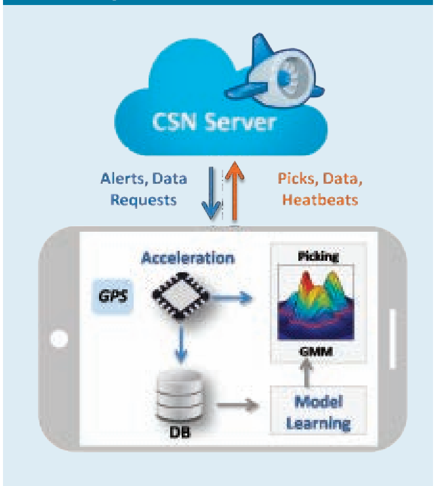


Figure 4. The CrowdShake app processes sensor data locally on an Android phone or tablet, sends pick messages during potential quakes, receives alerts, and responds to data requests.



duced for text modeling[7] as a heuristic for equation (5)

$$\min_{BZ} \|Y - BZ\|_F^2 + \lambda \|Z\|_1 \text{ subject to } BB^T = I \qquad (6)$$

where $\|\cdot\|_F$ is the matrix Frobenius norm, and $\lambda > 0$ is a free parameter. Equation (6) essentially balances the difference between Y and X with the sparsity of Z; increasing λ more strongly penalizes choices of Z that are not sparse. For computational efficiency, the ℓ_0-norm is replaced by the convex and heuristically "sparsity-promoting" ℓ_1-norm.

Although equation (6) is non-convex, fixing either B or Z makes the objective function with respect to the other convex. The objective can then be efficiently solved (to a local minima) through an iterative two-step convex-optimization process.

Building CSN

Managing a CSR and processing its data in real time is a challenge in terms of scalability and data security. Cloud computing platforms (such as Amazon EC2, Heroku, and Google App Engine) provide practical, cost-effective resources for reliably scaling Web applications. The CSN network is built on Google App Engine (see Figure 3). Heterogeneous sensors include cellphones, standalone sensors, and accelerometers connected through USB to host computers to the cloud; the cloud, in turn, performs event detection and issues notifications of potential seismic events. Additionally, a cloud infrastructure allows sensors anywhere in the world to connect just by specifying a URL.

The CSN network includes two kinds of sensor clients: a desktop client with USB accelerometer and an Android app for phones and tablets, as in Figure 1. The internal data flow and the messaging between the cloud and an Android client are outlined in Figure 4; desktop clients differ primarily in their picking algorithm and lack of GPS (see Figure 5). At the core of the application is a suite of sensors, including three-axis accelerometer and GPS. The Android client continually tests the accelerometer data for anomalies (reported as picks), logging raw data temporarily to a local database for post-event data collection. Clients listen for push notifications from the server implemented through Google's Cloud Messaging services.

Cloud computing services are well suited for the network maintenance and real-time response tasks of CSR systems. Figure 3 outlines the main data flows through the cloud: First, CSN writes client registration and heartbeat messages to multiple datacenters via App Engine's high replication datastore. Next, incoming picks are spatially aggregated by geographic hashing into memcache, a distributed in-memory data cache. Although memcache is not persistent (objects can be ejected from the cache due to memory constraints), it is much faster than the datastore. Memcache is also ideal for computations that must occur quickly, and, because it allows values to set an expiration time, it is also ideal for data whose usefulness expires after a period of time. Finally, the CSN cloud performs event detection on the aggregated picks. Implementing this architecture on App Engine offers several advantages:

Dynamic scaling. Incoming requests are automatically load balanced between instances created and destroyed based on current demand levels, an arrangement that simplifies algorithmic development and reduces costs during idle periods;

Robust data. Datastore writes are automatically replicated to geographically separate data centers. Replication is prudent for any application but especially for disaster response systems; and

Easy deployment. Deploying applications on App Engine is fairly straightforward, as individual server instances need not be configured and coordinated. Additionally, by using the same front ends that power Google's search platform, App Engine applications can expect low latency from any geographical location in the world. Such scalability allows the network to readily include volunteers from new cities or countries.

Experimental Evaluation

Evaluating a CSR system involves assessing both hardware and algorithms. For CSN, this means determining whether community hardware is able to detect strong quakes, evaluate detection algorithms on their ability to detect future quakes that cannot be modeled or predicted, and verify that implementing the system is practical on mobile devices and cloud platforms (see Figure 6).

The CSN team evaluated community hardware, and found that low-cost MEMS accelerometers are capable of measuring seismic motion. Experiments with a large actuator called a "shake table" expose sensors to accurate reproductions of historic, moderately large (magnitude 4.5–5.5) earthquakes. The shake table demonstrates that both USB sensors and the lower-quality phone accelerometers can detect the smaller initial shaking (P-wave) and stronger secondary shaking (S-wave) that produce the characteristic signature of an earthquake (see Figure 7). These laboratory experiments are confirmed by measurements of actual earthquakes observed by the CSN network; Figure 5 shows similar signatures involving a subset of measurements of a magnitude 3.6 quake. A second experiment assesses whether community sensors can detect changes in the motion of buildings caused by earthquakes. The CSN team oscillated the 10-story Millikan Library on the Caltech campus using a large eccentric weight on the roof. CSN sensors measured the resonant frequency of the building (approximately 1.7Hz), confirming low-cost sensors are able to perform structure monitoring.

Here, we evaluate the ability of community sensors to detect future quakes

Figure 5. CSN sensors produced picks (blue and red bars) for both P-wave and S-wave of the March 2013 Anza M4.7 earthquake; time series plots are arranged by distance from the quake's epicenter.

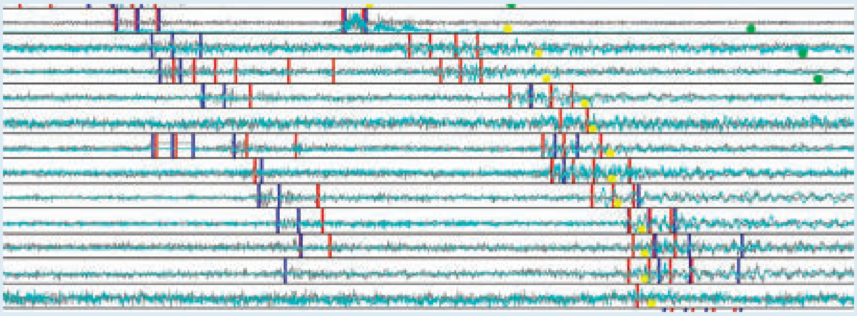


Figure 6. Eccentric weights oscillate Millikan Library, showing CSN hardware is sensitive to resonant frequencies in buildings.

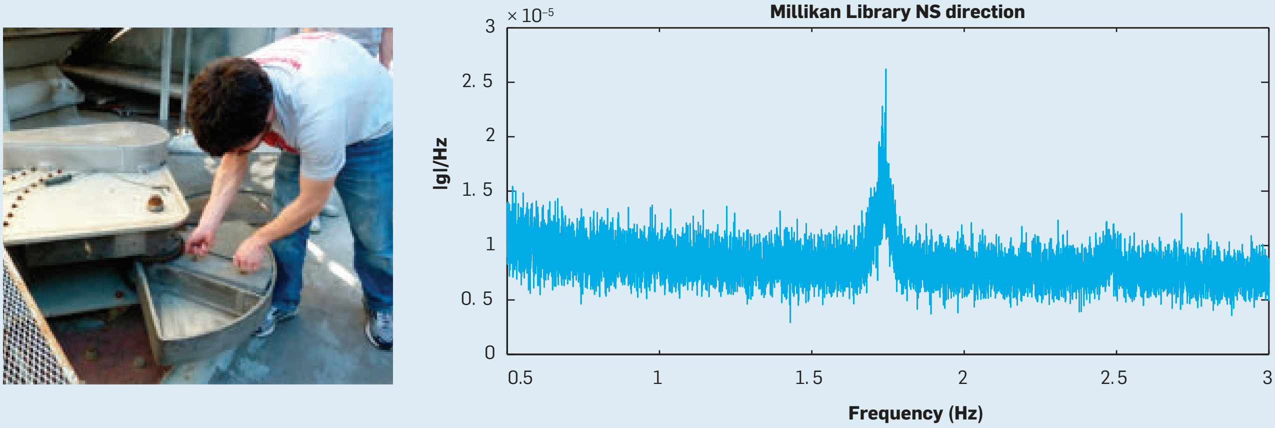


Figure 7. Android accelerometers accurately record strong shaking during a shake-table experiment: (a) shake-table experimental setup; (b) ground truth; and (c) Android phone; and (d) Android phone in backpack.

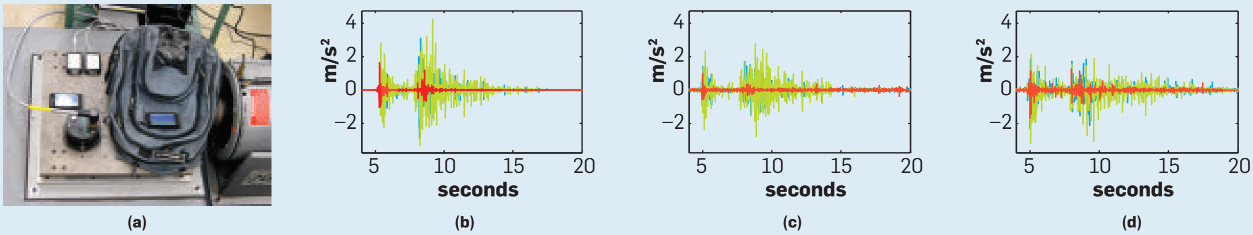
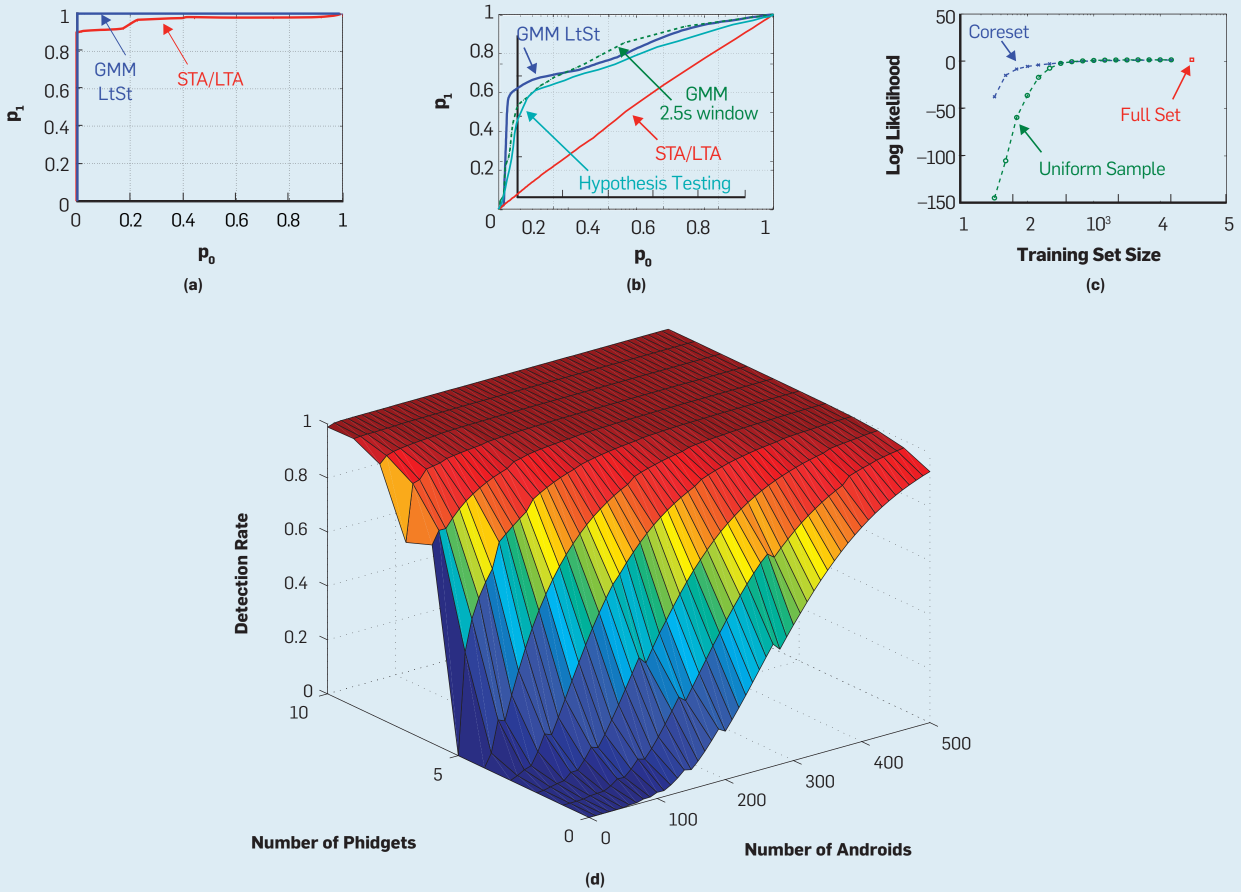
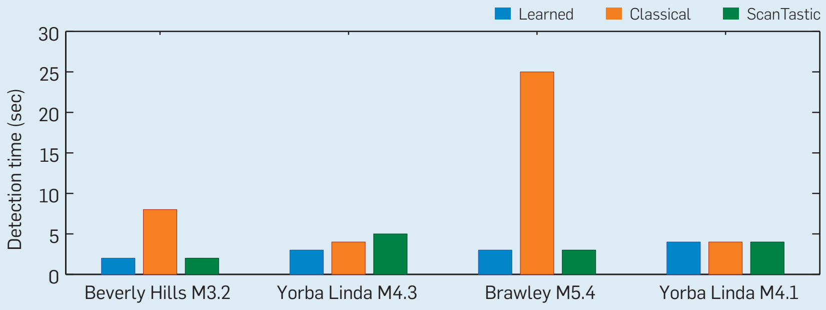


Figure 8. Attainable true-positive and false-positive pick rates for: (a) USB accelerometer and (b) Android accelerometer; (c) coresets of accelerometer data require less storage to produce accurate acceleration models; and (d) estimated quake detection rates for a mixture of USB and mobile phone sensors in a given area.



for which no training data is available. While earthquakes are rare, data gathered from community sensors may be plentiful. To characterize "normal" (background) data, seven student and faculty volunteers at Caltech carried Android phones during their daily routines to gather more than 7GB of phone accelerometer data; 20 desktop USB accelerometers recorded 55GB of acceleration. From this data, the CSN team estimated models for each sensor type's normal operating behavior, evaluating anomaly-detection performance on 32 historic records of moderately large (magnitude 5–5.5) events (as recorded by the Southern California Seismic Network (http://www.scsn.org). Individual sensors were able to transmit "event" or "no event" to the cloud server in the form of receiver operating characteristic curves, showing anomaly detection outperforming several standard baselines (see Figure 8). The vertical axis is the attainable detection (pick) rate of a single sensor against the horizontal axis of allowable false detection (pick) rate.

Figure 9. Learned "sparsifying" outperforms standard spatial event-detection algorithms and provides faster detection of four Los Angeles area quakes recorded by CSN in 2012.



Coresets provide a promising way to learn GMMs of accelerometer data on smartphones, yielding more accurate models for a given training set size than uniformly subsampling the available data. Combining the accuracy results for USB and Android sensors, Figure 8d outlines the trade-off of detecting with a mix of sensor types while limited to one false alarm per year. These results indicate approximately 50 phones or 10 Phidgets should be enough to detect a nearby magnitude 5 or larger event with close to 100% detection rate.

While earthquakes are inherently unpredictable, simulations provide a qualitative idea of spatial dependencies among sensors that can be used to train detectors. Using a prior distribution constructed from historic earthquakes in the U.S. Geological Survey database (http://www.data.scec.org) a simulator for community sensors similar to the one developed by Liu et al.[17] simulated picks from 128 CSN sensors during 1,000 simulated quakes. These picks are used as training data for a sparsifying basis, a networkwide hypothesis test, and a spatial scan statistic. Each algorithm is evaluated on its ability to detect four recent events using real measurements recorded by the network (see Figure 9); for each event, the vertical bars give the time to detection for the learned bases, classical hypothesis testing, and a competitive scan statistic algorithm. The bases learned from simple simulations in general achieve faster detection (such as eight seconds faster than competitive algorithms detecting the September 3, 2102 Beverly Hills, CA, magnitude 3.2 event).

Conclusion

We have outlined several algorithmic and systems principles that facilitate detecting rare and complex spatial signals using large numbers of low-cost community sensors. Employing machine learning at each stage of a decentralized architecture allows efficient use of sensor-level and cloud-level resources, essential for providing performance guarantees when little can be said about a particular community sensor or when little is known about the events of interest. Community sensing is applicable to a variety of application domains, including fires, floods, radiation, epidemics, and traffic accidents, as well as monitoring pollution, pedestrian traffic, and acoustic noise levels in urban environments. In all cases, "responding" can range from taking physical action to merely devoting additional resources to an event of interest. While the CSN project is motivated by the public need to detect and react to strong earthquakes, CSR systems for these domains and others will require a similar blueprint for machine learning and scalable systems.

Acknowledgments

We would like to thank the Gordon and Betty Moore Foundation, the National Science Foundation (awards CNS0932392, IIS0953413), and European Research Council Starting Grant 307036. Andreas Krause was supported in part by a Microsoft Research Faculty Fellowship. We also thank Signal Hill Petroleum and Nodal Seismic for data from the Long Beach Network, and the Southern California Seismic Network for data from the permanent earthquake network in Southern California.

References

1. Aberer, K., Sathe, S., Chakraborty, D., Martinoli, A., Barrenetxea, G., Faltings, B., and Thiele, L. Opensense: Open community driven sensing of environment. In *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming* (San Jose, CA, Nov. 2–5). ACM Press, New York, 2010, 39–42.
2. Aoki, P.M., Honicky, R.J., Mainwaring, A., Myers, C., Paulos, E., Subramanian, S., and Woodruff, A. A vehicle for research: Using street sweepers to explore the landscape of environmental community action. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems* (Boston, Apr. 4–9). ACM Press, New York, 2009, 375–384.
3. J. Bilmes. *A Gentle Tutorial on the EM Algorithm Including Gaussian Mixtures and Baum-Welch. International Computer Science Institute Technical Report TR-97-021*, May 1997.
4. Borokhov, P., Blandin, S., Samaranayake, S., Goldschmidt, O., and Bayen, A. An adaptive routing system for location-aware mobile devices on the road network. In *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems* (Washington, D.C., Oct. 5–7). IEEE Computer Society Press, New York, 2011, 1839–1845.
5. Boulos, M.N.K, Resch, B., Crowley, D.N., Breslin, J.G., Sohn, G., Burtner, R., Pike, W.A., Jezierski, E., and Chuang, K.-Y.S. Crowdsourcing, citizen sensing and sensor Web technologies for public and environmental health surveillance and crisis management: Trends, OGC standards, and application examples. *International Journal of Health Geographics 10*, 1 (2011).
6. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., and Srivastava, M.B. Participatory sensing. In the Workshop on World Sensor Web Workshop (Boulder, CO, Oct. 31–Nov. 3, 2006), 1–5.
7. Chen, X., Qi, Y., Bai, B., Lin, Q., and Carbonell, J.G. Sparse latent semantic analysis. In *Proceedings of the SIAM International Conference on Data Mining* (Mesa, AZ, Apr. 28–30). SIAM, Philadelphia, 2011, 474–485.
8. Cochran, E.S., Lawrence, J.F., Christensen, C., and Jakka, R.S. The Quake-Catcher Network: Citizen science expanding seismic horizons. *Seismological Research Letters 80*, 1 (2009), 26–30.
9. Ervasti, M., Dashti, S., Reilly, J., Bray, J.D., Bayen, A., and Glaser, S. iShake: Mobile phones as seismic sensors, user study findings. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia* (Beijing, Dec. 7–9). ACM Press, New York, 2011, 43–52.
10. Faulkner, M., Liu, A., and Krause, A. A fresh perspective: Learning to sparsify for detection in massive noisy sensor networks. In *Proceedings of the 12th ACM/IEEE International Conference on Information Processing in Sensor Networks* (Philadelphia, Apr. 8–11). ACM Press, New York, 2013, 7–18.
11. Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, K.M., and Krause, A. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks* (Chicago, Apr. 12–14). ACM Press, New York, 2011, 13–24.
12. Feldman, D., Faulkner, M., and Krause, A. Scalable training of mixture models via coresets. In *Proceedings of the Neural Information Processing Systems Annual Conference* (Granada, Spain, Dec. 12–14, 2011).
13. Har-Peled, S. and Mazumdar, S. On coresets for *k*-means and *k*-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing* (Chicago, June 13–15). ACM Press, New York, 2004, 291–300.
14. Hoh, B., Gruteser, M., Herring, R., Ban, J., Work, D., Herrera, J.C., Bayen, A.M. Annavaram, M., and Jacobson, Q. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services* (Breckenridge, CO, June 17–20, 2008), 17–20.
15. Kapoor, A., Eagle, N., and Horvitz, E. People, quakes, and communications: Inferences from call dynamics about a seismic event and its influences on a population. In *Proceedings of AAAI Symposium on Artificial Intelligence for Development* (Atlanta, July 11–15). AAAI, Palo Alto, CA, 2010, 51–56.
16. Krause, A., Horvitz, E., Kansal, A., and Zhao, F. Toward community sensing. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks* (St. Louis, MO, Apr. 22–24). IEEE Computer Society Press, Washington, D.C., 2008, 481–492.
17. Liu, A., Olson, M., Bunn, J., and Chandy, K.M. Towards a discipline of geospatial distributed event-based systems. In *Proceedings of the Sixth ACM International Conference on Distributed Event-Based Systems* (Berlin, July 16–20). ACM Press, New York, 2012, 95–106.
18. Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., and Boda, P. Peir: The personal environmental impact report as a platform for participatory sensing systems research. In *Proceedings of the Seventh International Conference on Mobile Systems, Applications, and Services* (Kraków, Poland, June 22–25). ACM Press, New York, 2009, 55–68.
19. Neill, D.B. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74 (2012), 337–360.
20. Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., and Eriksson, J. VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the Seventh ACM Conference on Embedded Networked Sensor Systems* (Berkeley, CA, Nov. 4–6). ACM Press, New York, 2009, 85–98.
21. Tsitsiklis, J.N. Decentralized detection by a large number of sensors. *Mathematics of Control, Signals, and Systems 1*, 2 1988, 167–182.

Matthew Faulkner (mfaulk@caltech.edu) is a Ph.D. candidate in computer science at Caltech, Pasadena, CA.

Robert Clayton (clay@gps.caltech.edu) is a professor of geophysics at Caltech, Pasadena, CA.

Thomas Heaton (heaton@gps.caltech.edu) is a professor of geophysics and civil engineering at Caltech, Pasadena, CA.

K. Mani Chandy (mani@cs.caltech.edu) is a professor of computer science at Caltech, Pasadena, CA.

Monica Kohler (kohler@caltech.edu) is a senior research fellow in mechanical and civil engineering at Caltech, Pasadena, CA.

Julian Bunn (julian.bunn@caltech.edu) is a principal computational scientist in the Center for Advanced Computing Research at Caltech, Pasadena, CA.

Richard Guy (rguy@gps.caltech.edu) is the Community Seismic Network project manager at Caltech, Pasadena, CA.

