

COMMUNICATIONS

CACM.ACM.ORG

OF THE

ACM

11/08 VOL.51 NO.11

Remembering Jim Gray

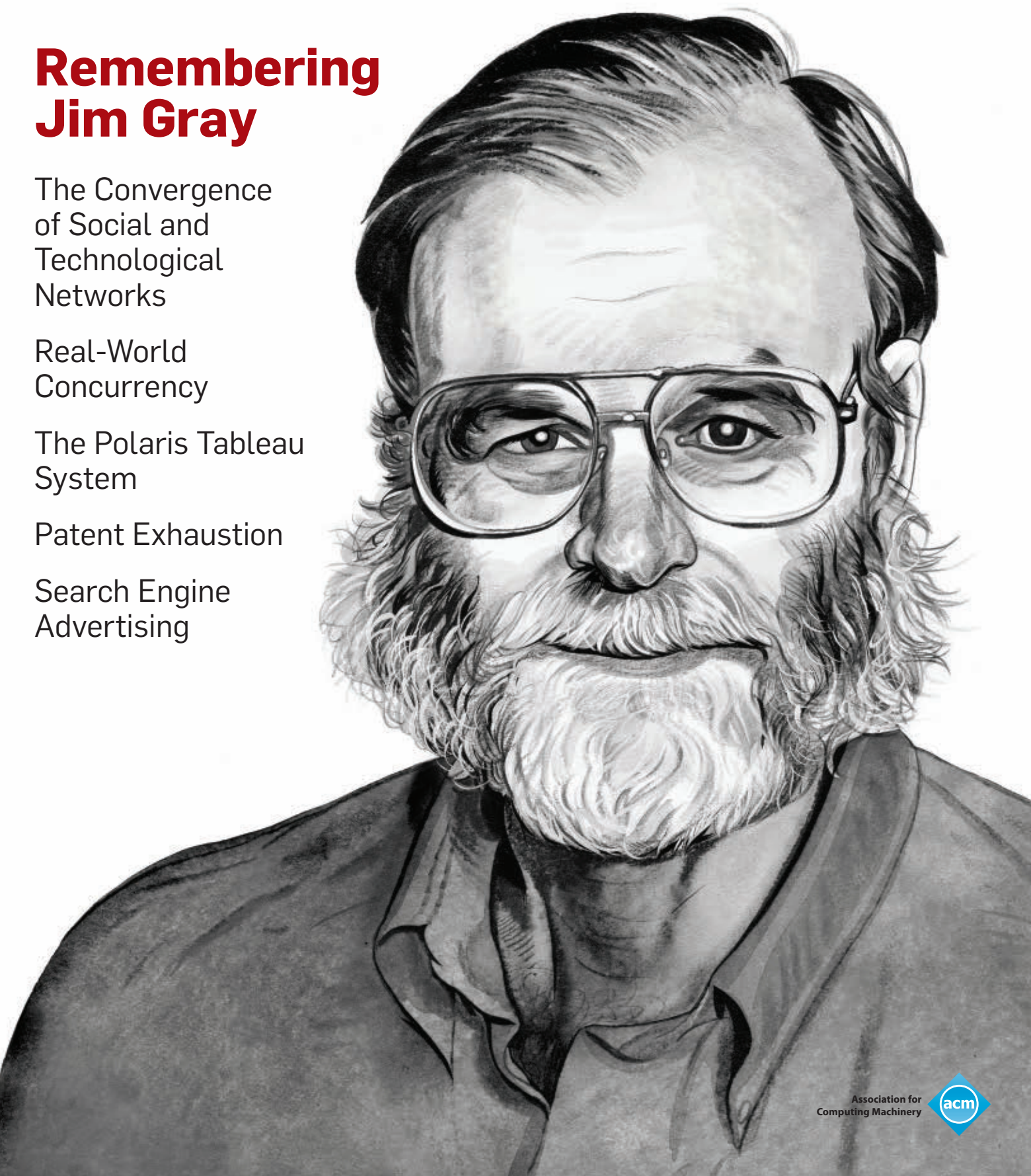
The Convergence
of Social and
Technological
Networks

Real-World
Concurrency

The Polaris Tableau
System

Patent Exhaustion

Search Engine
Advertising



New and Noteworthy



Security in Computing Systems

Challenges, Approaches, and Solutions

J. Biskup, University of Dortmund, Germany

This monograph provides a broad and comprehensive description of computer security threats and countermeasures, ideal for graduate students or researchers in academia and industry who require an introduction to the state of the art in this field. In addition, it can be used as the basis for graduate courses on security issues in computing.

2009. Approx. 700 p. Hardcover
ISBN 978-3-540-78441-8 ► **\$99.00**



The IT Measurement Compendium

Estimating and Benchmarking Success with Functional Size Measurement

M. Bundschuh, Bergisch-Gladbach, Germany; C. Dekkers, Quality Plus Technologies Inc., Seminole, FL, USA

Based on their many years of practical experience as software managers and consultants, Manfred Bundschuh and Carol Dekkers present a framework of value to anyone involved with software project management. They present all five ISO/IEC-acknowledged Functional Sizing Methods, with variants, experiences, counting rules and case studies.