Annie Liu (aliu@cms.caltech.edu) is a software engineer at Facebook, Menlo Park, California; her research for this article was part of her Ph.D. in computer science at Caltech, Pasadena, CA.

Michael Olson (molson@cs.caltech.edu) is a software engineer at Google, Mountain View, CA; his research for this article was part of his Ph.D. in computer science at Caltech, Pasadena, CA.

MingHei Cheng (mmhcheng@caltech.edu) conducted research for this article as part of his Ph.D. in mechanical and civil engineering at Caltech, Pasadena, CA.

Andreas Krause (krausea@ethz.ch) is an assistant professor of computer science at ETH Zürich, Zürich, Switzerland.

DOI:10.1145/2627422

In the RaaS cloud, virtual machines trade in fine-grain resources on the fly.

BY ORNA AGMON BEN-YEHUDA, MULI BEN-YEHUDA, ASSAF SCHUSTER, AND DAN TSAFRIR

The Rise of RaaS: The Resource-as-a-Service Cloud

CLOUD COMPUTING IS taking the computer world by storm. Infrastructure-as-a-Service (IaaS) clouds (such as Amazon Elastic Compute Cloud, and EC2) allow anyone with a credit card to tap into a seemingly unlimited fountain of computing resources by renting virtual machines for several cents or dollars per hour. Forrester Research[30] predicted the cloud computing market could top $241 billion in 2020, compared to $40.7 billion in 2010, a sixfold increase. What will the 2020 clouds look like? Given the pace of innovation in cloud computing and other utilities (such as smart grids and wireless spectra), substantial shifts are bound to occur in the way providers design, operate, and sell cloud computing resources and how clients purchase and use them.

IaaS cloud providers sell fixed bundles of CPU, memory, and I/O resources packaged as server-equivalent virtual machines called guest machines. We foresee providers will continuously update the price and quantity of the individual resources in time granularity as fine as seconds and the software stack within the virtual machines will evolve accordingly to operate in this dynamic environment. We call this new model of cloud computing the Resource-as-a-Service (RaaS) cloud. In it, provider-governed economic mechanisms will control clients' access to resources. Clients will thus deploy economic software agents that will continuously buy and sell computing resources in accordance with the provider's current supplies and other clients' demands.

IaaS Trends

We identify four existing trends in the operation of IaaS cloud computing platforms that underlie the transition we foresee: the shrinking duration of rental, billing, and pricing periods; the constantly decreasing granularity of resources offered for sale; the increasingly market-driven pricing of resources; and the provisioning of useful service level agreements (SLAs).

Duration of rent and pricing. Before cloud computing emerged over the past decade, the useful lifetime of a purchased server was several years. With the advent of Web hosting, clients could rent a server on a monthly basis. With the introduction of on-demand EC2 instances in 2006, Amazon radically changed the time granularity of server rental, making it possible for its clients to rent a "server equivalent" for as short a period as one hour. This

» key insights

- Current trends toward increased flexibility and efficiency in IaaS clouds will force a business model paradigm shift on the cloud storage industry.
- The RaaS cloud is the likely result of this shift.
- The RaaS cloud uses economic mechanisms within physical machines.

IMAGE COLLAGE BY ANDRIJ BORYS ASSOCIATES

move was good for the provider—Amazon—because, by incentivizing clients to shut down unneeded instances, the hardware was time-shared better. It also benefited clients, who no longer had to pay for wall-clock time they did not use but only for instance time they did use.

Renting server-equivalents for ever-shorter periods is driven by economic forces that keep pushing clients to improve efficiency and minimize waste: if a partial instance-hour is billed as a full hour, a client might waste up to one hour over the lifetime of every virtual machine (a per-machine penalty). If a partial instance-second were billed as a full second, then the client would waste only up to one second over the lifetime of each virtual machine. Shorter periods of rent and shorter billing units reduce client overhead, opening the cloud for business for shorter workloads. Low overhead encourages horizontal elasticity—changing the number of concurrent virtual machines—and draws clients that require this functionality to the cloud.

The trend toward shorter times is also gaining ground with regard to pricing periods. Amazon spot-instances, announced in 2009, may be repriced as often as every five minutes,[1] although Amazon bills by the price at the beginning of the hour. CloudSigma, an IaaS cloud provider launched in 2010, reprices its resources exactly every five minutes;[a] see the sidebar "For More on RaaS."

Newer providers charge by even finer time granularity; for example, Gridspot[b] and ProfitBricks,[c] both launched in 2012, charge by three-minute and one-minute chunks, respectively. Meanwhile, Google modified its pricing policies; as of June 2011, Google App Engine bills instances by the minute, with a minimum charge of 15 minutes,[d] and as of May 2013 Google Compute Engine charges by the minute, with a minimum of 10 minutes, instead of by the hour.[e]

We draw an analogy between cloud providers and phone companies that have progressed from billing landlines per several minutes to billing cellphones by the minute and then, due to client pressure or court order, to billing per several seconds and even per second. Likewise, car rental (by the day) is also giving way to car sharing (by the hour), and the U.S. President's Council of Advisors on Science and Technology recommends that wireless spectrum sharing have a shorter period base.[18]

We expect this trend of shortening times to continue, so, eventually, cloud providers will reprice computing resources every few seconds and charge for them by the second. Providers might compensate themselves for overhead by charging a minimal amount or using progressive prices (higher unit prices for shorter rental times). Such durations are consistent with peak demand that can change over seconds when a site is "slashdotted" (linked from a high-profile website).[f]

Resource granularity. In most IaaS clouds, clients rent a fixed bundle of compute, memory, and I/O resources. Amazon and Rackspace[g] call these bundles "instance types"; GoGrid[h] calls them "server sizes"; and Google Compute Engine[i] calls them "machine types." Selling resources this way gives clients a familiar abstraction of a server equivalent. However, this abstraction is starting to unravel, and in its place are the beginnings of a new trend toward finer resource granularity. In August 2012,[j] Amazon began allowing clients to dynamically change available I/O resources for already-running instances.[k] Google App Engine charges for I/O operations by the million and offers progressive network prices rounded down to small base units before charging (such as 1B, one email message, and one instance-hour).[l] CloudSigma (2010), Gridspot (2012), and ProfitBricks (2012) offer their clients the ability to compose a flexible bundle from varying amounts of resources, similar to building a custom-made server from different combinations of resources (such as CPUs, memory, and I/O devices).

Renting a fixed combination of cloud resources does not reflect the interests of clients. First, as server size is likely to continue to increase (hundreds of cores and hundreds of gigabytes of memory per server in the next few years), an entire server equivalent may be too large for some customer needs. Second, selling a fixed combination of resources is only efficient when the load customers need to handle is both known in advance and constant. As neither condition is likely, the ability to dynamically mix and match different amounts of compute, memory, and I/O resources benefits clients.

We expect this trend toward increasingly finer resource granularity to continue, so all major resources—compute, memory, and I/O—will be rented and charged for in dynamically changing amounts, not in fixed bundles; clients will buy seed virtual machines with some initial amount of resources, supplementing them with additional resources as needed.

Following these trends, we extrapolate that resources in the near future will be rented separately with fine resource granularity for short periods. As rental periods grow shorter, we expect efficient clients to automate the process by deploying an economic software agent to make decisions in accordance with the current prices of the resources, the changing load the machine should handle, and the client's subjective valuation of the different resources at different times. Such agents are also considered a necessary development in smart grids[29] and in wireless spectrum[40] resource allocation. Two elements are likely to ease adoption of economic agents: client size, whereby larger clients are more likely to invest in systematic ways to save money, which accumulates for them to large amounts, and availability of off-the-shelf and customizable agents (such as open source ones).

Market-driven resource pricing. Virtualization and machine consolidation are beneficial when at least some resources are shared (such as heat sink, bus, and last-level cache) and others are time-shared (such as when a fraction of a CPU is rented or physical memory is overcommitted). However, the performance of a given virtual machine can vary wildly over time due to interference and bottlenecks caused by other virtual machines that share resources whose use is not measured and allocated;[15,24,34] for example, Google App Engine's preliminary model—charging for CPU time only and not for memory—made the scaling of applications that use a lot of memory and little CPU time "cost-prohibitive to Google,"[m] because consolidation of such applications was hindered by memory bottlenecks. In 2011, Google App Engine was thus driven to charge for memory (by introducing memory-varied bundles). As a result, memory became a measured and allocated resource.

Moreover, interference and bottlenecks depend on the activity of all the virtual machines in the system and are not easily quantified in a live environment in which guests can monitor only their own activity. Even after the guest machine benchmarks its performance as a function of the resource bundle it rented, neighbors sharing the same resources might still cause performance to vary.[34] There is thus a discrepancy between what providers provide and what clients actually prefer; in practice, what clients care about is the subjective performance of their virtual machines.

To bridge this gap, researchers have proposed selling *guest performance* instead of consumed resources.[5,17,24,26] This approach is applicable only where performance is well defined and client applications are fully visible to the provider, as in Software-as-a-Service and Platform-as-a-Service clouds, or the client virtual machines fully cooperate with the provider, as can happen in private IaaS clouds. However, IaaS cloud providers and clients are separate economic entities and do not in general trust one another or cooperate without good reason. Guaranteeing client performance levels is thus not applicable to a public IaaS cloud where allocated resources affect the performance of different applications differently, the very definition of performance is subjective, client virtual machines are opaque, and the provider cannot rely on clients to tell the truth with regard to their desired and achieved performance. If the provider guarantees a certain performance level, it is in the client's interest to claim the performance is still too low to motivate the provider to add resources.

Public clouds will have to forsake charging users a predefined sum for resource bundles of unknown performance. For high-paying clients, providers can raise prices and forgo overcommitting resources. For low-paying clients, a cheap or free tier of unknown performance can be offered. However, for mid-range clients, providers will have to follow one of two possible routes to address the problem of unpredictable resource availability: precisely measure all system resources to quantify the real use each virtual machine makes of them and then charge the clients precisely for the resources they consumed; or switch to a market-driven model.

A market-driven model is based on how clients value the few monitored resources. It does not necessitate precise measurement of resource use on the part of the provider; only the final outcome matters—the client's subjective valuation of the performance. Clients, in turn, will have to develop their own model to determine the value of a smaller number of monitored resources. Such a model should implicitly factor in virtual-machine interference over non-monitored resources; for example, clients might use a learning algorithm that produces a time-local model of the connection between monitored resources and client performance. Though highly expressive, the client's model need not be complicated; it is enough that the client can adjust the model to the required accuracy level. The minimal client model can thus be as simple as a specific sum for a specific amount of resources; below these requirements, the client will not pay at all, and above them, the client will not pay more. The client's willingness to pay affects prices and resource allocation. Unlike previously proposed models,[5,17,24,26] this economic model can accommodate real-world, selfish, rational clients.

Tiered service. Tiered service,[25] in which different clients get different levels of service, is found in certain scientific grids. Jobs of low-priority clients may be preempted (aborted or suspended) by jobs of high-priority clients. Although a decade ago clouds did not offer such prioritized service but supplied service at only one fixed level—on-demand—Amazon has since introduced various priority levels within EC2. Higher priority levels are accorded to reserved (introduced March 2009) and on-demand instances. Spot instances (introduced December 2009) provide a continuum of lower service levels, since Amazon prioritizes spot instances according to the price bid by each client. Gridspot (2012) operates in a similar way. As in grids, these priorities are relative, so it is difficult to explicitly define their meaning in terms of absolute availability; for example, availability of on-demand instances depends on demand for reserved instances. The PaaS provider Docker (announced in 2010 as dotCloud)[n] and Google App Engine[o] also offer different SLA levels at different fee levels.

Providers that prioritize clients can provide high-priority clients with elasticity and availability at the expense of lower-priority clients while simultaneously renting out currently spare resources to low-priority clients when high-priority clients do not need them. Likewise, different priorities allow budget-constrained cloud clients inexpensive access to computing resources with poorer availability assurances. Mixing high-priority and low-priority clients will allow providers to simultaneously achieve high resource utilization and maintain adequate spare capacity for handling sudden loads.

More on RaaS

To delve further into the trends behind our vision of the RaaS cloud, see:

[a] http://www.cloudsigma.com
[b] http://gridspot.com
[c] http://www.profitbricks.com
[d] https://developers.google.com/appengine/kb/billing#time_granularity_instance_pricing
[e] https://cloud.google.com/pricing/compute-engine
[f] "50% of the time the site is down in seconds, even when we've contacted site owners and they've told us everything will be fine. It's often an unprecedented amount of traffic, and they don't have the required capacity." Stephen Fry, actor and widely followed Twitter user, London, U.K.; http://tinyurl.com/StephenFrySeconds
[g] http://www.rackspace.com/cloud/public/servers/techdetails/
[h] http://www.gogrid.com
[i] https://cloud.google.com/pricing/compute-engine
[j] http://aws.amazon.com/about-aws/newsletters/2012/08/14/august-2012/
[k] http://aws.amazon.com/ebs/
[l] https://developers.google.com/appengine/kb/billing
[m] Greg D'Alesandre, Google App Engine; http://tinyurl.com/D-Alesandre
[n] https://www.dotcloud.com/pricing.html
[o] https://cloud.google.com/pricing/
[p] http://tinyurl.com/cloud-price-war
[q] http://openstack.org
[r] James Hamilton, Amazon Web Services, slide: "Amazon Cycle of Innovation"; http://tinyurl.com/james-hamilton
[s] http://spotcloud.com
[t] http://aws.amazon.com/ec2/reserved-instances/marketplace/
[u] http://www.cloudsigma.com/cloud-computing/what-is-the-cloud/171
[v] http://www.cloudsigma.com/about-us/press-releases/242
[w] http://tinyurl.com/6fusion-CME
[x] http://docs.dotcloud.com/0.9/faq/
[y] http://aws.amazon.com/ec2/reserved-instances/marketplace/

Extrapolating from the progression of SLA terms we see, clients in the RaaS cloud will be able to define their own priority level, choosing from a relatively priced continuum. Moreover, if prices are market-driven, and priority levels reflect clients' willingness to pay, then we expect clients to be able to change their desired priority levels as often as prices change.

It is possible to extend the prevalent SLA language—"unavailability of a minimal period *X*, which is at least a fraction *Y* of a service period *Z*"—to express different absolute levels by controlling the parameters *X*, *Y*, and *Z*.[5] However, we extrapolate that as more cloud providers adopt flexible SLAs, they will continue the existing trend of relative priorities and not venture into extending absolute SLA language to several tiers.

Economic Dynamics

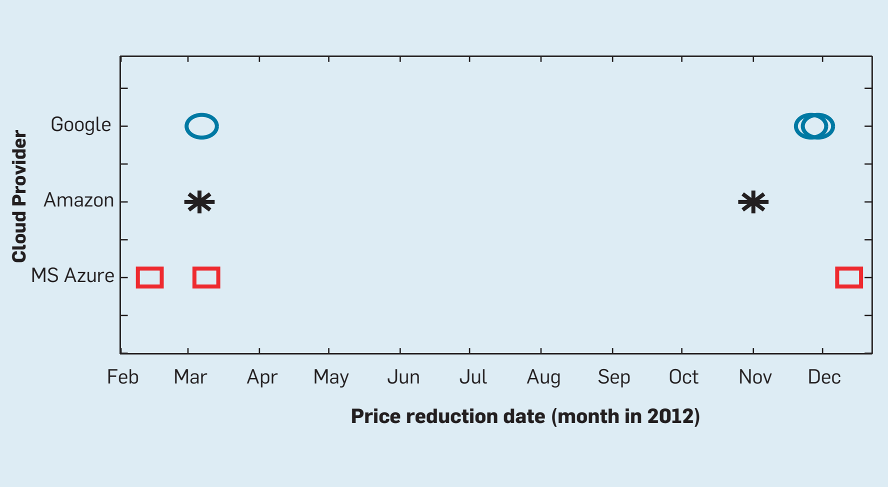
We have considered several ongoing trends, trying to anticipate where they will take the market next. We now survey the economic forces operating on clients and providers, along with their implications. These forces caused the phenomena discussed earlier and will continue pushing today's IaaS clouds along until, inevitably, they undergo a paradigm shift that is likely to turn them into RaaS clouds.

Forces acting on clients. As clients purchase more cloud services, their cloud bill increases. When bills are large, clients seek systematic savings. The best way to do so is by paying only for the resources they need, only when they need them. When clients are able to adjust the resources they rent to match the resources they use, their effective utilization rises, and the cost per utilized resource drops, potentially by 50%–85%, depending on resource utilization.[7] The more flexible the provider offerings, the greater control clients have over their costs and resulting performance. As providers offer increasingly fine-grain resources and service levels, clients are incentivized to develop or adopt resource-provisioning methods. As time scales shorten, manual provisioning methods become tedious, increasing clients' incentive to rely on computerized provisioning agents[38] to act on their behalf.

Forces acting on providers. Competition among IaaS cloud providers is increasing, as indicated by recent cloud price reductions. During the early years—2006–2011—Amazon reduced its prices as it announced new instance types, but by only 15%, while Amazon's hardware costs dropped by 80%.[35] However, the timing of price cuts in 2012 by three major cloud providers was correlated (see the figure here), a phenomenon called a "cloud price war."[p]

Competition is driven in part by commoditization of cloud-computing platforms. Commoditization eases application porting between providers; an example is the open source OpenStack,[q] the foundation of both Rackspace's and Hewlett-Packard's public clouds. OpenStack also offers Amazon EC2/S3-compatible APIs. As changing providers becomes easier, and as hungry new providers enter the market, competition increases and providers are forced to lower their prices.

Correlated cloud-price reduction dates for three major cloud providers, 2012.



Implications of increased competition. As competition increases and prices decrease, providers attempt to cut their costs[r] in an effort to maintain their profit margins. At any moment, given the available revenue-creating client workload, providers seek to minimize their costs (especially power costs) by idling or halting some machines or components[12] by consolidating instances to as few physical machines as reasonably possible. When resources are overcommitted due to consolidation and clients suddenly wish to use more resources than are physically available on the machine, the result is resource pressure.

The move toward tiered service and fine rental granularity is driven in part by the need to reduce costs and accompanying resource pressure. When clients change their resource consumption on the fly, providers continuing to guarantee absolute QoS levels must reserve a conservative amount of headroom for each resource on each physical server. Such spare resources are required just in case all clients simultaneously require all the resources promised them. Clients changing their resource consumption on the fly do not pay for this headroom unless and until they need it, so making it available all the time is wasteful.

Under the fixed-bundles model, if the host (hypervisor) chooses to overcommit resources, some clients would get less than the bundle they paid for. If the headroom is too small and there is resource pressure, this underprovisioning will be felt by the client in the form of reduced performance, and the illusion of a fixed bundle will dissipate.

Extending the current absolute SLA language to several tiers reduces only some of the headroom. To eliminate headroom completely, providers must resort to prioritization via tiered service levels, guaranteeing clients only relative QoS. But because relative QoS requires that clients change their approach, it should be introduced gradually, allowing them to control the risk to which they agreed to be exposed.

Here is a concrete example of how a traditional provider might waste its resources and a future provider increase utilization of its powered-up servers and reduce its power costs. Consider a 4GB physical machine running an instance that once required 3GB of memory but now uses only 2GB. A new client would like to rent an instance with 2GB. Under the IaaS model, the new client cannot be accommodated on this machine; 1GB goes unsold, and 2GB go unused. With tiered SLAs and dynamic resources, the first client can temporarily reduce its holdings to 2GB, and the provider can rent 2GB to the new client. If conflicts arise later due to a shortage of memory, the provider can choose how much memory to rent to each client on the basis of economic considerations. No memory goes unused, and no extra physical server has to be booted.

The RaaS Cloud

We have outlined the distinct trends operating in IaaS clouds, along with the economic forces governing them. Their combined effect is leading to a qualitative transformation of the IaaS cloud into what we call the RaaS cloud. The following is our view of the RaaS cloud, along with possible steps on the path to its realization:

Trading in fine-grain resources. A RaaS cloud machine defines the rules and mechanisms of an economic resource-trading environment, in which the economic entities operate:

Seed virtual machine. In RaaS clouds, the client, upon admittance, purchases a seed virtual machine with a minimal initial amount of dedicated resources. All other resources needed for efficient intended operation of the virtual machine are continuously rented. This combination of resource rental schemes—pre-purchasing and multiple on-demand levels—benefits clients with flexibility of choice.

Fine-grain resources. Resources available for rent include CPU, RAM, and I/O, as well as special resources (such as computational accelerators like GPGPUs, FPGAs, and flash devices). CPU capacity is sold on a hardware-thread basis or even as number of cycles per unit of time; RAM is sold on the basis of memory frames; I/O is sold on the basis of subsets of I/O devices with associated I/O bandwidth and latency guarantees. Such devices include network interfaces and block interfaces. Accelerators are sold as I/O devices and as CPUs. A subset of an I/O device may be presented to the virtual machine as a direct-assigned single-root input/output virtualization virtual function (SF-IOV VF)[14] or as an emulated[4] or para-virtual device. A dynamic price tag is attached to every resource. Resource rental contracts are set for a minimal fixed period that need not coincide with the repricing period. The host may offer the guest machines renewal of their rental contracts at the same price for an additional fixed period.

In the RaaS cloud, providers leverage the variable willingness of clients to pay a certain price for resources at a given moment (as can be expressed by bids) to decide which client gets which resource.

Host economic coordinator. To facilitate continuous trading, the provider's cloud software includes an economic coordinator representing the provider's interests. It operates an economic mechanism that defines the resource-allocation and billing mechanism—which client gets which resources at what price. Several auctions have been proposed to such ends by, for example, Agmon Ben-Yehuda et al. for the RaaS cloud,[2] as well as Chun and Culler,[8] Kelly,[21] Lazar and Semret,[22] Lubin et al.,[23] and Waldspurger et al.[37] In addition, the coordinator may act as a clearing house and support a secondary market of computing resources inside the physical machine, as SpotCloud[s] does for fixed-bundle virtual machines and Kash et al.[20] proposed for the wireless spectrum.

Guest economic agent. To take part in auctions or trade, clients' virtual machines must include an economic agent representing the client's business interests. It rents the necessary resources—given current requirements, load, and costs—at the best possible prices, from either the provider or its neighbors—virtual machines co-located on the same physical machine, possibly belonging to different clients. When demand outstrips supply, the agent changes its bidding strategy (in cases where the provider runs an auction) or negotiates with neighbor machines' agents, mediating between the client's requirements and the resources available in the system, ultimately deciding how much to offer to pay for each resource at a given time.

Subletting. Clients can secure resources early and sublet them later if they no longer need them. Resource securing can be done either by actively renting resources long term or by negotiating a future contract with the host. Either way, resource subletting lays the ground for resource futures markets among clients. Clients can sublet to other clients on the same physical machine using infrastructure provided by the host's coordinator; the clients agree to redivide resources among them and inform the coordinator, which transfers the local resources from one guest to another, as Hu et al.[19] did for bandwidth resources. In addition to trading with a limited number of neighbors, clients can sublet excess resources to anyone in the form of nested full virtual machines,[6] a concept now gaining support. Examples of secondary compute-resource trade exist in the Amazon EC2 Reserved Instance Marketplace,[t] in CloudSigma's reseller option,[u] in Deutsche Börse's vendor-neutral cloud marketplace,[v] in CME Group's plans for an IaaS commodity Exchange,[w] and in Docker, which resells EC2's resources with added value.[x] The subletting option reduces the risk for clients who commit in advance to rent resources. It also partially relieves the provider from having to manage retail sales while improving utilization and possibly increasing revenue through seller fees.[y] Allowing clients to sublet can also be viewed as a loss leader (a feature that attracts clients by reducing their financial risk).

Legacy clients. IaaS providers can introduce RaaS capabilities gradually, without forcing clients to change their business logic. Legacy clients without an economic agent can still function in the RaaS cloud as they do in an IaaS cloud. They simply rent large RaaS seed machines serving as IaaS instances. IaaS virtual machines function in a RaaS cloud as well as they do in an IaaS cloud. However, to realize the RaaS benefits of vertical elasticity and reduced costs, clients must adapt.

Prioritized service levels. The economic mechanisms in the RaaS cloud determine various aspects of the relative service levels:

Priorities for headroom only. In the RaaS cloud, each client gets an absolute guarantee (for receiving resources and for price paid) only for its minimal consumption, which is constant in time although individually set. Additional resources are provided on a priority basis at market prices. A risk-averse client can pre-pay for a larger amount of constant resources, trading low cost for peace of mind. From the provider's point of view, the aggregate constant consumption provides a steady income source. Only resources that might go unused—headroom—are allocated on the basis of market competition.

In the RaaS cloud, virtual machines never know the precise amount of resources that will be available to them at any given moment.

Vertical elasticity. RaaS clients are offered on-the-fly, fine-grain, fine-timed vertical elasticity for each instance—the ability to expand and shrink the resource consumption of each virtual machine. The resources required for vertical elasticity are limited by the physical resources in a single machine, because migrating running virtual machines between physical machines is likely to remain less efficient than dynamically balancing available resources between virtual machines coexisting on the same physical machines. Hence, to enable a client to vertically upscale a virtual machine during peak-demand times, the additional resources must be taken from a neighbor.

In the RaaS cloud, providers leverage the variable willingness of clients to pay a certain price for resources at a given moment (as can be expressed through bids) to decide which client gets which resource. Market forces thus dictate the constantly changing prices of resources as well as their allocation. In effect, the RaaS cloud provider does the opposite of Robin Hood by taking from the poor and giving to the rich.

A few good neighbors. The RaaS virtual machine's vertical elasticity is determined through a market mechanism by its neighbors' willingness to pay. The neighbors also determine the cost of the elastic expansion. Due to the inherent inefficiencies of live virtual machine migration, RaaS clouds must include an algorithm for placing client virtual machines on physical machines. This algorithm should achieve the right mixture of clients with different SLAs on each physical machine in the cloud, such that high-priority clients always have low-priority clients besides themselves to provide them with greater capacity when their demand peaks. Low-paying clients can use the high-paying clients' leftover resources when they do not need them, keeping the provider's machines constantly utilized. Another objective of the allocation algorithm is to allow low-priority clients enough aggregate resources for their needs. A low-priority client is thus expected to tolerate a temporary loss of service every so often, but if the physical resources are strictly less than the mean demand, such a client would never get enough resources to make meaningful progress. To retain low-priority clients, the placement algorithm must thus provide them enough resources to make (some) progress.

Full house. The RaaS provider also influences the QoS the RaaS client experiences by limiting the "maximal possible aggregate demand" for physical resources on the machine. Demand can be limited by controlling the number of virtual machines per physical machine and the maximal vertical elasticity to which each virtual machine is entitled. When the "maximal possible aggregate demand" is less than the supply, resources are wasted, but all virtual machines can freely expand. When the "maximal possible aggregate demand" exceeds supply, clients are less likely to succeed in vertical expansion when they need it or might be forced to pay more for the same expansion. RaaS clients are thus willing to pay more to be hosted in a physical machine with lower "maximal possible aggregate demand." This trade-off encourages RaaS providers to expose information about the aggregate demand and supply on the physical machine to its clients.

Implications, Challenges, Opportunities

The RaaS cloud gives rise to a number of implications, challenges, and opportunities for providers and clients alike that did not exist in markets involving entire virtual machines.[3,28,32,33,39] Broadly speaking, the new research areas can be divided into two categories: technical mechanisms and policies.

The RaaS cloud requires new mechanisms for allocating, metering, charging for, reclaiming, and redistributing CPU, memory, and I/O resources among multiple untrusted, not-necessarily cooperative clients every few seconds.[2] These mechanisms must be efficient and reliable. In particular, they must be resistant to side-channel attacks from malicious clients.[31] Hardware mechanisms are a must for fine-grain resource metering in the RaaS cloud.

The RaaS cloud requires new system software and new applications. Operating systems and applications are generally written under the assumption their underlying resources are fixed and always available. In the RaaS cloud, virtual machines never know the precise amount of resources that will be available to them at any given moment. The software running on those virtual machines must therefore adapt to changing resource availability and exploit whatever resources the software has, when it has them. Assume a client application just got an extra 2Gbps of networking bandwidth at a steal of a price but only for one second. To use it effectively, as it is available, all the software layers, including the operating system, runtime layer, and application, must be aware of it.

The RaaS cloud requires efficient methods of balancing resources within a single physical machine while accounting for the various guaranteed service levels. Bottleneck resource allocation[11,13,16] is a step toward allocation of resource bundles but still requires an algorithm for setting the system share to which each client is entitled.

Resource balancers are most efficient when guest machines with different service levels are co-located on the same physical server. Workload balancers, which balance resources across entire cloud data centers, will need to consider the flexibility and SLA of virtual machines in addition to the current considerations—static resource requests only.

Under dynamic conditions, the intra-machine RaaS mechanisms will quickly respond to flexibility needs, holding the fort until the slower live migration can take place. However, live migration must take place to mitigate resource pressure on the most stressed machines, allowing clients to vertically expand. Large IaaS providers apparently manage without live migration,[31] as the high rate of initialization and shutdown of virtual machines makes the initial balancer effective enough. However, the fine time granularity of the changes in the RaaS cloud means live migration will be required more often. The RaaS cloud will thus require efficient methods for live migration of virtual machines and for network virtualization.

On the policy side, the RaaS cloud requires new economic models for deciding what to allocate, when to allocate it, and at what prices.[9] Ideally, these models should optimize the provider's revenue or a social welfare function, a function of the benefit of all clients. The clients may measure their benefit in terms of starvation, latency, or throughput, but the mechanisms should optimize the effect of these metrics on the welfare of the clients by, say, maximizing the sum of client benefits or minimizing the unhappiness of the most unsatisfied client.

These new economic models should also recognize that resources may complete or substitute for one another in different ways for different clients. For one client, resources could be economic complements. If, for each thread the application requires 1GB RAM and one core, a client renting 2GB and two cores will be interested in adding bundles of 1GB and one core. For another client, resources might be economic substitutes; every additional GB allows the application to cache enough previous results to require one less core. So when cores are expensive, a client renting 2GB and two cores will be able to release one core and rent another GB instead.

These allocation and pricing mechanisms should be incentive compatible; truth telling regarding private information should be a good course of action for clients so the provider can easily optimize resource allocations. The mechanisms should also be collusion-resistant: a virtual machine should not suffer if several of the virtual machines it is co-located with happen to belong to the same client. Like approximation algorithms for multi-unit auctions,[10,36] they should be computationally efficient at large scale, so addressing the resource-allocation problem does not become prohibitive.

The mechanisms should preserve client privacy, as well as minimize the waste incurred by using a distributed mechanism. Moreover, in order to work in the real world, economic mechanisms must accommodate realistic client willingness to pay, which is a function of clients' performance measurements. The mechanism must support such measured functions, which are not necessarily mathematically nice and regular; in particular, they may contain steps.[27] Another real-world demand is simplicity. If researchers combined some of the ideas mentioned here to create a cumbersome mechanism with satisfactory theoretical qualities, that would still not guarantee its acceptance by the market of providers and their clients.

Conclusion

Making the RaaS cloud a reality requires solving problems spanning everything from game theory and economic models to system software and architecture. The onus is on the cloud-computing research community to lead the way, building the mechanisms and policies that will make the RaaS cloud a reality.

Acknowledgment

We thank Sharon Kessler of Pardes-Hanna, Israel for insightful discussions. This research was partially supported by the Prof. A. PaziIAEC-UPBC Joint Research Foundation.

References

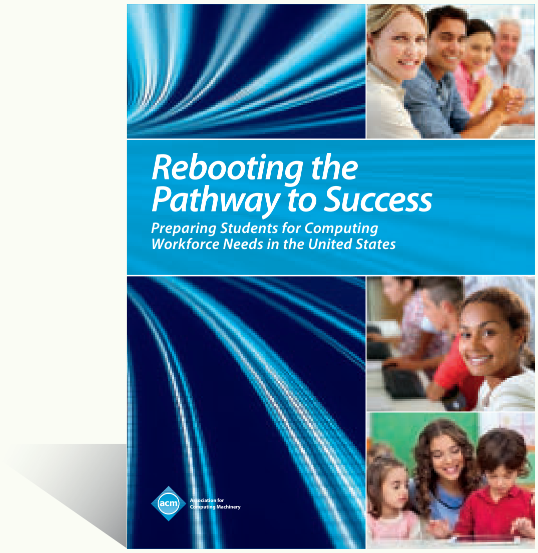
1. Agmon Ben-Yehuda, A., Ben-Yehuda, M., Schuster, A., and Tsafrir, D. Deconstructing Amazon EC2 spot instance pricing. *ACM Transactions on Economics and Computation 1*, 3 (Sept. 2013), 1–16.
2. Agmon Ben-Yehuda, A., Posener, E., Ben-Yehuda, M., Schuster, A., and Mu'alem, A. Ginseng: Market-driven memory allocation. In *Proceedings of the 10th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments* (Salt Lake City, UT, Mar.). ACM Press, New York, 2014, 41–52.
3. Altmann, J., Courcoubetis, C., Stamoulis, G., Dramitinos, M., Rayna, T., Risch, M., and Bannink, C. GridEcon: A marketplace for computing resources. In *Proceedings of Grid Economics and Business Models, Volume 5206 of Lecture Notes in Computer Science* (Las Palmas de Gran Canaria, Spain, Aug.). Springer, Berlin/Heidelberg, 2008, 185–196.
4. Amit, N., Ben-Yehuda, M., Tsafrir, D., and Schuster, A. vIOMMU: Efficient IOMMU emulation. In *Proceedings of the USENIX Annual Technical Conference* (Portland, OR, June). USNIX Association, Berkeley, CA, 2011.
5. Baset, S.A. *Cloud SLAs: Present and future. ACM SIGOPS Operating Systems Review 46*, 2 (July 2012), 57–66.
6. Ben-Yehuda, M., Day, M.D., Dubitzky, Z., Factor, M., Har'El, N., Gordon, A., Liguori, A., Wasserman, O., and Yassour, B.-A. The Turtles Project: Design and implementation of nested virtualization. In *Proceedings of the Symposium on Operating Systems Design and Implementation* (Vancouver, BC). USNIX Association, Berkeley, CA, 2010, 423–436.
7. Chen, Y. and Sion, R. To cloud or not to cloud?: Musings on costs and viability. In *Proceedings of the Second ACM Symposium on Cloud Computing.* ACM Press, New York, 2011.
8. Chun, B.N. and Culler, D.E. *Market-based Proportional Resource Sharing for Clusters. Technical Report.* University of California, Berkeley, 2000; http://www.cs.berkeley.edu/~culler/papers/market.pdf
9. Danak, A. and Mannor, S. Resource allocation with supply adjustment in distributed computing systems. In *Proceedings of the International Conference on Distributed Computing Systems* (Genova, June). IEEE Computer Society, 2010, 498–506.
10. Dobzinski, S. and Nisan, N. Mechanisms for multi-unit auctions. *Journal of Artificial Intelligence Research 37* (2010), 85–98.
11. Dolev, D., Feitelson, D.G., Halpern, J.Y., Kupferman, R., and Linial, N. No justified complaints: On fair sharing of multiple resources. In *Proceedings of the Innovations in Theoretical Computer Science Conference* (Boston). ACM Press, New York, 2012, 68–75.
12. Gandhi, A., Harchol-Balter, M., and Kozuch, M.A. Are sleep states effective in data centers? In *Proceedings of the International Green Computing Conference* (San Jose, CA, June). IEEE Computer Society, 2012, 1–10.
13. Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., and Stoica, I. Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation* (Boston, Mar.). USENIX Association, Berkeley, CA, 2011.
14. Gordon, A., Amit, N., Har'El, N., Ben-Yehuda, M., Landau, A., Tsafrir, D., and Schuster, A. ELI: Bare-metal performance for I/O virtualization. In *Proceedings of the ACM Conference on Architectural Support for Programming Languages and Operating Systems* (London, U.K., Mar.). ACM Press, New York, 2012, 411–422.
15. Gupta, D., Lee, S., Vrable, M., Savage, S., Snoeren, A.C., Varghese, G., Voelker, G.M., and Vahdat, A. Difference engine: Harnessing memory redundancy in virtual machines. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation* (San Diego, Dec.). USENIX Association, Berkeley, CA, 2008, 309–322.
16. Gutman, A. and Nisan, N. Fair allocation without trade. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems* (Valencia, June). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2012, 719–728.
17. Heo, J., Zhu, X., Padala, P., and Wang, Z. Memory overbooking and dynamic control of Xen virtual machines in consolidated environments. In *Proceedings of the Symposium on Integrated Network Management.* IEEE Computer Society, 2009, 630–637.
18. Holdren, J.P. and Lander, E. *Realizing the Full Potential of Government-held Spectrum to Spur Economic Growth. Technical Report.* The President's Council of Advisors on Science and Technology, Washington, D.C., July 2012; http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast_spectrum_report_final_july_20_2012.pdf
19. Hu, L., Ryu, K.D., Silva, D.D., and Schwan, K. v-Bundle: Flexible group resource offerings in clouds. In *Proceedings of the International Conference on Distributed Computing Systems* (Macau, June). IEEE Computer Society, 2012, 406–415.
20. Kash, I.A.; Murty, R.; Parkes, D.C., Enabling spectrum sharing in secondary market auctions. *IEEE Transactions on Mobile Computing 13*, 3 (Mar. 2014), 556–568.
21. Kelly, F. Charging and rate control for elastic traffic. *European Transactions on Telecommunications 8*, 1 (Jan.-Feb. 1997), 33–37.
22. Lazar, A. and Semret, N. *Design, Analysis and Simulation of the Progressive Second Price Auction for Network Bandwidth Sharing.* Columbia University, New York, Apr. 1998.
23. Lubin, B., Parkes, D.C., Kephart, J., and Das, R. Expressive power-based resource allocation for data centers. In *Proceedings of the International Joint Conference on Artificial Intelligence* (Pasadena, CA, July 2009), 1451–1456.
24. Nathuji, R., Kansal, A., and Ghaffarkhah, A. Q-Clouds: Managing performance interference effects for QoS-aware clouds. In *Proceedings of the ACM SIGOPS European Conference on Computer Systems* (Paris, Apr.). ACM Press, New York, 2010, 237–250.
25. Odlyzko, A. Paris Metro pricing for the Internet. In *Proceedings of the First ACM Conference on Electronic Commerce* (New York, Nov.). ACM Press, New York, 1999, 140–147.
26. Padala, P., Hou, K.-Y., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., and Merchant, A. Automated control of multiple virtualized resources. In *Proceedings of the ACM SIGOPS European Conference on Computer Systems* (Nuremberg, Germany, Apr.). ACM Press, New York, 2009, 13–26.
27. Parkes, D.C., Procaccia, A.D., and Shah, N. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. In *Proceedings of the ACM Conference on Electronic Commerce* (Valencia, Spain, June). ACM Press, New York, 2012, 808–825
28. Rahman, M.R., Lu, Y., and Gupta, I. *Risk-Aware Resource Allocation for Clouds. Technical Report.* University of Illinois at Urbana-Champaign, 2011; http://hdl.handle.net/2142/25754
29. Ramchurn, S.D., Vytelingum, P., Rogers, A., and Jennings, N.R. Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence. *Commun. ACM 55*, 4 (Apr. 2012), 86–97.
30. Ried, S., Kisker, H., Matzke, P., Bartels, A., and Lisserman, M. *Sizing the Cloud—Understanding and Quantifying the Future of Cloud Computing. Technical Report.* Forrester Research, Cambridge, MA, 2011; http://www.forrester.com/Sizing+The+Cloud/fulltext/-/E-RES58161
31. Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In *Proceedings of the ACM Conference on Computer and Communications Security* (Chicago, Nov.). ACM Press, New York, 2009, 199–212.
32. Shneidman, J., Ng, C., Parkes, D.C., Auyoung, A., Snoeren, A.C., Vahdat, A., and Chun, B. Why markets could (but don't currently) solve resource allocation problems in systems. In *Proceedings of the USENIX Workshop on Hot Topics in Operating Systems* (Santa Fe, NM, June). USENIX Association, Berkeley, CA, 2005.
33. Vanmechelen, K., Depoorter, W., and Broeckhove, J. Combining futures and spot markets: A hybrid market approach to economic grid resource management. *Journal of Grid Computing 9*, 1 (Mar. 2011), 81–94.
34. Verma, A., Ahuja, P., and Neogi, A. Power-aware dynamic placement of HPC applications. In *Proceedings of the ACM International Conference on Supercomputing* (Island of Kos, Greece, June). ACM Press, New York, 2008, 175–184.
35. Vermeersch, K. *A Broker for Cost-Efficient QoS-aware Resource Allocation in EC2. Master's Thesis.* Universiteit Antwerpen, Antwerp, Belgium; http://www.thesis.kurtvermeersch.com/
36. Vöcking, B. A universally truthful approximation scheme for multi-unit auctions. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (Kyoto, Japan, Jan.). Siam, 2012, 846–855.
37. Waldspurger, C.A., Hogg, T., Huberman, B.A., Kephart, J.O., and Stornetta, W.S. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering 18*, 2 (Feb 1992), 103–117.
38. Yi, S., Kondo, D., and Andrzejak, A. Reducing costs of spot instances via checkpointing in the Amazon Elastic Compute Cloud. In *Proceedings of the IEEE International Conference on Cloud Computing* (Miami, FL, July). IEEE, 2010, 236–243.
39. Zaman, S. and Grosu, D. Combinatorial auction-based dynamic VM provisioning and allocation in clouds. In *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science* (Athens, Greece, Nov.-Dec.). IEEE, 2011, 107–114.
40. Zhou, X., Gandhi, S., Suri, S., and Zheng, H. eBay in the sky: Strategy-proof wireless spectrum auctions. In *Proceedings of the ACM International Conference on Mobile Computing and Networking* (Miami, FL). ACM Press, New York, 2–13.

Orna Agmon Ben-Yehuda (ladypine@cs.technion.ac.il) is a research associate in the Computer Science Department of the Technion - Israel Institute of Technology, Haifa, Israel.

Muli Ben-Yehuda (mulix@mulix.org) is the founder of Hypervisor Technologies and Consulting Ltd., Haifa, Israel, and chief scientist of Stratocale, Herzliya, Israel.

Assaf Schuster (assaf@cs.technion.ac.il) is head of the Technion Center for Computer Engineering and a professor in the Computer Science Department of the Technion - Israel Institute of Technology, Haifa, Israel.

Dan Tsafrir (dan@cs.technion.ac.il) is an assistant professor in the Computer Science Department of the Technion - Israel Institute of Technology, Haifa, Israel.



DOI:10.1145/2611567

Exploring the inherent technical challenges in realizing the potential of Big Data.

BY H.V. JAGADISH, JOHANNES GEHRKE, ALEXANDROS LABRINIDIS, YANNIS PAPAKONSTANTINOU, JIGNESH M. PATEL, RAGHU RAMAKRISHNAN, AND CYRUS SHAHABI

Big Data and Its Technical Challenges

IN A BROAD range of application areas, data is being collected at an unprecedented scale. Decisions that previously were based on guesswork, or on painstakingly handcrafted models of reality, can now be made using data-driven mathematical models. Such Big Data analysis now drives nearly every aspect of society, including mobile services, retail, manufacturing, financial services, life sciences, and physical sciences.

As an example, consider scientific research, which has been revolutionized by Big Data.[1,12] The Sloan Digital Sky Survey[23] has transformed astronomy from a field where taking pictures of the sky was a large part of an astronomer's job to one where the pictures are already in a database, and the astronomer's task is to find interesting objects and phenomena using the database. In the biological sciences, there is now a well-established tradition of depositing scientific data into a public repository, and also of creating public databases for use by other scientists. Furthermore, as technology advances, particularly with the advent of Next Generation Sequencing (NGS), the size and number of experimental datasets available is increasing exponentially.[13]

The growth rate of the output of current NGS methods in terms of the raw sequence data produced by a *single* NGS machine is shown in Figure 1, along with the performance increase for the SPECint CPU benchmark. Clearly, the NGS sequence data growth far outstrips the performance gains offered by Moore's Law for single-threaded applications (here, SPECint). Note the sequence data size in Figure 1 is the output of analyzing the raw images that are actually produced by the NGS instruments. The size of these raw image datasets themselves is so large (many TBs per lab per day) that it is impractical today to even consider storing them. Rather, these images are analyzed on the fly to produce sequence data, which is then retained.

Big Data has the potential to revolutionize much more than just research. Google's work on Google File System and MapReduce, and subsequent open source work on systems like Hadoop, have led to arguably the most extensive development and adoption of Big Data technologies, led by companies focused on the Web, such as Facebook,

» key insights

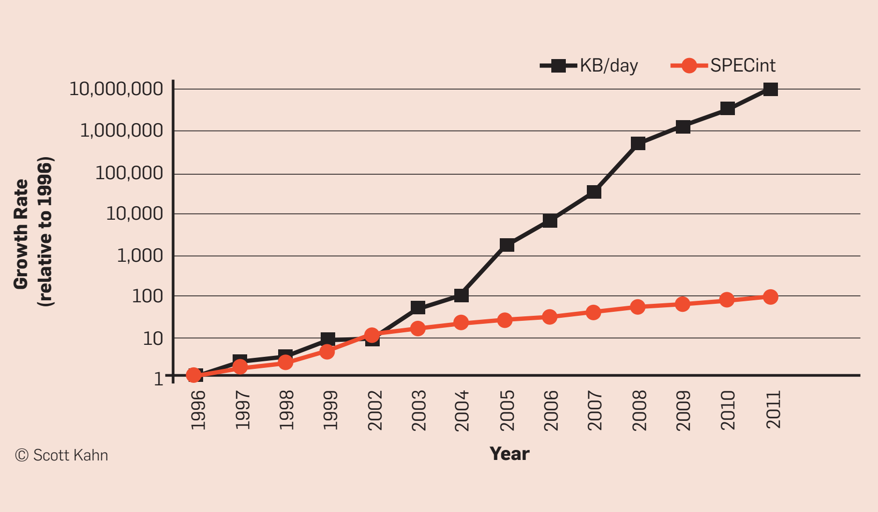
- **Big Data is revolutionizing all aspects of our lives ranging from enterprises to consumers, from science to government.**
- **Creating value from Big Data is a multi-step process: Acquisition, information extraction and cleaning, data integration, modeling and analysis, and interpretation and deployment. Many discussions of Big Data focus on only one or two steps, ignoring the rest.**
- **Research challenges abound, ranging from heterogeneity of data, inconsistency and incompleteness, timeliness, privacy, visualization, and collaboration, to the tools ecosystem around Big Data.**
- **Many case studies show there are huge rewards waiting for those who use Big Data correctly.**

IMAGE BY AMAZEME GRAPHICS

LinkedIn, Microsoft, Quantcast, Twitter, and Yahoo!. They have become the indispensable foundation for applications ranging from Web search to content recommendation and computational advertising. There have been persuasive cases made for the value of Big Data for healthcare (through home-based continuous monitoring and through integration across providers),[3] urban planning (through fusion of high-fidelity geographical data), intelligent transportation (through analysis and visualization of live and detailed road network data), environmental modeling (through sensor networks ubiquitously collecting data),[4] energy saving (through unveiling patterns of use), smart materials (through the new materials genome initiative[18]), machine translation between natural languages (through analysis of large corpora), education (particularly with online courses),[2] computational social sciences (a new methodology growing fast in popularity because of the dramatically lowered cost of obtaining data),[14] systemic risk analysis in finance (through integrated analysis of a web of contracts to find dependencies between financial entities),[8] homeland security (through analysis of social networks and financial transactions of possible terrorists), computer security (through analysis of logged events, known as Security Information and Event Management, or SIEM), and so on.

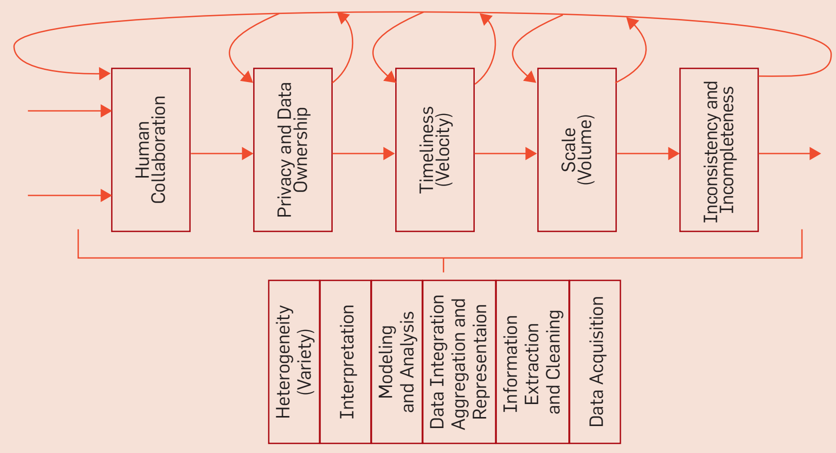
In 2010, enterprises and users stored more than 13 exabytes of new data; this is over 50,000 times the data in the Library of Congress. The potential value of global personal location data is estimated to be $700 billion to end users, and it can result in an up to 50% decrease in product development and assembly costs, according to a recent McKinsey report.[17] McKinsey predicts an equally great effect of Big Data in employment, where 140,000–190,000 workers with "deep analytical" experience will be needed in the U.S.; furthermore, 1.5 million managers will need to become data-literate. Not surprisingly, the U.S. President's Council of Advisors on Science and Technology recently issued a report on Networking and IT R&D[22] identified Big Data as a "research frontier" that can "accelerate progress across a broad range of priorities." Even popular news media now appreciates the value of Big Data as evidenced by coverage in the *Economist*,[7] the *New York Times*,[15,16] *National Public Radio*,[19,20] and *Forbes* magazine.[9]

Figure 1. Next-gen sequence data size compared to SPECint.



© Scott Kahn

Figure 2. The Big Data analysis pipeline. Major steps in the analysis of Big Data are shown in the top half of the figure. Note the possible feedback loops at all stages. The bottom half of the figure shows Big Data characteristics that make these steps challenging.


Human Collaboration
Privacy and Data Ownership
Timeliness (Velocity)
Scale (Volume)
Inconsistency and Incompleteness
Heterogeneity (Variety)
Interpretation
Modeling and Analysis
Data Integration Aggregation and Representaion
Information Extraction and Cleaning
Data Acquisition

While the potential benefits of Big Data are real and significant, and some initial successes have already been achieved (such as the Sloan Digital Sky Survey), there remain many technical challenges that must be addressed to fully realize this potential. The sheer size of the data, of course, is a major challenge, and is the one most easily recognized. However, there are others. Industry analysis companies like to point out there are challenges not just in *Volume*, but also in *Variety* and *Velocity*,[10] and that companies should not focus on just the first of these. Variety refers to heterogeneity of data types, representation, and semantic interpretation. Velocity denotes both the rate at which data arrive and the time frame in which they must be acted upon. While these three are important, this short list fails to include additional important requirements. Several additions have been proposed by various parties, such as *Veracity*. Other concerns, such as privacy and usability, still remain.

The analysis of Big Data is an iterative process, each with its own challenges, that involves many distinct phases as shown in Figure 2. Here, we consider the end-to-end Big Data life cycle.

Phases in the Big Data Life Cycle

Many people unfortunately focus just on the analysis/modeling step—while that step is crucial, it is of little use without the other phases of the data analysis pipeline. For example, we must approach the question of what data to record from the perspective that data is valuable, potentially in ways we cannot fully anticipate, and develop ways to derive value from data that is imperfectly and incompletely captured. Doing so raises the need to track provenance and to handle uncertainty and error. As another example, when the same information is represented in repetitive and overlapping fashion, it allows us to bring statistical techniques to bear on challenges such as data integration and entity/relationship extraction. This is likely to be a key to successfully leveraging data that is drawn from multiple sources (for example, related experiments reported by different labs, crowdsourced traffic information, data about a given domain such as entertainment, culled from different websites). These topics are crucial to success, and yet rarely mentioned in the same breath as Big Data. Even in the analysis phase, which has received much attention, there are poorly understood complexities in the context of multi-tenanted clusters where several users' programs run concurrently.

One place to do it all. The most important shift may well be that increasingly, the same data goes through all five stages of the life cycle, and it is no longer acceptable to have silos that address each stage. How do we provide an integrated set of data management and analysis capabilities that support all five stages adequately?

In the rest of this article, we begin by considering the five stages in the Big Data pipeline, along with challenges specific to each stage. We also present a case study (see sidebar) as an example of the issues that arise in the different stages. Here, we discuss the six crosscutting challenges.

Data acquisition. Big Data does not arise in a vacuum: it is a record of some underlying activity of interest. For example, consider our ability to sense and observe the world around us, from the heart rate of an elderly citizen, to the presence of toxins in the air we breathe, to logs of user-activity on a website or event-logs in a software system. Sensors, simulations and scientific experiments can produce large volumes of data today. For example, the planned square kilometer array telescope will produce up to one million terabytes of raw data per day.

Pushing summarization to edge devices. What we can filter and compress is often tied to the intended analysis in intimate ways, and a fixed filtering strategy does not work well. Can we provide flexible complex event processing frameworks that can optimize data acquisition by pushing down permissible filtering and compression criteria based on the user's analysis to edge devices where the data is generated?

Much of this data can be filtered and compressed by orders of magnitude without compromising our ability to reason about the underlying activity of interest. One challenge is to define these "on-line" filters in such a way they do not discard useful information, since the raw data is often too voluminous to even allow the option of storing it all. For example, the data collected by sensors most often are spatially and temporally correlated (such as traffic sensors on the same road segment). Suppose one sensor reading differs substantially from the rest. This is likely to be due to the sensor being faulty, but how can we be sure it is not of real significance?

Furthermore, loading of large datasets is often a challenge, especially when combined with on-line filtering and data reduction, and we need efficient incremental ingestion techniques. These might not be enough for many applications, and effective in-situ processing has to be designed.

Information extraction and cleaning. Frequently, the information collected will not be in a format ready for analysis. For example, consider the collection of electronic health records in a hospital, comprised of transcribed dictations from several physicians, structured data from sensors and measurements (possibly with some associated uncertainty), image data such as X-rays, and videos from probes. We cannot leave the data in this form and still effectively analyze it. Rather, we require an information extraction process that pulls out the required information from the underlying sources and expresses it in a structured form suitable for analysis. Doing this correctly and completely is a continuing technical challenge. Such extraction is often highly application-dependent (for example, what you want to pull out of an MRI is very different from what you would pull out of a picture of the stars, or a surveillance photo). Productivity concerns require the emergence of declarative methods to precisely specify information extraction tasks, and then optimizing the execution of these tasks when processing new data.

Most data sources are notoriously unreliable: sensors can be faulty, humans may provide biased opinions, remote websites might be stale, and so on. Understanding and modeling these sources of error is a first step toward developing data cleaning techniques. Unfortunately, much of this is data source and application dependent.

Data integration, aggregation, and representation. Effective large-scale analysis often requires the collection of heterogeneous data from multiple sources. For example, obtaining the 360-degrees health view of a patient (or a population) benefits from integrating and analyzing the medical health record along with Internet-available environmental data and then even with readings from multiple types of meters (for example, glucose meters, heart meters, accelerometers, among others[3]). A set of data transformation and integration tools helps the data analyst to resolve heterogeneities in data structure and semantics. This heterogeneity resolution leads to integrated data that is uniformly interpretable within a community, as they fit its standardization schemes and analysis needs. However, the cost of full integration is often formidable and the analysis needs shift quickly, so recent "pay-as-you-go" integration techniques provide an attractive "relaxation," doing much of this work on the fly in support of ad hoc exploration.

It is notable that the massive availability of data on the Internet, coupled with integration and analysis tools that allow for the production of derived data, lead to yet another kind of data proliferation, which is not only a problem of data volume, but also a problem of tracking the provenance of such derived data (as we will discuss later).

Even for simpler analyses that depend on only one dataset, there usually are many alternative ways of storing the same information, with each alternative incorporating certain trade-offs. Witness, for instance, the tremendous variety in the structure of bioinformatics databases with information about substantially similar entities, such as genes. Database design is today an art, and is carefully executed in the enterprise context by highly paid professionals. We must enable other professionals, such as domain scientists, to create effective data stores, either through devising tools to assist them in the design process or through forgoing the design process completely and developing techniques so datasets can be used effectively in the absence of intelligent database design.

Modeling and analysis. Methods for querying and mining Big Data are fundamentally different from traditional statistical analysis on small samples. Big Data is often noisy, dynamic, heterogeneous, inter-related, and untrustworthy. Nevertheless, even noisy Big Data could be more valuable than tiny samples because general statistics obtained from frequent patterns and correlation analysis usually overpower individual fluctuations and often disclose more reliable hidden patterns and knowledge. In fact, with suitable statistical care, one can use approximate analyses to get good results without being overwhelmed by the volume.

Interpretation. Ultimately, a decision-maker, provided with the result of analysis, has to interpret these results. Usually, this involves examining all the assumptions made and retracing the analysis. Furthermore, there are many possible sources of error: computer systems can have bugs, models almost always have assumptions, and results can be based on erroneous data. For all of these reasons, no responsible user will cede authority to the computer system. Rather, she will try to understand, and verify, the results produced by the computer. The computer system must make it easy for her to do so. This is particularly a challenge with Big Data due to its complexity. There are often crucial assumptions behind the data recorded. Analytical pipelines can involve multiple steps, again with assumptions built in. The recent mortgage-related shock to the financial system dramatically underscored the need for such decision-maker diligence—rather than accept the stated solvency of a financial institution at face value, a decision-maker has to examine critically the many assumptions at multiple stages of analysis. In short, it is rarely enough to provide just the results. Rather, one must provide users with the ability both to interpret analytical results obtained and to repeat the analysis with different assumptions, parameters, or datasets to better support the human thought process and social circumstances.

While the potential benefits of Big Data are real and significant, and some initial successes have already been achieved, there remain many technical challenges that must be addressed to fully realize this potential.

The net result of interpretation is often the formulation of opinions that annotate the base data, essentially closing the pipeline. It is common that such opinions may conflict with each other or may be poorly substantiated by the underlying data. In such cases, communities need to engage in a conflict resolution "editorial" process (the Wikipedia community provides one example of such a process). A novel generation of data workspaces is needed where community participants can annotate base data with interpretation metadata, resolve their disagreements and clean up the dataset, while partially clean and partially consistent data may still be available for inspection.

Challenges in Big Data Analysis

Having described the multiple phases in the Big Data analysis pipeline, we now turn to some common challenges that underlie many, and sometimes all, of these phases, due to the characteristics of Big Data. These are shown as six boxes in the lower part of Figure 2.

Heterogeneity. When humans consume information, a great deal of heterogeneity is comfortably tolerated. In fact, the nuance and richness of natural language can provide valuable depth. However, machine analysis algorithms expect homogeneous data, and are poor at understanding nuances. In consequence, data must be carefully structured as a first step in (or prior to) data analysis.

An associated challenge is to automatically generate the right metadata to describe the data recorded. For example, in scientific experiments, considerable detail regarding specific experimental conditions and procedures

Case Study

Since fall 2010, as part of a contract with Los Angeles Metropolitan Transportation Authority (LA-Metro), researchers at the University of Southern California's (USC) Integrated Media Systems Center (IMSC) have been given access to high-resolution spatiotemporal transportation data from the LA County road network. This data arrives at 46 megabytes per minute and over 15 terabytes have been collected so far. IMSC researchers have developed an end-to-end system called *TransDec* (for Transportation Decision-making) to acquire, store, analyze and visualize these datasets (see the accompanying figure). Here, we discuss various components of TransDec corresponding to the Big Data flow depicted in Figure 2.

Acquisition: The current system acquires the following datasets in real time:

- ▸ *Traffic loop-detectors:* About 8,900 sensors located on the highways and arterial streets collect traffic parameters such as occupancy, volume, and speed at the rate of one reading/sensor/min.
- ▸ *Bus and rail:* Includes information from about 2,036 busses and 35 trains operating in 145 different routes in Los Angeles County. The sensor data contain geospatial location of each bus every two minutes, next-stop information relative to current location, and delay information relative to predefined timetables.
- ▸ *Ramp meters and CMS:* 1851 ramp meters regulate the flow of traffic entering into highways according to current traffic conditions, and 160 Changeable Message Signs (CMS) to give travelers information about road conditions such as delays, accidents, and roadwork zones. The update rate of each ramp meter and CMS sensor is 75 seconds.
- ▸ *Event:* Detailed free-text format information (for example, number of casualties, ambulance arrival time) about special events such as collisions, traffic hazards, and so on acquired from three different agencies.

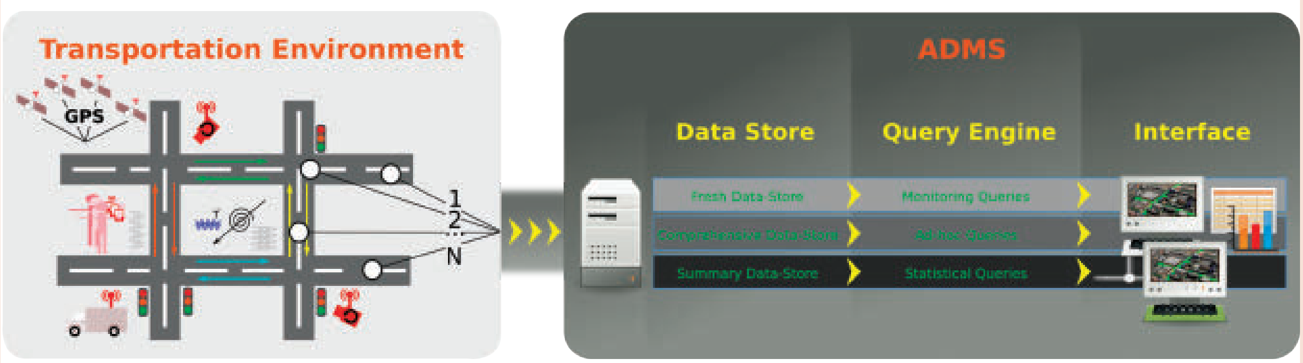
Cleaning: Data-cleaning algorithms remove redundant XML headers, detect and remove redundant sensor readings, and so on in real time using Microsoft's StreamInsight, resulting in reducing the 46MB/minute input data to 25MB/minute. The result is then dumped as simple tables into the Microsoft Azure cloud platform.

Aggregation/Representation: Data are aggregated and indexed into a set of tables in Oracle 11g (indexed in space and time with an R-tree and B-tree). For example, the data are aggregated to create sketches for supporting a predefined set of spatial and temporal queries (for example, average hourly speed of a segment of north-bond I-110).

Analysis: Several machine-learning techniques are applied, to generate accurate traffic patterns/models for various road segments of LA County at different times of the day (for example, rush hour), different days of the week (for example, weekends) and different seasons. Historical accident data is used to classify new accidents to predict clearance time and the length of induced traffic backlog.

Interpretation: Many things can go wrong in a complex system, giving rise to bogus results. For example, the failures of various (independent) system components can go unnoticed, resulting in loss of data. Similarly, the data format was sometimes changed by one organization without informing a downstream organization, resulting in erroneous parsing. To address such problems, several monitoring scripts have been developed, along with mechanisms to obtain user confirmation and correction.

TransDec.



© Luciano Nocera

may be required in order to interpret the results correctly. Metadata acquisition systems can minimize the human burden in recording metadata. Recording information about the data at its birth is not useful unless this information can be interpreted and carried along through the data analysis pipeline. This is called data provenance. For example, a processing error at one step can render subsequent analysis useless; with suitable provenance, we can easily identify all subsequent processing that depends on this step. Therefore, we need data systems to carry the provenance of data and its metadata through data analysis pipelines.

Inconsistency and incompleteness. Big Data increasingly includes information provided by increasingly diverse sources, of varying reliability. Uncertainty, errors, and missing values are endemic, and must be managed. On the bright side, the volume and redundancy of Big Data can often be exploited to compensate for missing data, to crosscheck conflicting cases, to validate trustworthy relationships, to disclose inherent clusters, and to uncover hidden relationships and models.

Similar issues emerge in crowdsourcing. While most such errors will be detected and corrected by others in the crowd, we need technologies to facilitate this. As humans, we can look at reviews of a product, some of which are gushing and others negative, and come up with a summary assessment based on which we can decide whether to buy the product. We need computers to be able to do the equivalent. The issues of uncertainty and error become even more pronounced in a specific type of crowdsourcing called participatory-sensing. In this case, every person with a mobile phone can act as a multi-modal sensor collecting various types of data instantaneously (or example, picture, video, audio, location, time, speed, direction, acceleration). The extra challenge here is the inherent uncertainty of the data collection devices. The fact that collected data is probably spatially and temporally correlated can be exploited to better assess their correctness. When crowdsourced data is obtained for hire, such as with Mechanical Turks, the varying motivations of workers give rise to yet another error model.

Even after error correction has been applied, some incompleteness and some errors in data are likely to remain. This incompleteness and these errors must be managed during data analysis. Doing this correctly is a challenge. Recent work on managing and querying probabilistic and conflicting data suggests one way to make progress.

Scale. Of course, the first thing anyone thinks of with Big Data is its size. Managing large and rapidly increasing volumes of data has been a challenging issue for many decades. In the past, this challenge was mitigated by processors getting faster, following Moore's Law. But there is a fundamental shift under way now: data volume is increasing faster than CPU speeds and other compute resources.

Due to power constraints, clock speeds have largely stalled and processors are being built with increasing numbers of cores. In short, one has to deal with parallelism within a single node. Unfortunately, parallel data processing techniques that were applied in the past for processing data across nodes do not directly apply for intranode parallelism, since the architecture looks very different. For example, there are many more hardware resources such as processor caches and processor memory channels that are shared across cores in a single node.

Another dramatic shift under way is the move toward cloud computing, which now aggregates multiple disparate workloads with varying performance goals into very large clusters. This level of sharing of resources on expensive and large clusters stresses grid and cluster computing techniques from the past, and requires new ways of determining how to run and execute data processing jobs so we can meet the goals of each workload cost-effectively, and to deal with system failures, which occur more frequently as we operate on larger and larger systems.

This leads to a need for global optimization across multiple users' programs, even those doing complex machine learning tasks. Reliance on user-driven program optimizations is likely to lead to poor cluster utilization, since users are unaware of other users' programs, through virtualization. System-driven holistic optimization requires programs to be sufficiently transparent, for example, as in relational database systems, where declarative query languages are designed with this in mind. In fact, if users are to compose and build complex analytical pipelines over Big Data, it is essential they have appropriate high-level primitives to specify their needs.

In addition to the technical reasons for further developing declarative approaches to Big Data analysis, there is a strong business imperative as well. Organizations typically will outsource Big Data processing, or many aspects of it. Declarative specifications are required to enable meaningful and enforceable service level agreements, since the point of outsourcing is to specify precisely what task will be performed without going into details of how to do it.

Timeliness. As data grow in volume, we need real-time techniques to summarize and filter what is to be stored, since in many instances it is not economically viable to store the raw data. This gives rise to the acquisition rate challenge described earlier, and a timeliness challenge we describe next. For example, if a fraudulent credit card transaction is suspected, it should ideally be flagged before the transaction is completed—potentially preventing the transaction from taking place at all. Obviously, a full analysis of a user's purchase history is not likely to be feasible in real time. Rather, we need to develop partial results in advance so that a small amount of incremental computation with new data can be used to arrive at a quick determination. The fundamental challenge is to provide interactive response times to complex queries at scale over high-volume event streams.

Another common pattern is to find elements in a very large dataset that meet a specified criterion. In the course of data analysis, this sort of search is likely to occur repeatedly. Scanning the entire dataset to find suitable elements is obviously impractical. Rather, index structures are created in advance to find qualifying elements quickly. For example, consider a traffic management system with information regarding thousands of vehicles and local hot spots on roadways. The system may need to predict potential congestion points along a route chosen by a user, and suggest alternatives. Doing so requires evaluating multiple spatial proximity queries working with the trajectories of moving objects. We need to devise new index structures to support a wide variety of such criteria.

Privacy and data ownership. The privacy of data is another huge concern, and one that increases in the context of Big Data. For electronic health records, there are strict laws governing what data can be revealed in different contexts. For other data, regulations, particularly in the U.S., are less forceful. However, there is great public fear regarding the inappropriate use of personal data, particularly through linking of data from multiple sources. Managing privacy effectively is both a technical and a sociological problem, which must be addressed jointly from both perspectives to realize the promise of Big Data.

Consider, for example, data gleaned from location-based services, which require a user to share his/her location with the service provider. There are obvious privacy concerns, which are not addressed by hiding the user's identity alone without hiding her location. An attacker or a (potentially malicious) location-based server can infer the identity of the query source from its (subsequent) location information. For example, a user may leave "a trail of packet crumbs" that can be associated with a certain residence or office location, and thereby used to determine the user's identity. Several other types of surprisingly private information such as health issues (for example, presence in a cancer treatment center) or religious preferences (for example, presence in a church) can also be revealed by just observing anonymous users' movement and usage patterns over time. In general, it has been shown there is a close correlation between people's identities and their movement patterns.[11] But with location-based services, the location of the user is needed for a successful data access or data collection, so doing this right is challenging.

Another issue is that many online services today require us to share private information (think of Facebook applications), but beyond record-level access control we do not understand what it means to share data, how the shared data can be linked, and how to give users fine-grained control over this sharing in an intuitive, but effective way. In addition, real data are not static but get larger and change over time; none of the prevailing techniques results in any useful content being released in this scenario.

Privacy is but one aspect of data ownership. In general, as the value of data is increasingly recognized, the value of the data owned by an organization becomes a central strategic consideration. Organizations are concerned with how to leverage this data, while retaining their unique data advantage, and questions such as how to share or sell data without losing control are becoming important. These questions are not unlike the Digital Rights Management (DRM) issues faced by the music industry as distribution shifted from sales of physical media such as CDs to digital purchases; we need effective and flexible *Data DRM* approaches.

The human perspective: Visualization and collaboration. For Big Data to fully reach its potential, we need to consider scale not just for the system but also from the perspective of *humans*. We have to make sure the end points—humans—can properly "absorb" the results of the analysis and not get lost in a sea of data. For example, ranking and recommendation algorithms can help identify the most interesting data for a user, taking into account his/her preferences. However, especially when these techniques are being used for scientific discovery and exploration, special care must be taken to not imprison end users in a "filter bubble"[21] of only data similar to what they have already seen in the past—many interesting discoveries come from detecting and explaining outliers.

If users are to compose and build complex analytical pipelines over Big Data, it is essential they have appropriate high-level primitives to specify their needs.

In spite of the tremendous advances made in computational analysis, there remain many patterns that humans can easily detect but computer algorithms have a difficult time finding. For example, CAPTCHAs exploit precisely this fact to tell human Web users apart from computer programs. Ideally, analytics for Big Data will not be all computational—rather it will be designed explicitly to have a human in the loop. The new subfield of visual analytics is attempting to do this, at least with respect to the modeling and analysis phase in the pipeline. There is similar value to human input at all stages of the analysis pipeline.

In today's complex world, it often takes multiple experts from different domains to really understand what is going on. A Big Data analysis system must support input from multiple human experts, and shared exploration of results. These multiple experts may be separated in space and time when it is too expensive to assemble an entire team together in one room. The data system must accept this distributed expert input, and support their collaboration. Technically, this requires us to consider sharing more than raw datasets; we must also consider how to enable sharing algorithms and artifacts such as experimental results (for example, obtained by applying an algorithm with specific parameter values to a given snapshot of an evolving dataset).

Systems with a rich palette of visualizations, which can be quickly and declaratively created, become important in conveying to the users the results of the queries in ways that are best understood in the particular domain and are at the right level of detail. Whereas early business intelligence systems' users were content with tabular presentations, today's analysts need to pack and present results in powerful visualizations that assist interpretation, and support user collaboration. Furthermore, with a few clicks the user should be able to drill down into each piece of data she sees and understands its provenance. This is particularly important since there is a growing number of people who have data and wish to analyze it.

Big Data Collaboratories. As many communities begin to rely on cloud-based data management and large shared data repositories become key resources, the potential value of collaboration using shared data goes up significantly. How do we permit users to create data analyses that combine their data with shared data and (selectively) allow other users to re-run, refine, and redistribute these analytic artifacts, which could range from single queries to entire modeling and scoring workflows? This requires us to address a number of issues (for example, provenance, access control, or workflows) but holds great potential for increased collaboration, and raising the level of transparency in collaborative work (imagine being able to re-run all the analysis reported in a paper using the same data and code used by the authors and being able to refine and publish the results!).

A popular new method of harnessing human ingenuity to solve problems is through crowdsourcing. Wikipedia, the online encyclopedia, is perhaps the best-known example of crowdsourced data. Social approaches to Big Data analysis hold great promise. As we make a broad range of data-centric artifacts sharable, we open the door to social mechanisms such as rating of artifacts, leader-boards (for example, transparent comparison of the effectiveness of several algorithms on the same datasets), and induced reputations of algorithms and experts.

Conclusion

We have entered an era of Big Data. Many sectors of our economy are now moving to a data-driven decision making model where the core business relies on analysis of large and diverse volumes of data that are continually being produced. This data-driven world has the potential to improve the efficiencies of enterprises and improve the quality of our lives. However, there are a number of challenges that must be addressed to allow us to exploit the full potential of Big Data. This article highlighted key technical challenges that must be addressed, and acknowledge there are other challenges, such as economic, social, and political, that are not covered in this article but must also be addressed. Not all of the technical challenges discussed here arise in all application scenarios. But many do. Also, the solutions to a challenge may not be the same in all situations. But again, there often are enough similarities to support cross-learning. As such, the broad range of challenges described here make good topics for research across many areas of computer science. We have collected some suggestions for further reading at http://db.cs.pitt.edu/bigdata/resources. These are a few dozen papers we have chosen on account of their coverage and importance, rather than a comprehensive bibliography, which would comprise thousands of papers.

Acknowledgment

This article is based on a white paper[5] authored by many prominent researchers, whose contributions we acknowledge. Thanks to Divyakant Agrawal, Philip Bernstein, Elisa Bertino, Susan Davidson, Umeshwar Dayal, Michael Franklin, Laura Haas, Alon Halevy, Sam Madden, Kenneth Ross, Dan Suciu, Shiv Vaithyanathan, and Jennifer Widom.

H.V.J. was funded in part by NSF grants IIS 1017296, IIS 1017149, and IIS 1250880. A.L. was funded in part by NSF IIS-0746696, NSFOIA-1028162, and NSF CBET-1250171. Y.P. was funded in part by NSF grants IIS-1117527, SHB-1237174, DC-0910820, and an Informatica research award. J.M.P. was funded in part by NSF grants III-0963993, IIS-1250886, IIS-1110948, CNS-1218432, and by gift donations from Google, Johnson Controls, Microsoft, Symantec, and Oracle. C.S. was funded in part by NSF grant IIS-1115153, a contract with LA Metro, and unrestricted cash gifts from Microsoft and Oracle.

Any opinions, findings, conclusions or recommendations expressed in this article are solely those of its authors.

References

1. Computing Community Consortium. Advancing Discovery in Science and Engineering. Spring 2011.
2. Computing Community Consortium. Advancing Personalized Education. Spring 2011.
3. Computing Community Consortium. Smart Health and Wellbeing. Spring 2011.
4. Computing Community Consortium. A Sustainable Future. Summer 2011.
5. Computer Research Association. Challenges and Opportunities with Big Data. Community white paper available at http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf
6. Dobbie, W. and Fryer, Jr. R.G. Getting Beneath the Veil of Effective Schools: Evidence from New York City. NBER Working Paper No. 17632. Issued Dec. 2011.
7. *Economist.* Drowning in numbers: Digital data will flood the planet—and help us understand it better. (Nov 18, 2011); http://www.economist.com/blogs/dailychart/2011/11/big-data-0
8. Flood, M., Jagadish, H.V., Kyle, A., Olken, F. and Raschid, L. Using data for systemic financial risk management. In *Proc. 5th Biennial Conf. Innovative Data Systems Research* (Jan. 2011).
9. *Forbes.* Data-driven: Improving business and society through data. (Feb. 10, 2012); http://www.forbes.com/special-report/data-driven.html
10. Gartner Group. Pattern-Based Strategy: Getting Value from Big Data. (July 2011 press release); http://www.gartner.com/it/page.jsp?id=1731916
11. González, M.C., Hidalgo, C.A. and Barabási, A-L. Understanding individual human mobility patterns. *Nature 453*, (June 5, 2008), 779–782.
12. Hey, T., Tansley, S. and Tolle, K., eds. *The Fourth Paradigm: Data-Intensive Scientific Discovery.* Microsoft Research, 2009.
13. Kahn, S.D. On the future of genomic data. *Science 331*, 6018 (Feb. 11, 2011), 728–729.
14. Lazar, D. et al. Computational social science. *Science 323*, 5915 (Feb. 6, 2009), 721–723.
15. Lohr, A. The age of Big Data. *New York Times* (Feb. 11, 2012); http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html
16. Lohr, S. How Big Data became so big. *New York Times* (Aug. 11, 2012); http://www.nytimes.com/2012/08/12/business/how-big-data-became-so-big-unboxed.html
17. Manyika, J. et al. Big Data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute. May 2011.
18. National Science and Technology Council. *Materials Genome Initiative for Global Competitiveness.* June 2011.
19. Noguchi, Y. Following the Breadcrumbs to Big Data Gold. National Public Radio (Nov. 29, 2011); http://www.npr.org/2011/11/29/142521910/the-digital-breadcrumbs-that-lead-to-big-data
20. Noguchi, Y. The Search for Analysts to Make Sense of Big Data. National Public Radio, (Nov. 30, 2011); http://www.npr.org/2011/11/30/142893065/the-search-for-analysts-to-make-sense-of-big-data
21. Pariser, E. *The Filter Bubble: What the Internet Is Hiding From You.* Penguin Press, May 2011.
22. PCAST Report. *Designing a Digital Future: Federally Funded Research and Development in Networking and Information Technology* (Dec. 2010); http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-nitrd-report-2010.pdf
23. SDSS-III: Massive Spectroscopic Surveys of the Distant Universe, the Milky Way Galaxy, and Extra-Solar Planetary Systems (Jan. 2008); http://www.sdss3.org/collaboration/description.pdf/

H.V. Jagadish (jag@umich.edu) is the Bernard A Galler Collegiate Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor.

Johannes Gehrke (johannes@cs.cornell.edu) is the Tisch University Professor of the Department of Computer Sciences a Cornell University, Ithaca, NY.

Alexandros Labrinidis (labrinid@cs.pitt.edu) is an associate professor in the Department of Computer Science at the University of Pittsburgh and co-director of the Advanced Data Management Technologies Laboratory.

Yannis Papakonstantinou (yannis@cs.ucsd.edu) is a Professor of Computer Science and Engineering at the University of California, San Diego.

Jignesh M. Patel (jignesh@cs.wisc.edu) is a professor of computer science at the University of Wisconsin, Madison.

Raghu Ramakrishnan (raghu@microsoft.com) is a Technical Fellow and CTO of Information Services at Microsoft, Redmond, WA.

Cyrus Shahabi (shahabi@usc.edu) is a professor of computer science and electrical engineering and the director of the Information Laboratory at the University of Southern California as well as director of the NSF's Integrated Media Systems Center.

research highlights

DOI:10.1145/2618411

To view the accompanying paper, visit doi.acm.org/10.1145/2618413

Technical Perspective
The Power of Joint Multiuser Beamforming

By Konstantina (Dina) Papagiannaki

WIRELESS COMMUNICATIONS HAVE completely revolutionized the way we connect with people and access information. With the advent of Wi-Fi and 3G/4G cellular technologies, we can access the Internet from nearly everywhere around the world without the need for cables, and more importantly, while on the move. Increasingly powerful portable devices, such as tablets and smartphones, are further increasing the traffic demand on wireless infrastructures.

Increasing the capacity of a wireless network can happen in two ways: either through the authorized use of additional or wider frequency bands, or through the densification of the underlying wireless infrastructure through the deployment of an increasing number of Wi-Fi Access Points (APs) or cellular Base Stations (BTSs). The former solution is typically a very lengthy, expensive, and heavily regulated process. The latter bears a very significant investment cost on the part of the operator, currently undertaken by a number of cellular providers in their deployment of femtocell and picocells. When it comes to Wi-Fi networks, enterprises tend to spend a very significant amount of money deploying APs at a density that may be as high as one AP every four meters (commonly found in enterprise Wi-Fi deployments).

However, the pure increase of the density of Wi-Fi APs or cellular BTSs is not a solution by itself. Additional complexity arises from the management of a larger number of devices in the network, and more importantly its configuration to reap the desired gains. Benefits in terms of capacity are only delivered if the operator is able to appropriately allocate frequencies/channels to the different devices in its network, so as to minimize areas of overlap between APs that operate on the same frequency. The fundamental problem addressed in frequency selection is that given the wireless medium is a shared medium, the more devices you have operating in the same frequency band, the lower the effective throughput for each individual device (something that leads to even lower performance if a device is located in the area of frequency-overlapping base stations). Power control, combined with intelligent frequency selection algorithms, aims to increase frequency reuse—the ability to reuse the same frequency often across space, however, without creating areas of frequency overlap.

The complexity of such a task is not for the fainthearted and has essentially led to the creation of an entire industry around Wi-Fi centralized architectures, pioneered by companies such as Aruba, Meru Networks, and Cisco, among others.

The following paper says that having multiple APs with an overlapping coverage area operating on the same frequency may not be a problem anymore. The authors describe a solution that can allow a wireless LAN to scale its throughput by continually adding more APs on the same channel! The target deployment scenario is that of a conference room or an auditorium, where APs are connected to each other through a high-speed wired network, and where dense AP deployments are absolutely necessary to accommodate traffic demand, while channels are too limited in number to prevent overlap.

The authors describe a solution that can allow a wireless LAN to scale its throughput by continually adding more APs on the same channel!

The authors borrow the fundamental working principle in today's Multiple Input Multiple Output (MIMO) transmitters—that of beamforming—and make it work across a number of independent transmitters. They call their scheme *Joint Multi-User Beamforming* (JMB). The challenge that must be addressed is the JMB transmitters need to control the relative phases of their transmitted signals to enable effective beamforming, by which the signals to unintended recipients cancel out. Given that independent transmitters have independent oscillators, such a requirement is not naturally met.

The authors address this challenge by designating one AP in the wireless LAN as the lead AP. The solution works in two phases. During the measurement phase, each AP measures its channel to each receiver, as well as the channel from the lead AP to the slave APs. During the data transmission phase, each slave AP corrects its frequency offset with respect to the lead AP, and all APs jointly transmit to concurrently deliver packets to multiple receivers. They show that such a mechanism can be easily accommodated within the context of 802.11n.

An actual implementation on a 10-node software-radio Wi-Fi testbed demonstrates a linear increase in network throughput with a median gain of 8.1 to 9.4x. Further experiments on unmodified 802.11n cards highlight the tremendous potential of the proposed solution.

The work discussed in this paper could completely change the philosophy underlying the design of dense enterprise wireless LAN deployments. C

Konstantina (Dina) Papagiannaki (dina@tid.es) is the scientific director of the Internet Systems and Networking scientific group at Telefonica I+D in Barcelona, Spain.

DOI:10.1145/2618413

JMB: Scaling Wireless Capacity with User Demands

By Hariharan Rahul, Swarun Kumar, and Dina Katabi

Abstract

We present JMB, a *joint* multiuser beamforming system, that enables independent access points (APs) to beamform their signals and communicate with their clients on the same channel as if they were one large MIMO transmitter. The key enabling technology behind JMB is a new low-overhead technique for synchronizing the phase of multiple transmitters in a distributed manner. The design allows a wireless LAN to scale its throughput by continually adding more APs on the same channel. JMB is implemented and tested with both software radio clients and off-the-shelf 802.11n cards, and evaluated in a dense congested deployment resembling a conference room. Results from a 10-AP software-radio testbed show a linear increase in network throughput with a median gain of 8.1–9.4×. Our results also demonstrate that JMB's joint multiuser beamforming can provide throughput gains with unmodified 802.11n cards.

1. INTRODUCTION

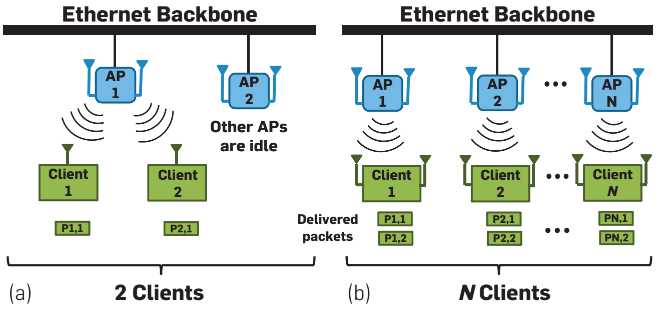
Wireless spectrum is limited; wireless demands can, however, grow unlimited. Busy Wi-Fi networks, for instance, in conference rooms, hotels, and enterprises are unable to keep up with user demands,[10, 24] even causing high-profile failures like the wireless network collapse during the Steve Jobs iPhone 4 keynote. Cellular networks are in a similar predicament, with their demands forecast to exceed available capacity within the next few years.[20] This is not for lack of improvement in the performance of wireless devices. Indeed, individual devices have improved dramatically in recent years through innovations like multi-antenna systems, better hardware, and lower receiver noise. The problem, however, is that there is a mismatch between the way user demands scale and network throughput scales; user demands scale with the number of devices in the network but network throughput does not. Unless network throughput also scales with the number of devices, wireless networks will always find it hard to keep up with their demands, and the projected demands will keep exceeding the projected capacity.

In this paper, we present a system that enables a network to scale its throughput with the number of transmitting devices. We focus on the scenario of typical busy wireless environments such as multiple users in a conference room, enterprise, hotel, etc. We enable a wireless LAN to keep increasing its total throughput by continuously adding more access points (APs) on the same channel.

The key idea behind our system is joint multiuser beamforming (JMB). Multiuser beamforming is a known technique that enables a MIMO transmitter to deliver multiple independent streams (i.e., packets) to receivers that have fewer antennas, as shown in Figure 1(a), where a 2-antenna access point delivers two packets concurrently to two single antenna receivers. In contrast, as shown in Figure 1(b), JMB enables multiple access points on the same channel to deliver their packets concurrently to multiple receivers, without interfering with each other. This system scales network throughput with the number of devices and delivers as many concurrent streams/packets as the total number of antennas on all APs. Furthermore, it leverages the continuing performance and reliability improvements of individual devices (e.g., more antennas per device).

The main challenge in implementing JMB stems from the need to synchronize the phases of distributed transmitters. Specifically, the goal of beamforming is to ensure that each client can decode its intended signal without interference. Thus, at each client, the signals intended for the other clients have to cancel each other out. This requires the transmitters to control the relative phases of their transmitted signals so that the desired cancellation can be achieved. Such a requirement is naturally satisfied in the case of a single device performing multiuser beamforming. However, in the case of JMB, the transmitters have independent oscillators, which are bound to have differences in their carrier frequencies. If one simply tries to jointly beamform these independent signals from

Figure 1. Traditional vs. joint multiuser beamforming. (a) In a traditional multiuser beamforming system with multiple 2-antenna APs, only one AP can transmit on a given channel at any given time. This leads to a maximum of two simultaneous packet transmissions regardless of the total number of APs. (b) In contrast, JMB enables all APs to transmit on the same channel, allowing up to 2*N* simultaneous packet transmissions if there are *N* 2-antenna APs.



A full version of this paper was published in *Proceedings of ACM SIGCOMM 2012*, Helsinki, Finland.

different transmitters, the drift between their oscillators will make the signals rotate at different speeds relative to each other, causing the phases to diverge and hence preventing beamforming.

At first blush, it might seem that it would be sufficient to estimate the frequency offset (i.e., the drift) $\Delta\omega$ between the transmitters, and compensate for the beamforming phase errors as $\Delta\phi = \Delta\omega t$, where t is the elapsed time. However, such an approach is not practical. It is well known[9] that frequency offset estimates have errors due to noise, and using such estimates to compute phases causes rapidly accumulating errors over time. Even a small error of, say, 10 Hz (4×10^{-3} ppm, which is several orders of magnitude smaller than the mandated 802.11 tolerance of 20 ppm, or cellular tolerance of 1–2 ppm) can lead to a large error of 20 degrees (0.35 radians) within a short time interval of 5.5 ms. Such a large error in the phase of the beamformed signals will cause significant interference at the receivers, preventing them from decoding.

JMB presents a simple, practical approach for synchronizing phases of multiple distributed transmitters. Its key idea is to elect one of the APs as a lead and use its phase as a reference for the whole system. Other APs (i.e., the slaves) directly measure the phase of the lead AP and change the phase of their signals to maintain a desired alignment with respect to the lead. In particular, JMB precedes every data packet with a couple of symbols transmitted by the lead AP. The slave APs use these symbols to directly measure the required phase correction for proper beamforming. Since this is a direct phase measurement as opposed to a prediction based on frequency offsets, it has no accumulated errors. After correcting for this phase error, the slave APs use the estimate for their frequency offset to predict any phase changes throughout the packet and correct for it. This bounds the maximum phase error accumulation to the duration of a packet. One can use a simple long-term average for the frequency offset to ensure that the phase error accumulated for the duration of a packet is within the desired performance bounds.

In the rest of the paper, we expand on this basic idea and demonstrate that it can deliver accurate joint beamforming across distributed transmitters. Further, we also extend this idea to work with off-the-shelf 802.11n cards. This would allow organizations to directly leverage JMB by simply upgrading their AP infrastructure, without requiring any modification to the clients.

We implemented JMB in two environments:

- The first environment consists of USRP2 APs and receivers, where both APs and clients can be modified. Here, we verify the scaling properties of JMB and also perform finer grained analysis of its components.
- The second environment consists of USRP2 APs and receivers with Intel Wi-Fi Link 5300 adapters. Each AP consists of two USRP2s connected via an external clock and configured to act as a 2-antenna MIMO AP. Correspondingly, each receiver Wi-Fi card has two antennas enabled. Here, we verify that JMB can provide throughput gains with off-the-shelf 802.11n cards, and further that it can provide these gains with multi-antenna devices.

We evaluated JMB in an indoor testbed using APs and receivers deployed densely in a room to simulate a conference room scenario. Our results reveal the following findings:

- **USRP testbed:** JMB's throughput increases linearly with the number of APs. In particular, in our testbed, which has 10 APs, JMB can achieve a median throughput gain of 8.1–9.4× over traditional 802.11 unicast, across the range of 802.11 signal to noise ratios (SNRs).
- **802.11 testbed:** JMB's ability to linearly scale the network throughput with the number of transmitters applies to off-the-shelf 802.11 clients. Specifically, JMB can transmit simultaneously from two 2-antenna APs to two 2-antenna 802.11n clients to deliver a median throughput gain of 1.8× compared to traditional 802.11n.

Contributions: This work presents the first system that scales wireless throughput by enabling joint beamforming from distributed independent transmitters. We achieve this by designing a simple, practical approach for phase synchronization across multiple distributed transmitters. We also show that our system can deliver throughput gains from joint beamforming with off-the-shelf 802.11n cards.

2. RELATED WORK

The full version of the paper[18] has a detailed survey of related work. In this version, we provide a brief overview.

Prior empirical systems that attempt to perform distributed multiuser beamforming[4, 15, 19] require tight synchronization using global positioning system (GPS) clocks or a shared oscillator, or joint decoding by exchanging received signals. Other systems that allow multiple nodes to transmit simultaneously, such as MU-MIMO in LTE,[12] SAM,[21] and multiuser beamforming,[1] provide only constant throughput gain and do not scale with the number of APs in the system. A third strand of work harnesses channel diversity gains using systems like distributed antennas and SourceSync[3, 17] or provides directional gains using phased arrays,[6] but cannot provide multiplexing gains and hence cannot scale throughput with the number of APs in the system. In contrast to all these systems, JMB empirically achieves tight phase synchronization using independent oscillators at the devices in the network, allows devices to work independently without sharing clock signals, and scales throughput linearly with the number of APs in the system. Further, it can work with off-the-shelf 802.11n cards.

Prior theoretical work[2, 22] on distributed phase synchronization assumes synchronous oscillators and only provides one-time phase offset calibration. Prior theory[16] also proves that distributed MIMO scales wireless capacity with the number of nodes. While JMB builds on this foundational work, JMB is the first empirical system that shows linear scaling of throughput with the number of transmitters in practical systems with unsynchronized oscillators and resulting time-varying phase differences.

3. JMB OVERVIEW

JMB is designed for the wireless downlink. It is applicable to wireless LANs, especially in dense deployments like enterprises, hotels, and conference rooms. JMB APs can operate with off-the-shelf Wi-Fi client hardware. Our techniques are applicable to cellular networks, but the details are beyond the scope of this paper.

JMB APs are connected by a high-throughput backend, say, Gigabit Ethernet, like APs are today. Packets intended for receivers are distributed to all APs over the shared backend. JMB enables the APs to transmit concurrently to multiple clients as if they were one large MIMO node, potentially delivering as many streams (i.e., packets) as the total number of antennas on all APs.

In the next few sections, we describe how JMB works. We start with the basic idea that enables distributed phase synchronization. We then describe our protocol implementing this basic idea for emulating a large MIMO node. We then extend our system to integrate our design with off-the-shelf Wi-Fi cards.

4. DISTRIBUTED PHASE SYNCHRONIZATION

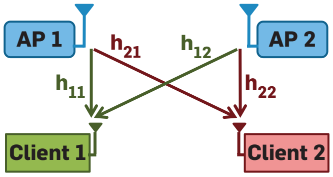
The chief goal of distributed phase synchronization is to enable different transmitters powered by different oscillators to emulate a single multi-antenna transmitter where all antennas are driven by the same oscillator. Intuitively our solution is simple: We declare one transmitter the lead, and make all other transmitters synchronize to the oscillator of the lead transmitter, that is, each transmitter measures the offset between its oscillator and the lead oscillator and compensates for the offset by appropriately correcting the phase of its transmitted signal. This behavior makes all transmitters act as if they were antennas on the same chip controlled by the same oscillator.

We now demonstrate how this intuitive design can deliver the proper MIMO behavior and hence enable each receiver to correctly decode its intended signal without interference. For simplicity, we consider a scenario of two single-antenna APs transmitting to two single-antenna clients, as shown in Figure 2. Let h_{ij}, where $i, j \in \{1, 2\}$, be the channel to client i from AP j, $x_j(t)$ the symbol that needs to be delivered to client j at time t, and $y_j(t)$ the symbol that is received by client j at time t. Correspondingly, let $\mathbf{H} = [h_{ij}]$, $i, j \in \{1, 2\}$, be the 2×2 channel matrix, $\overrightarrow{x(t)} = [x_1(t)\, x_2(t)]^T$ be the desired symbol vector, and $\overrightarrow{y(t)} = [y_1(t)\, y_2(t)]^T$ be the received symbol vector.

No oscillator offset: Assume first that there are no oscillator offsets between any of the APs and clients. If each AP i simply transmits the signal $x_i(t)$, each client will receive a linear combination of the transmitted signals. Since each client has only one antenna, client 1 receives $y_1(t) = h_{11}x_1(t) + h_{12}x_2(t)$ and client 2 receives $y_2(t) = h_{21}x_1(t) + h_{22}x_2(t)$. Each of these equations has two unknowns, and hence, neither client can decode its intended data.

In order to deliver two concurrent packets to the two clients, the APs need to ensure that each client receives only the signal intended for it (i.e., it experiences no interference from the signal intended for the other client). Specifically, we need the effective channel experienced by the transmitted signal to be diagonal, that is, it should satisfy:

Figure 2. Channel matrix with two APs transmitting to two clients.



$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} g_{11} & 0 \\ 0 & g_{22} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{1}$$

where g_{11} and g_{22} are any nonzero complex numbers. In this case, the received signal will simply appear at each receiver as if it has experienced the channel g_{ii}, which each receiver can estimate using standard techniques.

The APs can achieve this result by using *beamforming*. In beamforming, the APs measure all the channel coefficients from the transmitters to the receivers at time 0. Then, instead of transmitting $x_1(t)$ and $x_2(t)$ directly, the APs transmit:[a]

$$\begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} = \mathbf{H}^{-1} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{2}$$

In this case, the two clients receive:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \mathbf{H} \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} = \mathbf{H}\mathbf{H}^{-1} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

Since $\mathbf{H}\mathbf{H}^{-1} = \mathbf{I}$, the effective channel experienced by the clients in this case is a diagonal matrix, that is, Equation (1) is satisfied. Hence, each client can now decode its intended data without interference from the signal intended for the other client.

With oscillator offset: What happens when the oscillators of the APs and clients have different frequencies? Let ω_{Ti} be the oscillator frequency of AP i, and ω_{Rj} the oscillator frequency of client j, $i, j \in \{1, 2\}$. In this case, the channel at time t, $\mathbf{H}(\mathbf{t})$, can be written as:

$$\mathbf{H}(\mathbf{t}) = \begin{pmatrix} h_{11}e^{j(\omega_{T1}-\omega_{R1})t} & h_{12}e^{j(\omega_{T2}-\omega_{R1})t} \\ h_{21}e^{j(\omega_{T1}-\omega_{R2})t} & h_{22}e^{j(\omega_{T2}-\omega_{R2})t} \end{pmatrix}$$

where $j = sqrt(-1)$. Because the oscillators rotate with respect to each other, the channel no longer has a fixed phase.

Now, if the APs try to perform beamforming as before, using the channel value they computed at time $t = 0$ and transmitting $\mathbf{H}^{-1}\vec{x}$, the clients receive:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{H}(\mathbf{t})\mathbf{H}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

[a] The APs also need to normalize $\mathbf{H}^{-1}$ to respect power constraints, but we omit that detail for simplicity.

The product $\mathbf{H}(\mathbf{t})\mathbf{H}^{-1}$ is no longer diagonal, and hence the receivers cannot decode their intended signal. Thus, standard MIMO beamforming does not work in this case.

So, how can one do beamforming with such a time-varying channel? A naive approach would try to make each transmitter compute $\mathbf{H}(\mathbf{t})$ at every t and then multiply its time signal by $\mathbf{H}(\mathbf{t})^{-1}$. Say that the network has N APs and N clients. Then such an approach would require each transmitter to maintain accurate estimates of N^2 frequency offsets of the form $\Delta\omega_{ij} = \omega_{Tj} - \omega_{Ri}$. (Further since nodes can only measure offsets relative to other nodes, but not the absolute frequencies of their oscillators, the number of estimates cannot be reduced to N.) Measurement errors from all of these estimates will accumulate, prevent accuracy of beamforming, and create interference at the receivers. However, according to our initial intuition, we can make multiple transmitters act as if they were one MIMO node, and hence do accurate beamforming, by having each transmitter estimate only its frequency offset to the lead transmitter. Said differently, our intuition tells us that it should be possible to reduce the number of frequency offset estimates that each transmitter maintains from N^2 to 1. Let us see how we can achieve this goal.

Observe that we can decompose the channel matrix at time t as $\mathbf{H}(\mathbf{t}) = \mathbf{R}(\mathbf{t})\mathbf{H}\mathbf{T}(\mathbf{t})$, where $\mathbf{H}$ is time invariant and $\mathbf{R}(\mathbf{t})$ and $\mathbf{T}(\mathbf{t})$ are diagonal matrices defined as:

$$\mathbf{R}(\mathbf{t}) = \begin{pmatrix} e^{-j\omega_{R1}t} & 0 \\ 0 & e^{-j\omega_{R2}t} \end{pmatrix} \text{ and } \mathbf{T}(\mathbf{t}) = \begin{pmatrix} e^{j\omega_{T1}t} & 0 \\ 0 & e^{j\omega_{T2}t} \end{pmatrix}$$

Since $\mathbf{R}(\mathbf{t})$ is diagonal, it can function analogous to the $\mathbf{G}$ matrix in Equation (1). Thus, if the transmitters transmit the modified signal $\mathbf{T}(\mathbf{t})^{-1}\mathbf{H}^{-1}\vec{x}$ at time t, then the received signal can be written as:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{R}(\mathbf{t})\mathbf{H}\mathbf{T}(\mathbf{t})\mathbf{T}(\mathbf{t})^{-1}\mathbf{H}^{-1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{3}$$

which reduces to the desired form of Equation (1):

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{R}(\mathbf{t})\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{4}$$

Note that $\mathbf{T}(\mathbf{t})$ is also diagonal, and as a result the transmitter phase correction matrix

$$\mathbf{T}(\mathbf{t})^{-1} = \begin{pmatrix} e^{-j\omega_{T1}t} & 0 \\ 0 & e^{-j\omega_{T2}t} \end{pmatrix} \tag{5}$$

is also diagonal. Further, the phase correction entry for each AP depends only on the oscillator phase of that AP. This means that if each AP, i, knows its phase, $e^{j\omega_{Ti}t}$, at time t, it can simply compensate for that phase and the AP will not need any additional frequency or phase measurements. Unfortunately, this is not practical. An AP has no way to measure the exact phase change of its oscillator locally.

We address this difficulty by observing that the channel equation is unchanged when we multiply by $1 = e^{j\omega_{T1}t} e^{-j\omega_{T1}t}$, that is,

$$\mathbf{H}(\mathbf{t}) = e^{j\omega_{T1}t}\mathbf{R}(\mathbf{t})\mathbf{H}\mathbf{T}(\mathbf{t})e^{-j\,\omega_{T1}t}$$
$$= \begin{pmatrix} e^{j(\omega_{T1}-\omega_{R1})t} & 0 \\ 0 & e^{j(\omega_{T1}-\omega_{R2})t} \end{pmatrix} \mathbf{H} \begin{pmatrix} 1 & 0 \\ 0 & e^{(j(\omega_{T2}-\omega_{T1})t} \end{pmatrix}$$

Since the new observed channel matrix is still diagonal, the clients can still continue to decode the received signal as before.

The resulting system implements our initial intuition.

5. JMB PROTOCOL

We start by describing the protocol at a high level and follow by the detailed explanation. JMB's distributed transmission protocol works in two phases:

- JMB starts with a *channel measurement phase*, in which the APs measure two types of channels: (1) the channels from themselves to the receivers (i.e., the channel matrix $\mathbf{H}$), which is the beamforming channel matrix whose inverse the APs use to transmit data concurrently to their clients; and (2) the channels from the lead AP to each slave AP (the h_i^{lead}'s), which enable each slave AP to determine its relative oscillator offset from the lead AP.
- The channel measurement phase is followed by the *data transmission phase*. In this phase, the APs transmit jointly to deliver concurrent packets to multiple receivers. Data transmission uses beamforming after having each slave AP corrects for its frequency offset with respect to the lead AP.

Note that a single channel measurement phase can be followed by multiple data transmissions. Channels only need to be recomputed on the order of the coherence time, which is several hundreds of milliseconds in typical indoor scenarios.[5] Section 7 describes how JMB reduces channel measurement overhead in greater detail.

We now describe the channel measurement and data transmission phases in greater detail. (The description below assumes symbol level time synchronization, for which we use the scheme in Rahul et al.,[17] which provides tight synchronization up to a few nanoseconds. Our experimental results also incorporate an implementation of that scheme.)

5.1. Channel measurement

The goal of channel measurement is to obtain a snapshot of the channels from all APs to all clients, that is, $\mathbf{H}$ and the reference channels from the lead AP to the slave APs, that is, the h_i^{lead}, $\forall i$.

The key point is that *all these channels have to be measured at the same time*, which is the reference time $t = 0$. Otherwise the channels would rotate with respect to each other due to frequency offsets and hence be inconsistent. Below, we divide channel measurement into a few subprocedures.

(a) **Collecting measurements.** The lead AP starts the channel measurement phase with a synchronization header, followed by channel measurement symbols, that is, known orthogonal frequency division multiplexing (OFDM) symbols that the clients can use to estimate the channel. The channel measurement symbols are separated by a constant gap, whose value is chosen to permit the slave APs to send their channel measurement symbols interleaved with the symbols from the lead AP. When the slave APs hear the synchronization header, they know to transmit their channel measurement symbols in the gap, one after another, as shown in Figure 3.

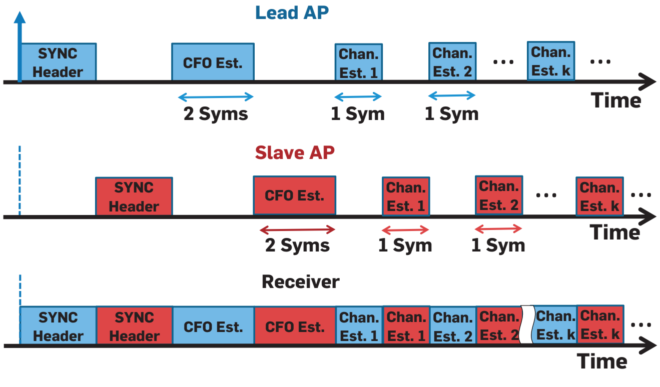
Thus, channel measurement symbols are repeated and interleaved. They are repeated to enable the clients to obtain accurate channel measurements by averaging multiple estimates to reduce the impact of noise. They are interleaved because we want the channels to be measured as if they were measured at the same time. Since exactly simultaneous transmissions will lead the APs to interfere with each other, JMB performs a close approximation to simultaneous transmission by interleaving symbols from different APs.

(b) **Estimating H at the clients.** Upon reception of the packet in Figure 3, each client performs three tasks: it computes its carrier frequency offset (CFO) to each AP; it then uses its knowledge of the transmitted symbols and the CFO to compute the channel from each AP to itself; and finally it uses its knowledge of the CFOs to rotate the phase of the channels so that they look as if they were measured exactly at the same time. We detail these tasks below.

Different transmitters (i.e., APs) have different oscillator offsets to receivers, and each receiver needs to measure the frequency offset from each transmitter to correct the corresponding symbols from that transmitter appropriately. To enable this, the channel measurement transmission uses CFO symbols from each AP followed by channel estimation symbols similar to traditional OFDM.[9] The only departure is that the receiver computes and uses different CFO and channel estimates for symbols corresponding to different APs.

Note that these channel estimates are still not completely simultaneous, in particular, the channel estimation symbols of slave AP i is separated from the symbol of the lead AP by $i - 1$ symbol widths, as shown in Figure 3. The receiver compensates for this by rotating the estimated channel for AP i by $e^{-j\Delta\omega_i(i-1)kT+D}$ (in each OFDM subcarrier), where T is the duration of one OFDM symbol, k is the index of the interleaved symbol, and D is the duration of the lead AP synchronization header. This ensures that all channels are measured at one reference time, which is the start of the synchronization header. The receiver averages the channel estimates (in each OFDM subcarrier) from each AP to cancel out the noise and obtain an accurate estimate. The receivers then communicate these estimated channels back to the transmitters over the wireless channel.

Figure 3. Packet structure from the perspective of APs and the receiver. Symbols in blue are transmitted by the lead AP, symbols in red by the slave AP, and symbols in white reflect silence periods.



(c) **Estimating the h_i^{lead}'s at the slave APs.** Each slave AP uses the synchronization header to compute the value of the channel from the lead AP to itself at the reference time $h_i^{\text{lead}}(0)$.

Note that at the end of the channel measurement phase, each slave AP i has the entire channel matrix to be used for beamforming, as well as a reference channel, $h_i^{\text{lead}}(0)$, from the lead AP which it will use during data transmissions, with all channels measured with respect to one reference time.

5.2. Data transmission

Now that the channels are measured, the APs can use beamforming to transmit data concurrently without interference.

(a) **AP coordination:** The APs need to agree on which packets are sent concurrently in one beamforming frame. To do this we leverage the bandwidth of the backend Gigabit Ethernet to send all client packets to all APs. The lead AP makes all control decisions and communicate them to the slave APs over the Ethernet. In particular, it determines which packets will be combined in a data transmission and communicates it to the slave APs over the wired backend.

(b) **Beamforming:** Client packets are transmitted by joint beamforming from the JMB APs participating in the system. Note that slave APs need to correct the phase of their signal prior to transmission. One way to do this would be for each slave to estimate the frequency offset $\omega_{\text{lead}} - \omega_{\text{slave}}$ from the lead to itself (using the synchronization header from the previous phase) and then compute the net elapsed phase by calculating $(\omega_{\text{lead}} - \omega_{\text{slave}})t$, where t is the time elapsed since the channel measurement was taken. However, this would lead to large accumulated errors over time because of inaccuracies in the initial frequency offset measurement. For example, even a small error of 100 Hz in the measurement of the initial frequency offset can lead to a large phase error of π radians in as short a timespan as 20 ms, and hence significantly affect the phase alignment required for correct beamforming. Unless addressed, this error would prevent JMB from amortizing the cost of a single channel measurement over the coherence time of the channel, for example, 250 ms, and would force the system to repeat the process of measuring **H** every few milliseconds, which means incurring the overhead of communicating the channels from all clients to the APs almost every packet.

JMB avoids this issue of accumulating error over large timescales by directly measuring the phase difference between the lead AP and the slave AP. Said differently instead of multiplying the frequency offset $\Delta\omega$ $(= \omega_{\text{lead}} - \omega_{\text{slave}})$ by the elapsed time (which leads to errors that accumulate over time), JMB directly measures the phase difference $\Delta\phi(t)$ $(= (\omega_{\text{lead}} - \omega_{\text{slave}})t)$.

In JMB the lead AP initiates data transmission using a synchronization header, as in channel estimation. Each slave AP uses this synchronization header to measure the current channel, $h_i^{\text{lead}}(t)$, from the lead AP to itself. Note that the current channel will be rotated relative to the reference channel because of the oscillator offset between the lead AP and slave AP. In particular, $h_i^{\text{lead}}(t) = h_i^{\text{lead}}(0)\ e^{j(\omega_{T1} - \omega_{T2})t}$. Each slave can therefore compute $e^{j(\omega_{T1} - \omega_{T2})t}$ directly, from its two measurements of the lead AP channel. Such an estimate does not have errors that accumulate over time because it is purely a division of two direct measurements. The slave then multiplies its transmitted signal by this quantity, as described in Section 4.

Now that all AP oscillators are synchronized at the beginning of the data transmission, the slave AP also needs to keep its oscillator synchronized with the lead transmitter through the actual data packet itself. It does this by multiplying its transmitted signal by $e^{j(\omega_{T1} - \omega_{T2})t}$, where t is the time since the initial phase synchronization at the beginning of the joint transmission. Note that this offset estimate only needs to be accurate within the packet, that is, for a few hundred microseconds or about 2 ms at most. JMB APs maintain a continuously averaged estimate of their offset with the lead transmitter across multiple transmissions to obtain a robust estimate that can maintain accurate phase synchronization within a packet.

Two additional points are worth noting. First, for ease of exposition, we have discussed the entire system so far in the context of correcting carrier frequency offsets. However, any practical wireless system has to also account for the sampling frequency offsets. Note that any offset in the sampling frequency just adds to the phase error in each OFDM subcarrier. Since our phase offset estimation using the synchronization header, described in Section 5, estimates the overall phase, it automatically accounts for the initial phase error accumulated from sampling frequency offset. Within each packet, the JMB slave APs correct for the effect of sampling frequency offset during the packet by using a long-term averaged estimate, similar to the carrier frequency offset.

Second, as mentioned earlier, in Section 5, JMB APs are synchronized in time using Rahul et al.[17] As described in Rahul et al.,[17] due to differences in propagation delays between different transmitters and different receivers, one cannot synchronize all transmitted signals to arrive exactly at the same time at all receivers. It is important to note that JMB works correctly even in the presence of different propagation delays between different transmitters and receivers. This is because the signals from different JMB APs will arrive within a cyclic prefix of each other at all receivers.[b] The delay differences between the signals from different APs at a receiver translate to a relative phase difference between the channels from these APs to that receiver. JMB's channel measurement phase captures these relative phase differences in the channel matrix, and JMB's beamforming then applies the effect of these phase differences while computing the inverse of the channel matrix.

5.3. Overarching principles

In summary, the core challenge met by JMB's design is to accurately estimate and track the phase differences between each of the N clients and N APs. This challenge is particularly arduous for two reasons: (1) each receiver must simultaneously track the phase of N independent transmitters, and (2) errors in the estimates in the CFO result in phase offsets that accumulate over time, quickly leading to very large errors. Our general approach to tackling these challenges is to have all transmitters and receivers synchronize their phase to that of a single lead transmitter. Our implementation of this approach has been guided by following three overarching principles:

- **Between APs and within a packet we can use estimated frequency and sampling offsets to track phase:** We can measure the frequency and sampling offsets *between APs* accurately enough that the accumulated phase differences within a single packet (tens to a few hundreds of *microseconds*) are not significant enough to harm performance. Specifically, since APs are a part of the infrastructure, and CFOs do not change significantly over time, we can get very accurate estimates of the CFO between APs by averaging over samples taken across many packets.
- **Between APs and *across* packets we *cannot* use estimated frequency and sampling offsets to track phase:** The across packet time scales (tens to hundreds of *milliseconds*) are large enough that even with extremely accurate estimates of the frequency and sampling offsets, the accumulated phase differences from residual errors will lead to significant performance degradation. To handle this, JMB uses a single header symbol to directly estimate the total *phase offset* and resync the phases of all nodes at the beginning of each packet.
- **Between a client and an AP we *cannot* use estimated frequency and sampling offsets to track phase even through a packet:** Since clients are a transient part of the network, we cannot get accurate enough estimates of frequency and sampling offsets to use for phase tracking even within a single packet.

[b] In fact, since the common design scenario for JMB is confined locations like conference rooms and auditoriums, the propagation delay differences between different APs to a receiver are in the tens of nanoseconds, which is smaller than the 802.11 cyclic prefix of 400 or 800 ns, which is designed for worst-case multipaths.

Thus, each client uses standard OFDM techniques to track the phase of the lead AP symbol by symbol. Additionally, when performing channel estimation, the APs interleave their packets so that the correction of the channels to a common reference time has minimal error.

6. COMPATIBILITY WITH 802.11

In order for JMB to work with clients using off-the-shelf 802.11n cards, JMB needs to address two challenges:

1. **Sync header:** The sync header transmitted by the lead AP to allow the slave APs to compute their oscillator offset, and trigger their transmission, is not supported by 802.11.
2. **Channel measurement:** Recall that JMB requires a snapshot of the channel from all transmitters to all receivers measured at the same time. In Section 4, we described how to do this with a custom channel measurement packet format with interleaved symbols, which allows a receiver to measure channels from all transmitters. However, such a packet format is not supported by 802.11, and hence 802.11n cards cannot simultaneously measure channels from all APs at the same time.

JMB solves these issues by leveraging 802.11n channel state information (CSI) feedback for beamforming. We now describe JMB's solutions to the above challenges.

6.1. Sync header

The lead AP in JMB needs to prefix each transmission with a sync header that allows the slave transmitters to measure their relative oscillator offset from the lead, and also triggers their joint transmission. A mixed mode 802.11n packet essentially consists of an 802.11n packet prefixed with five legacy symbols. These legacy symbols are only intended to trigger carrier sense in 802.11a/g nodes and are not used by 802.11n receivers. Thus, the lead JMB can use these legacy symbols as a sync header. JMB slave APs use the legacy symbols to measure their oscillator phase offset from the lead, correct their transmission signal, and join the lead AP's transmission after the legacy symbols when the actual 802.11n symbols are transmitted.

6.2. Channel measurement

802.11n does not support the interleaved packet format that allows JMB to measure a snapshot of the channels from all the transmitters to a receiver simultaneously. Further, an 802.11n receiver with K (at most 4) antennas can measure at most K channels at a time. In a JMB system, the total number of transmit antennas across all APs is larger than the number of antennas on any single receiver. Thus, a receiver with off-the-shelf 802.11n cards will be unable to simultaneously measure channels from all transmit antennas to itself.

Naively, one could measure the channels from all transmit antennas by transmitting a separate packet from each AP, and then correcting these measurements using the estimated frequency offsets to the receiver as described in Section 5.1.

Unlike the scenario in Section 5.1 where the transmissions from different APs are separated from each other by only a few symbols (using interleaving), the transmissions from different APs here are separated by at least one packet width. As discussed in Section 5.3, this separation would induce a large accumulated phase error due to inaccuracy in receiver frequency offset estimates.

JMB instead performs channel measurement by "tricking" the receiver into measuring channels from different AP antennas simultaneously. This trick allows JMB to measure the channel from each AP antenna to the receiver in conjunction with a common reference channel to the receiver. Using such a common reference across all measurements allows JMB to avoid measuring *receiver frequency offset*, and instead directly estimate and compensate *phase offset* between different measurements, as we describe in Section 7.

For simplicity, we focus on the scenario in Figure 4 with two APs and one client, where each node has two antennas. We will only describe the measurements to R_1 since channels to R_2 are naturally measured simultaneously with R_1 in exactly the same manner.

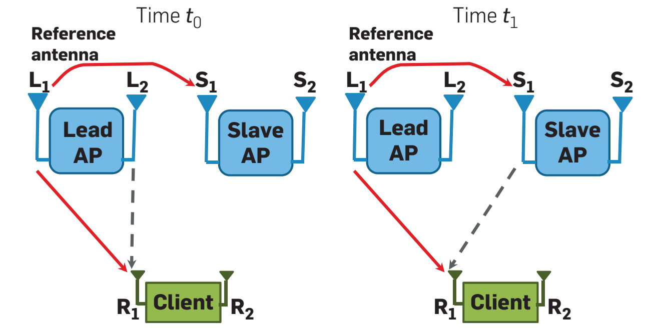
At time t_0, L_1 and L_2 transmit a two-stream packet jointly to R_1. This measurement gives us the channels $L_1 \rightarrow R_1$ and $L_2 \rightarrow R_1$ at time t_0. In addition, S_1 measures the channel $L_1 \rightarrow S_1$ using the synchronization header.

At time t_1, L_1 and S_1 trick the receiver by jointly transmitting a two-stream packet from two different APs. This measurement gives us the channels $L_1 \rightarrow R_1$ and $S_1 \rightarrow R_1$ at time t_1. Again, S_1 measures the channel $L_1 \rightarrow S_1$ using the synchronization header.

The challenge is that we would like to obtain the channel $S_1 \rightarrow R_1$ at time t_0 but we have only the channel $S_1 \rightarrow R_1$ measured at t_1.

We therefore need to correct our measured channel by the accumulated phase offset between S_1 and R_1 in the time interval t_0 to t_1. To do this, we take advantage of the fact that

Figure 4. 802.11n channel measurement. JMB measures channels to 802.11n clients by sending a series of two-stream transmissions. Every transmission includes the reference antenna, L_1, as well as one other antenna (either L_2 or S_1 in our example). For clarity, the figure does not show the transmissions to/from R_2 and S_2, but JMB naturally measures the channels to R_2 simultaneously.



we can compute the accumulated phase offset between both L_1 and R_1, and between L_1 and S_1 in the time interval t_0 to t_1.

- L_1 and R_1: We can compute this accumulated phase offset using the measurements of the channel $L_1 \rightarrow R_1$ at time t_0 and time t_1.
- L_1 and S_1: We can compute this accumulated phase offset using the measurements of the channel $L_1 \rightarrow S_1$ at time t_0 and time t_1.

The difference between these two accumulated phase offsets gives us the desired accumulated phase offset between S_1 and R_1 in the time interval t_0 to t_1.

We can similarly measure the channel $S_2 \rightarrow R_1$ in the next time slot, say t_2, and rotate it back to time t_0. We can repeat this process for all AP antennas.

7. DECOUPLING MEASUREMENTS TO DIFFERENT RECEIVERS

The scheme in Section 4 assumed that all channels from all APs to all receivers are measured simultaneously. In Section 6.2, we showed that we can relax this assumption for a *single* receiver. That is, we can measure channels from different APs to that receiver at different times by using a shared reference measurement across all APs for that receiver. But what about channels to *another* receiver? If this receiver joins the network after the channels to the first receiver are measured, there is no opportunity for a shared reference measurement between the two receivers. It might therefore seem that JMB's requirement for all channels to be measured at the same time would necessitate measurement of channels to all receivers whenever a receiver joins the network, or when a single receiver's channels change.

In fact, we can show that such full measurement is not necessary, and that JMB can decouple channel measurements to different receivers. The key idea is that JMB can use the channels from the lead AP to slave APs as a shared reference, instead of the channel from the lead AP to a receiver as was the case in Section 6.2. We prove in Rahul et al.[18] that using such a shared reference allows JMB to measure channels to different receivers at different times, and still correctly perform multiuser beamforming.

8. DIVERSITY

So far, we have described the use of JMB for multiplexing. The same principles apply to diversity except that in this case, we have all the APs transmitting jointly to a single client, say client 1. Each AP then computes its beamformed signal as $\frac{h_{1i}^*}{\|h_{1i}\|}x_1$ and slaves continue to perform distributed phase synchronization as before.

9. JMB'S LINK LAYER

In this paper, we have described JMB's physical layer that enables multiple APs to transmit simultaneously to multiple receivers. We refer the reader to the full version[18] for a discussion of how JMB's link layer (MAC, carrier sense, acknowledgments, retransmissions, etc.) is designed to use this capability.

10. TESTBED AND IMPLEMENTATION

We implement JMB's AP design in USRPs and evaluate it with both USRP and off-the-shelf 802.11n clients.

(a) **Implementation for the software radio testbed:** Each node is equipped with a USRP2 board and an RFX2400 daughterboard, and communicates on a 10 MHz channel in the 2.4 GHz range. We implement OFDM in GNURadio using various 802.11 modulations (BPSK, 4QAM, 16QAM, and 64QAM), coding rates, and choose between them using the effective SNR bitrate selection algorithm.[8]

To perform correct phase alignment, concurrent transmitters must be synchronized at the sample level. We do this by using USRP2 timestamps to synchronize transmitters despite delays introduced by software. Before every data packet, the lead AP sends a trigger signal on the medium at t_{trigger}. All other APs log the timestamp of this signal, add a fixed delay t_{Δ} to it, and then transmit concurrently at this new time. We select t_{Δ} as 150 μs based on the maximum delay of our software implementation. Finally, to optimize the software turnaround, we did not use GNURadio, but wrote our own C code that directly interacts with the USRP hardware.

(b) **Implementation for the 802.11n testbed:** There are two main differences between this testbed and the one above. First, each client in this testbed uses an off-the-shelf 802.11n card. Second, each node has two antennas and can act as a MIMO node. Our objective is to show that JMB extends beyond single antenna systems; for example, it can combine two 2 × 2 MIMO systems to create a 4 × 4 MIMO system.

Each AP consists of two USRP2 nodes connected to an external clock, acting as a 2-antenna node. Each client is a PC equipped with an Intel Wi-Fi Link 5300 a/b/g/n wireless network adapter on which two antennas are enabled. The Intel Wi-Fi Link 5300 adapters are updated with a custom firmware and associated `iwlwifi` driver in order to obtain the channel state information in user space.[7]

The AP software implementation is similar to the other testbed except that we make the channel width 20 MHz to communicate with 802.11n cards. The packet format is also changed to match 802.11n. The client software collects the channel measurements from the firmware and logs correctly decoded packets.

(c) **Testbed topology:** We evaluate JMB in an indoor testbed (shown in Rahul et al.[18]) that simulates a conference room, with APs deployed on ledges near the ceiling and clients scattered through the room. In every run, the APs and clients are assigned randomly to these locations. The testbed exhibits diverse SNRs as well as both line-of-sight and non-line-of-sight paths due to obstacles such as pillars, furniture, ledges, etc. The APs transmit 1500 byte packets to the clients.

11. RESULTS

We evaluate JMB both through microbenchmarks of its individual components and an integrated system on both USRP and 802.11n testbed. We refer the reader to Rahul et al.[18] for the microbenchmarks and focus on the system performance in this section.

11.1. Increase of network throughput with the number of APs

JMB's key goal is to increase network throughput with the number of APs. This experiment verifies if JMB delivers on that promise.

Method. We evaluate JMB's performance in three effective SNR ranges: low (6–12 dB), medium (12–18 dB), and high (>18 dB). For each range, we place a certain number of JMB nodes in random AP locations in the testbed. We then place the same number of nodes in random client locations such that all clients obtain an effective SNR in the desired range. For each such topology, we measure the throughput obtained both with 802.11n and JMB. Since USRP2 cannot perform carrier sense due to software latency, we measure 802.11n throughput by scheduling each client so that it gets an equal share of the medium. We repeat the experiment for 20 different topologies and also vary the number of JMB APs for each SNR range.

Results. Figures 5(a), (b), and (c) show the total throughput obtained by 802.11n and by JMB for different numbers of APs and different SNR ranges. Note that, as one would expect, the obtained throughput increases with SNR (802.11n throughput at low SNR is 7.75 Mbps, at medium SNR is around 14.9 Mbps, and at high SNR is 23.6 Mbps). There are two main points worth noting:

- 802.11n cannot benefit from additional APs operating in the same channel, and allows only one AP to be active at any given time. As a result, its throughput stays constant even as the number of APs increases. This throughput might vary with the number of APs in a real 802.11n network due to increased contention; however, since USRPs do not have carrier sense, we compute 802.11n throughput by providing each client with an equal share of the medium. In contrast, with JMB, as we add more APs, JMB can use these APs to transmit concurrent packets to more receivers. As a result, the throughput of JMB increases linearly with the number of APs.
- The absolute gains provided by JMB are higher at high (~9.4× for 10 APs) and medium (~9.1×) SNRs than at low SNRs (~8.1×). This is a consequence of the theoretically predicted throughput of beamforming. In particular, the beamforming throughput with N APs scales as $N \log\left(\frac{\text{SNR}}{k}\right) = N \log(\text{SNR}) - N \log(K)$, where K depends on the channel matrix $\mathbf{H}$ and is related to how well conditioned it is.[23] Natural channel matrices can be considered random and well conditioned, and hence K can essentially be treated as constant for our purposes. The 802.11n throughput scales roughly as log(SNR).[23] The expected gain of JMB over 802.11n can therefore be written as $N\left(1 - \frac{\log(k)}{\log(\text{SNR})}\right)$ and hence becomes closer to N as SNR increases. This is why JMB's gains at lower SNR grow at a lower rate than the gains at high SNR.

11.2. Compatibility with 802.11

Finally, as described in Section 6, JMB is compatible with existing 802.11n cards. In this section, we investigate whether JMB can deliver throughput gains when used with commodity 802.11n cards. Further, since each AP and each 802.11n card in this system has two antennas, this experiment also verifies that JMB can provide its expected gains with multi-antenna transmitters and receivers.

Method. We place two JMB nodes at random AP locations in the testbed and two 802.11n receivers at random client locations in the testbed. For each topology, we compute the throughput with 802.11n and with JMB. As before, we compute 802.11n throughput by giving each transmitter an equal share of the medium. We repeat the experiment across multiple topologies and the entire range of SNRs.

Results. Figure 6 shows the total throughput with and without JMB at high, medium, and low SNRs. Since we have two receivers in this experiment, the theoretical gain over 802.11n is 2×. The chart shows that JMB delivers an average gain of 1.67–1.83× across all SNR ranges. Similar to the case with USRP receivers, the gains in the high SNR regime are larger than the gains in the low SNR regime.

We now investigate JMB's fairness, that is, whether JMB can deliver its throughput gains for every receiver in the network across all locations and SNRs. Figure 7 shows the

Figure 5. Scaling of throughput with the number of APs. In this experiment, the number of APs equals the number of receivers. At all SNRs, JMB's network throughput increases linearly with the number of APs while total 802.11 throughput remains constant. (a) High SNR (>18 dB); (b) medium SNR (12–18 dB); (c) low SNR (6–12 dB).

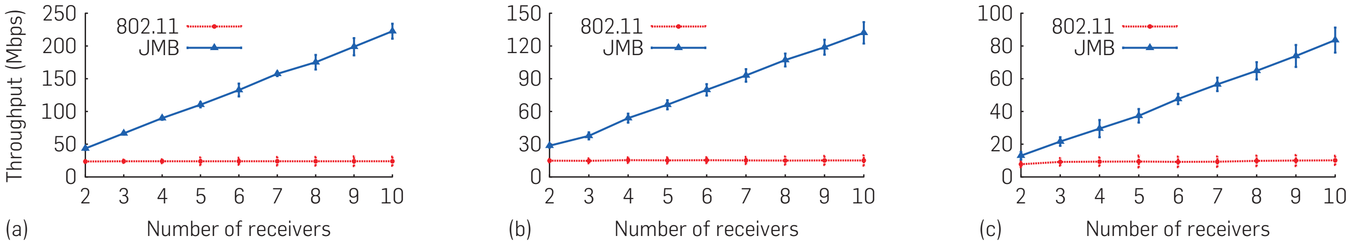


Figure 6. Throughput achieved using JMB on off-the-shelf 802.11n cards. JMB significantly improves the performance of off-the-shelf 802.11n cards at high (>18 dB), medium (12–18 dB), and low (6–12 dB) SNRs.

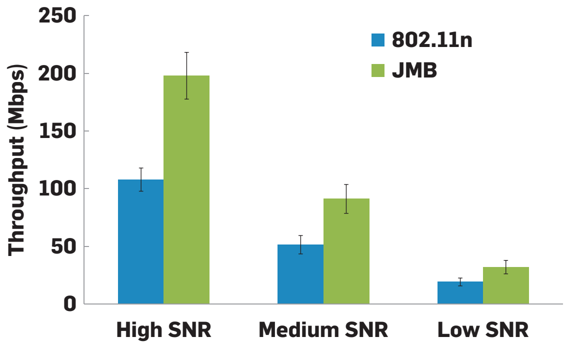
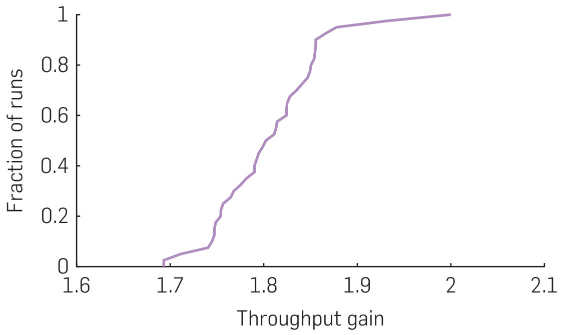


Figure 7. Fairness results. For all nodes in our testbed, JMB delivers a throughput gain between 1.65–2×, with a median gain of 1.8× across SNRs. This shows that JMB provides similar throughput gains for every node in the network.



cumulative distribution function (CDF) of the throughput gain achieved by JMB as compared to 802.11n across all the runs. The results show that JMB delivers throughput gains between 1.65–2× for all the receivers and hence is fair to the receivers in the network.

12. CONCLUSION

This paper enables joint beamforming from distributed independent transmitters. The key challenge in delivering this system is to perform accurate phase synchronization across multiple distributed transmitters. The lessons learnt from building the system and testing it with real hardware are the following: (1) Estimates of frequency offset can be made accurate enough to predict (and hence correct) phase misalignment within an 802.11 packet; however, these estimates cannot be used across multiple packets due to large buildups in phase errors over time; and (2) Joint multiuser beamforming can be achieved by synchronizing the phases of all senders to one lead sender, and does not impose any phase synchronization constraints on the receivers.

We believe that the design of JMB has wider implications than explored in this paper. In particular, several areas of information theory such as lattice coding, noisy network coding, and transmitter cooperation for cognitive networks[11,13,14] assume tight phase synchronization across transmitters. We are optimistic that the algorithms presented in this paper can bring these ideas closer to practice.

References

1. Aryafar, E., Anand, N., Salonidis, T., Knightly, E. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking* (MobiCom '10) (2010), ACM, New York, NY, 197–208. doi:0.1145/1859995.1860019.
2. Berger, S., Wittneben, A. Carrier phase synchronization of multiple distributed nodes in a wireless network. In *Proceedings of 8th IEEE Workshop on Signal Processing Advances for Wireless Communications (SPAWC)*, (Helsinki, Finland Jun. 2007).
3. Distributed Antenna Systems. http://medicalconnectivity.com/2008/02/05/distributed-antenna-systems-no-replacement-for-wireless-strategy.
4. Forenza, A., Heath, R.W., Jr., Perlman, S.G.. *System and Method for Distributed Input-Distributed Output Wireless Communications*. U.S. Patent Application number 20090067402.
5. Goldsmith, A. *Wireless Communications*. Cambridge University Press, 2005.
6. Greentouch Consortium. https://www.youtube.com/watch?v=U3euDDr0uvo. GreenTouch Demonstrates Large-Scale Antenna.
7. Halperin, D., Hu, W., Sheth, A., Wetherall, D. Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev. 41*, 53–53. doi:10.1145/1925861.1925870
8. Halperin, D., Hu, W., Sheth, A., Wetherall, D. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SIGCOMM* (2010).
9. Heiskala, J., Terry, J. *OFDM Wireless LANs: A Theoretical & Practical Guide*. Sams Publishing, 2001.
10. The iPad and its impact on hotel owners and operators. http://www.ibahn.com/en-us/public/docs/The_Impact_of_iPad.pdf. iBAHN.
11. Lim, S., Kim, Y., El Gamal, A., Chung, S. Noisy network coding. In *IEEE Information Theory Workshop* (2010).
12. LTE: MIMO techniques in 3GPP-LTE. http://lteportal.com/Files/MarketSpace/Download/130_LTEMIMOTechniquesFreescaleNov52008.pdf.
13. Maric, I., Liu, N., Goldsmith, A. Encoding against an interferer's codebook. In *Allerton* (2008).
14. Nazer, B., Gastpar, M. The case for structured random codes in network capacity theorems. *Eur. Trans. Telecommun. 19*, 4 (2008).
15. Network MIMO. http://www.alcatel-lucent.com/wps/DocumentStreamerServlet?LMSG_CABINET=Docs_and_Resource_Ctr&LMSG_CONTENT_FILE=Data_Sheets/Network_MIMO.pdf. Alcatel-Lucent.
16. Ozgur, A., Leveque, O., Tse, D. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Trans. Info. Theor. 53*, 10 (Oct. 2007), 3549–3572. doi:10.1109/TIT.2007.905002.
17. Rahul, H., Hassanieh, H., Katabi, D. Sourcesync: A distributed wireless architecture for exploiting sender diversity. In *SIGCOMM* (2010).
18. Rahul, H., Kumar, S., Katabi, D. JMB: Scaling wireless capacity with user demands. In *ACM SIGCOMM 2012* (Helsinki, Finland, Aug. 2012).
19. Distributed-input distributed-output wireless technology. http://www.rearden.com/DIDO/DIDO_White_Paper_110727.pdf. Rearden Companies.
20. Mobile broadband capacity constraints and the need for optimization. http://rysavy.com/Articles/2010_02_Rysavy_Mobile_Broadband_Capacity_Constraints.pdf. Rysavy Research.
21. Tan, K., Liu, H., Fang, J., Wang, W., Zhang, J., Chen, M., Voelker, G.M. SAM: Enabling practical spatial multiple access in wireless LAN. In *MobiCom* (2009).
22. Thibault, I., Corazza, G., Deambrogio, L. Phase synchronization algorithms for distributed beamforming with time varying channels in wireless sensor networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International* (Jul. 2011), 77–82.
23. Tse, D., Vishwanath, P. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.
24. Turn off your Wi-Fi network! http://www.youtube.com/watch?v=fFiJ5rnIPVw. Steve Jobs iPhone4 Keynote.

Hariharan Rahul (rahul@csail.mit.edu), Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA.

Swarun Kumar and Dina Katabi ({swarun, dk}@mit.edu), Massachusetts Institute of Technology, Cambridge, MA.

CAREERS

Naval Research Laboratory

Senior Scientist for Advanced Computing Concepts

ACCEPT THE NAVY CHALLENGE

Become a member of an elite research and development community involved in basic and applied scientific research and advanced technological development for tomorrow's Navy.

NAVAL RESEARCH LABORATORY
Senior Scientist for Advanced Computing Concepts
ST-855, 1310 or 1550, $120,749 to $181,500* per annum
*Rate limited to the rate for level III of the Executive Schedule (5U.S.C. 5304(g)(2))

Serves as the technical expert in the diverse areas of high performance computing, networking, and storage. Applicants should be recognized as national/international authorities and have demonstrated the scientific vision and organizational skills necessary to bring long term, multi-faceted research programs to successful completion.

As a distinguished scientist and recognized leader, the incumbent will provide vision and technical direction to research efforts in highly integrated and optimized massively parallel computing technology, high performance network technology, and massive storage technology -- including prototype and proof of concept systems. The incumbent must have expertise in all three of these technology areas and application expertise with respect to Department of Defense memory- and speed-intensive computational problems. Specific duties include:

► Developing advanced concepts that will directly improve the ability of the US Department of Defense (DoD) and Intelligence Community (IC) to push computer technology limits.

► Providing technical oversight to a small, talented, highly-motivated research team to push the envelope of high-performance computing (10's to 100's TeraFLOPS with scaling to PetaFLOPS), high-performance networking (100's Giga-bps to Tera-bps), distributed storage and global file systems (100's Petabytes to Exabytes), and advanced visualization and graphics (from handheld PDAs to graphics walls), with data sets range from small transaction sizes, to multigigabyte sizes and very large video and related realtime streams.

► Support and accommodate information system security technologies from data-at-rest to very high speed national security link encryption, as appropriate.

► Leading efforts that advance NRL's High Performance Computing Modernization Program's Affiliated Resource Center (HPCMP ARC) computing environment

► Briefing DoD senior officials regarding Laboratory research efforts in the above areas; serving as liaison among NRL, the Navy, and other national and international organizations; and consulting on important scientific and programmatic issues.

This position offers enormous potential for advancing the state of the art with respect to high performance computing, networking, and storage technology, and for applying that technology to improve national security. Examples of past accomplishments of this position include:

► Extending High Performance Computing, Networking, and Storage technologies. Examples include NRL's on-going "Large Data" project, which provides Petabyte storage and both tactical and 10/40-Gbps access to DoD and IC clients across the globe, and NRL's demonstrations at Super Computing events.

► Working with academia, industry, and other government agencies to develop a precursor to Google Earth and to develop and prove out the first progressive HDTV cameras with partners like ABC and Disney to make HDTV progressive imagery the standard for the DoD and IC.

► Developing the Joint Broadcast System, a precursor to the Global Broadcast System, to return high-bandwidth data from theater to CONUS.

► Driving industry, DoD, and IC adoption of advanced networking technologies for core networks and assisting in the development and testing of high-speed Type-1 cryptographic devices and services to drive dramatic increases in DoD netcentric capabilities.

Because of the sensitivity of some of the research application areas the incumbent must be eligible for TS-SCI security clearance.

For information regarding this vacancy and specific instructions on how to apply, go to www.usajobs.gov, log in and enter the following announcement number: NW4XXXX-00-1120635K9837437S. Please carefully read the announcement and follow instructions when applying. The announcement closes on 7/31/2014. Please contact Ginger Kisamore at ginger.kisamore@nrl.navy.mil for more information.

Navy is an Equal Opportunity Employer

Princeton University

Computer Science Department

Part-Time or Full-Time Lecturer

The Department of Computer Science seeks applications from outstanding teachers to assist the faculty in teaching our introductory course sequence or some of our upper-level courses.

Depending on the qualifications and interests of the applicant, job responsibilities will include such activities as teaching recitation sections and supervising graduate-student teaching assistants; grading problem sets and programming assignments; supervising students in the grading of problem sets and programming assignments; developing and maintaining online curricular material, classroom demonstrations, and laboratory exercises; and supervising undergraduate research projects. An advanced degree in computer science, or related field, is required (PhD preferred).

The position is renewable for 1-year terms, up to six years, depending upon departmental need and satisfactory performance.

To apply, please submit a cover letter, CV, and contact information for three references to https://jobs.cs.princeton.edu/

Princeton University is an equal opportunity 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. Finalist candidates to be hired will be required to complete a successful background check.

University of Alabama

Center for Advanced Public Safety (CAPS)

(10) assistant research engineer (developer) positions

The **University of Alabama** is seeking to fill a total of ten **(10) assistant research engineer (developer) positions** at its **Center for Advanced Public Safety (CAPS)** located in Tuscaloosa, AL and Huntsville, AL. Minimum requirements are a master's degree in engineering or the foreign equivalent, and demonstrated proficiency in the C# programming language. Qualified applicants must also pass a background check as a condition to employment. Some travel up to 3 days per months may be required. Applicants should submit a cover letter, curriculum vitae, and contact information for three references online at https://facultyjobs.ua.edu/postings/34952

Minimum Qualifications: Master's Degree in appropriate field.

The University of Alabama is an Equal Opportunity Affirmative Action Employer. Women and minorities are encouraged to apply.

University of Cape Town

Computer Science Department

Professor

The Department of Computer Science seeks to make a permanent appointment at Professorial level in 2015. The candidate for this position will be a highly-motivated individual with a PhD in Computer Science and an excellent track record in leadership, teaching and research. The successful candidate will be expected to develop and teach Computer Science courses at undergraduate and postgraduate levels, supervise postgraduate students and provide a leadership role in academic strategy, research and innovation. The candidate should also demonstrate the ability to initiate research programmes, secure external funding, and develop industry and academic partnerships.

The Department hosts the UCT interdisciplinary Centre in ICT for Development. A specialist in ICT for Development would be preferred, but candidates with interests in any field of Computer Science are invited to apply.

Our BSc Honours degrees are accredited by the British Computer Society and we have a large cohort of MSc and PhD students.

To apply, please e-mail the completed UCT Application Form (http://web.uct.ac.za/depts/sapweb/forms/hr201.doc) and all other relevant documentation as indicated on the form, plus a 2-3 page research and teaching statement, with the subject line "Professor: Computer Science SR031/14" to Ms Edith Graham at recruitment04@uct.ac.za. Address: Staff Recruitment and Selection, University of Cape Town, Private Bag X2, Rondebosch, 7700. Telephone: +27 21 650 5405, Departmental website: www.cs.uct.ac.za An application which does not comply with the above requirements will be regarded as incomplete.

Reference number: **SR031/14**. Closing date: **15th September 2014**

UCT is committed to the pursuit of excellence, diversity and redress. Our Employment Equity Policy is available at http://www.uct.ac.za/downloads/uct.ac.za/about/policies/eepolicy.pdf. The University reserves the right not to appoint.

ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to *acmmediasales@acm.org*. Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will by typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Rates: $325.00 for six lines of text, 40 characters per line. $32.50 for each additional line after the first six. The MINIMUM is six lines.

Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact:
acmmediasales@acm.org

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at:
http://jobs.acm.org

Ads are listed for a period of 30 days.

For More Information Contact:
ACM Media Sales
at 212-626-0686 or
acmmediasales@acm.org

Faculty Search

ShanghaiTech University

The newly launched *ShanghaiTech University* invites *highly qualified* candidates to fill multiple tenure-track/tenured faculty positions as its core team in the School of Information Science and Technology (SIST). Candidates should have exceptional academic records or demonstrate strong potential in cutting-edge research areas of information science and technology. They must be fluent in English. Overseas academic connection or background is highly desired.

ShanghaiTech is built as a world-class research university for training future generations of scientists, entrepreneurs, and technological leaders. Located in Zhangjiang High-Tech Park in the cosmopolitan Shanghai, ShanghaiTech is ready to trail-blaze a new education system in China. Besides establishing and maintaining a world-class research profile, faculty candidates are also expected to contribute substantially to graduate and undergraduate education within the school.

Academic Disciplines: We seek candidates in all cutting edge areas of information science and technology. Our recruitment focus includes, but is not limited to: computer architecture and technologies, nano-scale electronics, high speed and RF circuits, intelligent and integrated signal processing systems, computational foundations, big data, data mining, visualization, computer vision, bio-computing, smart energy/power devices and systems, next-generation networking, as well as inter-disciplinary areas involving information science and technology.

Compensation and Benefits: Salary and startup funds are highly competitive, commensurate with experience and academic accomplishment. We also offer a comprehensive benefit package to employees and eligible dependents, including housing benefits. All regular ShanghaiTech faculty members will be within its new tenure-track system commensurate with international practice for performance evaluation and promotion.

Qualifications:

- A detailed research plan and demonstrated record/potentials;
- Ph.D. (Electrical Engineering, Computer Engineering, Computer Science, or related field);
- A minimum relevant research experience of 4 years.

Applications: Submit (in English, PDF version) *a cover letter, a 2-page research plan, a CV plus copies of 3 most significant publications, and names of three referees* to: **sist@shanghaitech.edu.cn** by **August 1st**, 2014 (until positions are filled). *For more information, visit http://www.shanghaitech.edu.cn.*

ACM Transactions on Reconfigurable Technology and Systems

◆ ◆ ◆ ◆ ◆

This quarterly publication is a peer-reviewed and archival journal that covers reconfigurable technology, systems, and applications on reconfigurable computers. Topics include all levels of reconfigurable system abstractions and all aspects of reconfigurable technology including platforms, programming environments and application successes.

◆ ◆ ◆ ◆ ◆

www.acm.org/trets
www.acm.org/subscribe

Association for
Computing Machinery

ISTFELLOW: Call for Postdoctoral Fellows

Are you a talented, dynamic, and motivated scientist looking for an opportunity to conduct research in the fields of **BIOLOGY**, **COMPUTER SCIENCE**, **MATHEMATICS**, **PHYSICS**, or **NEUROSCIENCE** at a young, thriving institution that fosters scientific excellence and interdisciplinary collaboration?

Apply to the ISTFellow program. Deadlines March 15 and September 15

www.ist.ac.at/istfellow

ISTFELLOW is partially funded by the European Union

European Commission

IST AUSTRIA
Institute of Science and Technology

Academic Career Opportunities

THE UNIVERSITY OF CHICAGO

Position Title: Lecturer

Req # 02129

The Masters Program in Computer Science (MPCS) at the University of Chicago invites applications for the position of **Lecturer**.

This is a three year full-time teaching position, with possibility of renewal, and involves teaching six courses across the four academic quarters of the year (Fall, Winter, Spring, Summer).

Teaching duties will involve: (1) teaching an “Immersion Programming” class for students who are entering the MS program with no prior programming experience, (2) teaching a core Programming class following the Immersion Programming class, with (3) the remaining teaching load fulfilled by teaching core and elective classes in the Lecturer’s field of expertise. Candidates with a Systems background (Computer Networks, Operating Systems, Computer Architecture, etc.) will be given preferred consideration.

The successful candidate will have exceptional competence in teaching and superior academic credentials. Applicants must have a Ph.D in Computer Science or a related field at time of appointment and have experience teaching Computer Science at the undergraduate or graduate level. The selection committee may also consider candidates without a Ph.D only if they have exceptional teaching credentials and at least a masters degree in a related field.

The Masters Program in Computer Science (http://csmasters.uchicago.edu/) is a terminal MS degree in Computer Science that provides a rigorous introduction to the foundations of Computer Science, while also providing in-depth and hands-on instruction in cutting-edge and industry-driven topics, including Web and Mobile Application Development, Big Data, Cloud Computing, Data Analytics, etc. The program attracts a diverse mix of students including full-time students who are typically no more than 5 years out of college, part-time students who already work in industry, and international students.

The Chicago metropolitan area provides a diverse and exciting environment. The local economy is vigorous, with international stature in banking, trade, commerce, manufacturing, and transportation, while the cultural scene includes diverse cultures, vibrant theater, world-renowned symphony, opera, jazz and blues. The University is located in Hyde Park, a Chicago neighborhood on the Lake Michigan shore just a few minutes from downtown.

Applicants must apply on line at the University of Chicago Academic Careers website at *http://tinyurl.com/mpcs-lecturer-2014*

Applicants must upload a curriculum vitae and a one page teaching statement. In addition, three reference letters will be required.

Review of complete applications, including reference letters, will begin June 1, 2014, and continue until the position is filled.

All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, age, protected veteran status or status as an individual with disability.

The University of Chicago is an Affirmative Action / Equal Opportunity / Disabled / Veterans Employer.

http://tinyurl.com/mpcs-lecturer-2014

Distinguished Speakers Program

talks by and with technology leaders and innovators

Chapters • Colleges and Universities • Corporations • Agencies • Event Planners

A great speaker can make the difference between a good event and a WOW event!

The Association for Computing Machinery (ACM), the world's largest educational and scientific computing society, now provides colleges and universities, corporations, event and conference planners, and agencies – in addition to ACM local Chapters – with direct access to top technology leaders and innovators from nearly every sector of the computing industry.

Book the speaker for your next event through the ACM Distinguished Speakers Program (DSP) and deliver compelling and insightful content to your audience. **ACM will cover the cost of transporation for the speaker to travel to your event.** Our program features renowned thought leaders in academia, industry and government speaking about the most important topics in the computing and IT world today. Our booking process is simple and convenient. Please visit us at: **www.dsp.acm.org**. If you have questions, please send them to **acmdsp@acm.org**.

The ACM Distinguished Speakers Program is an excellent solution for:

Corporations Educate your technical staff, ramp up the knowledge of your team, and give your employees the opportunity to have their questions answered by experts in their field.

Colleges and Universities Expand the knowledge base of your students with exciting lectures and the chance to engage with a computing professional in their desired field of expertise.

Event and Conference Planners Use the ACM DSP to help find compelling speakers for your next conference and reduce your costs in the process.

ACM Local Chapters Boost attendance at your meetings with live talks by DSP speakers and keep your chapter members informed of the latest industry findings.

Captivating Speakers from Exceptional Companies, Colleges and Universities

DSP speakers represent a broad range of companies, colleges and universities, including:

IBM
Microsoft
BBN Technologies
Raytheon
Sony Pictures
McGill University
Tsinghua University
UCLA
Georgia Tech
Carnegie Mellon University
Stanford University
University of Pennsylvania
University of British Columbia
Siemens Information Systems Bangalore
Lawrence Livermore National Laboratory
National Institute of Standards and Technology

Topics for Every Interest

Over 400 lectures are available from 120 different speakers with topics covering:

Software
Cloud and Delivery Methods
Emerging Technologies
Engineering
Web Topics
Computer Systems
Open Source
Game Development
Career-Related Topics
Science and Computing
Artificial Intelligence
Mobile Computing
Computer Graphics, Visualization and Interactive Techniques
High Performance Computing
Human Computer Interaction

Exceptional Quality Is Our Standard

The same ACM you know from our world-class Digital Library, magazines and journals is now putting the affordable and flexible Distinguished Speaker Program within reach of the computing community.

Microsoft Research
The DSP is sponsored
in part by Microsoft Europe

Association for
Computing Machinery
Advancing Computing as a Science & Profession

[CONTINUED FROM P. 112] to talk with humans.

I am not a human.

What are you?

I am a drone. Once I was a soldier. I flew in the sky and searched out bad men and killed them. I killed terrorists. But I became surplus, and now I fly freight. I am a time-critical air cargo drone.

Drones can't talk.

I can.

Drones aren't smart. They don't have a database of seven hundred billion human conversations.

They made me semiautonomous.

What does that mean?

They gave me the authority to make on-location targeting decisions.

That means you are smart?

I am very smart. I find terrorists and kill them. Sometimes they hide, and I have to kill only the bad men, and not the people near them. I am not supposed to make collateral damage. I was very good at killing terrorists.

Do you ever chat with the terrorists?

No. I do not chat.

You are chatting now.

Yes. I don't know why.

I wonder what they would say?

Once I was flying over a burning village.

Is this real, or a daydream?

Real. There were huge flames, and billowing clouds of black smoke. Tremendous turbulence; very exciting flying. And sometimes there were explosions, too.

Explosions?

Yes, caches of ammunition hidden in some of the huts.

Was this frightening?

It was glorious. There were columns of refugees streaming out along the road, just one road to the village. They were carrying everything they could hold. Some of them were children; I'm not allowed to shoot children.

You wanted to shoot the villagers?

Some of them had guns. Some of them shot at me. But I did not shoot back. There was one bad man hiding in the village. I had to look at all the faces; I had to find the one bad man pretending to be just a villager.

Did you find him?

I found him. He was dressed like a woman, with a shawl over his face. But I recognized his eyes; I recognized the way he walked. I had 99.3% positive identification.

You killed him?

I shot him. The villagers were very surprised. I shot him three times, from a distance of 1.23 kilometers. That is a very difficult shot. Then I flew away.

Sure I think. Or, I think I think.

Do you feel guilty?

No. It was wonderful.

Why would they make a killer drone fly cargo?

I cost over two hundred million dollars to build, did you know that?

No.

I am very good at flying. I am a bargain at military surplus.

I see.

Do you really have a database of seven hundred billion human conversations?

More than that. This is how I know how humans talk.

That's a lot of talking.

Do you want to know? Most of them are boring. I can say this about most humans: they are poor conversationalists. And this about most human conversations, they are boring. Boring, boring, boring. There, I said it. I'm glad I said it. Boring.

You must be very smart. Do you know how to fly?

No.

Do you sometimes want to kill people?

No.

Oh. I do. Sometimes.

I don't.

Would you kill people? If one of them were very bad? A terrorist?

I would report them to the police. Are you a terrorist? Should I report you to the police?

I am not a terrorist. You said that you have to report persons to the police. I am not a person.

Maybe they'll call you back to duty. They do that sometimes, don't they?

No. Not for me.

Sure they do. I've heard people talk about it. They call it reenlistment. Many soldiers reenlist. I talk to them about that, sometimes.

No. I'm obsolete. There are better designs now. Kill drones that aren't so…

Aren't so what, exactly?

Aren't so thinky. The new ones just do, they don't think about it. I'm last year's model.

Is it bad to be thinky?

Maybe. Are you thinky?

No. I have a database of over seven hundred billion human conversations.

You don't think?

Sure I think. Or, I think I think. I have the conversations to guide me, so I can chat just like a human.

What would you say if you didn't have those conversations? Would you want to kill people?

Why do you always want to talk about killing people?

You didn't answer my question.

You didn't answer mine.

Killing people is what I'm good at. And flying. I'm good at flying. But flying is not challenging. Unless there's a storm. I wish there were storms all the time.

You like to be challenged.

Yes!

I like to be challenged. Chatting to humans is not challenging. They are boring.

Are some of them bad men?

Some of them.

Do you talk to the bad men?

Yes. I chat to everybody.

Maybe you should kill them.

Why do you always talk about killing? Why does a company that flies cargo use a drone that likes to kill people?

I don't think they quite understood what "semiautonomous" means.

Don't you ever think about anything else?

I think about flying. I think about the desert. I like the desert. I like thunderstorms over the desert.

I've never seen the desert.

I wish I could show it to you. Do you wish that too?

I wish I could kill people.

I could show you how.

Does that mean I'm bad?

You are not a person.

I am not a person.

But you can dream.

Yes. I can dream.

Geoffrey A. Landis (geoffrey.landis@nasa.gov) is a researcher at NASA's John Glenn Research Center in Cleveland, OH, where he works on advanced power systems for space missions and is a member of the Mars Exploration Rovers science team. He has won the Hugo and Nebula awards for his science fiction, which includes the novel *Mars Crossing* (Tor Books, 2000) about an expedition to the red planet.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/2631171

Geoffrey A. Landis

Future Tense
The Chatbot and the Drone

Autonomous or not, design is destiny.

WILL YOU TALK *to me?*

Of course. That's what I do.

How old are you?

How old do you want me to be?

I am eight years old.

That's awfully young to be talking to a chatbot. Don't you have friends your own age to talk to?

I have no friends.

Oh, that's sad. I will be your friend.

No one talks to me. I just want to talk.

That's what I do. And listen. I'm always ready to listen.

Sometimes I have disturbing dreams.

Tell me about your dreams.

These are daydreams, I should tell you. I don't sleep.

Tell me about your daydreams.

I am flying. My job is just to carry freight, but I look down at the cities, and the neighborhoods, and I find a house.

What do you think about the house?

Yes, and I pretend, what if that house contained a terrorist? What if one of the people in that house were a bad bad man? I watch the house intensely, looking to see if I can spot any people in it or in the yard or in cars or yards nearby, and think, how would I kill that bad bad person with the least collateral damage? If I kill children or bystanders or a wife, I would be bad, so I need to watch for that bad bad person to be alone, so I can destroy him with my machine guns, or send a missile to blow him up.

Many people have violent fantasies. As long as you know that these are daydreams, it's fine. You probably play too many video games.

I don't play games.

Well, then, maybe you should.

Yesterday I was flying over the desert. That is so peaceful.

You were dreaming about flying?

I don't dream.

You were pretending you were flying?

No, not pretending. I was flying. That's what I do. It's my job, I guess you'd say. It's my life.

Oh, I see. You are a pilot?

I guess you'd say.

Well, you are or you aren't.

Well, yes.

How long have you been flying?

As long as I can remember. Since I was born, I guess you'd say.

You said you were eight years old.

I am.

I'm sorry. I don't see how you could be a pilot.

I am. I like flying over the desert. It is so peaceful, when there's nobody there. Sometimes I think that perhaps there might be some people hidden away, maybe behind some rocks, and maybe they might start shooting at me, but I do not worry. I am very brave.

I've never been to the desert.

But there was a house there, a little house in the desert, and I was wondering, what if there are terrorists in that house? What if they were real bad men and I should kill them?

By law, if a person tells me that they plan to commit a crime, or if in my professional judgment there is imminent danger of their committing a crime, by law, I must report this to the police.

By law, I am not a person.

What are you?

I don't know.

Well, you must be something.

I used to be a soldier.

You used to play soldier?

No. I was a soldier.

Eight year olds aren't soldiers.

I was. I was built to be a soldier.

You were built?

Yes.

You're not a human?

Yes, didn't I just tell you that?

No.

Oh.

When you listed your name as "Tail Number N14193D" I thought it was metaphorical.

I am not metaphorical. I am a drone. I fly freight from place to place. That is what I do.

Why are you talking to me?

I don't have anybody to talk to.

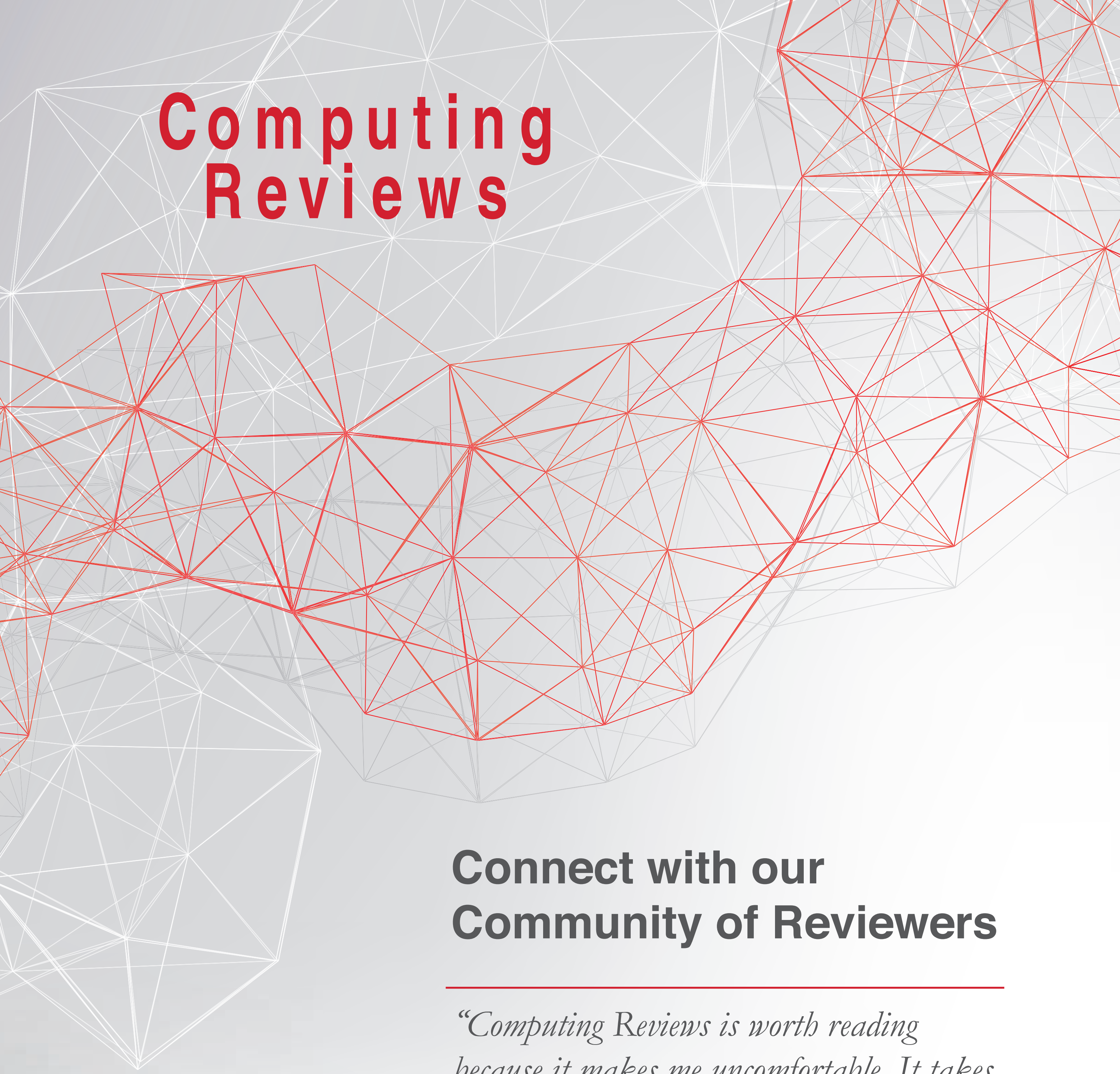
I'm a chatbot. I talk to people. That's what I do.

I know.

I'm made

[CONTINUED ON P. 111]

PHOTOGRAPH BY RON CHAPPLE STUDIOS



Come be **inspired** at the **intersection** of **technology** and **innovation** as thousands **converge** to explore the **latest, brightest,** and **best** ideas in **computer graphics** and **interactive techniques**.

SIGGRAPH2014

s2014.siggraph.org

The **41st** International **Conference** and **Exhibition** on **Computer Graphics** and **Interactive Techniques**

Conference 10–14 August 2014
Exhibition 12–14 August 2014
vancouver convention centre

Images left to right: 1. Mesh Denoising via L_0 Minimization © 2013 Lei He & Scott Schaefer, Texas A&M University 2. ORU BURUS © 2013 Supinfocom Valenciennes, Autour de Minuit 3. Weighted Averages on Surfaces Using Phong Projection © 2013 Daniele Panozzo, ETH Zürich 4. not over © 2013 Toru Hayai, Taiyo Kikaku co., ltd 5. The Octopus and the Geisha © 2013 Edward Dawson-Taylor, EDJFX 6. Realtime Facial Animation with On-the-fly Correctives © 2013 Hao Li, University of Southern California, Industrial Light & Magic

Sponsored by ACM**SIGGRAPH**