2008. Approx. 650 p. Hardcover
ISBN 978-3-540-68187-8 ► **\$99.00**



The Semantic Web

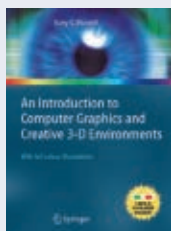
Semantics for Data and Services on the Web

V. Kashyap, Partners HealthCare Systems, Boston, MA, USA;

C. Bussler, DERI, Galway, Ireland; M. Moran, Nortel Networks Ltd., Galway, Ireland

With this textbook, the authors deliver an application-driven state-of-the-art presentation of Semantic Web technologies, ideally suited for academic courses on the Semantic Web and architectures of information systems, and for self-studying professionals engaged in the design and implementation of advanced application systems.

2009. Approx. 410 p. (Data-Centric Systems and Applications) Hardcover
ISBN 978-3-540-76451-9 ► **\$79.95**



An Introduction to Computer Graphics and Creative 3-D Environments

B. G. Blundell, Auckland University of Technology, Auckland, New Zealand

This book introduces the fundamentals of 2-D and 3-D computer graphics. Additionally, a range of emerging, creative 3-D display technologies are described, including stereoscopic systems, immersive virtual reality, volumetric, varifocal, and others. Included with the book are anaglyph, stereoscopic, and Pulfrich viewing glasses

2008. Approx. 500 p. 248 illus., 227 in color. Hardcover
ISBN 978-1-84800-041-4 ► **\$69.95**



RFID in Manufacturing

O. Günther, Humboldt-Universität zu Berlin, Germany; W. Kletti, MPDV Mikrolab GmbH, Mosbach, Germany; U. Kubach, SAP Research, Dresden, Germany

The authors of this book clearly explain the potential advantages of using Radio Frequency Identification (RFID) technology in a modern manufacturing and supply chain context. Areas of emphasis include integration of RFID data into legacy IT architectures, RFID-MES-ERP integration, and cost-benefit considerations.

2008. XVI, 163 p. 47 illus. Hardcover
ISBN 978-3-540-76453-3 ► **\$59.95**



Networked RFID Systems, Software and Services

G. Roussos, University of London, UK

This book introduces the technologies and techniques of large-scale

RFID-enabled mobile computing systems. The discussion is set in the context of specific system case studies where RFID has been the core enabling technology in retail, metropolitan transportation, logistics and e-passport applications.

RFID technology fundamentals are covered including operating principles, core system components and performance trade-offs involved in the selection of specific RFID platforms.

2008. Approx. 210 p. 64 illus. With online files/update. (Computer Communications and Networks) Softcover
ISBN 978-1-84800-152-7 ► **\$79.95**

CALL FOR PARTICIPATION
CTS 2009
Baltimore, Maryland, USA



**The 2009 International Symposium on
Collaborative Technologies and Systems**

May 18 – 22, 2009
The Westin Baltimore Washington International Airport Hotel
Baltimore, Maryland, USA

Important Dates:

Paper Submission Deadline -----	December 20, 2008
Workshop/Special Session Proposal Deadline -----	December 1, 2008
Tutorial/Demo/Panel Proposal Deadline -----	January 8, 2009
Notification of Acceptance -----	February 3, 2009
Final Papers Due -----	March 3, 2009

Conference Co-Chairs:

William McQuay, Air Force Research Laboratory, Wright Patterson AFB, USA
Waleed W. Smari, University of Dayton, USA

For more information, visit the CTS 2009 web site at:
<http://cisedu.us/cis/cts/09/main/callForPapers.jsp>



In cooperation with the IEEE, IFIP

Departments

- 5 **CEO's Letter**
On the 10th Anniversary of ACM's Digital Library
By John R. White
-
- 7 **Executive Editor's Corner**
Jim Gray: Humble Visionary
By Diane Crawford
-
- 8 **Letters To The Editor**
Even Science Would Benefit from Auctions
-
- 10 **CACM Online**
A First Look at the Redesigned Site
-
- 96 **Careers**

Last Byte

- 112 **Puzzled**
Circular Food
By Peter Winkler

News

- 11 **Damage Control**
The U.S. patent system is overdue for reform, but what needs fixing, and how, is a matter of some dispute.
By Leah Hoffmann
-
- 14 **Analyzing Online Social Networks**
Social network analysis explains why some sites succeed and others fail, how physical and online social networks differ and are alike, and attempts to predict how they will evolve.
By Bill Howard
-
- 17 **The Limits of Computability**
Computational complexity and intractability may help scientists better understand how humans process information and make decisions.
By David Lindley

Viewpoints

- 22 **Economic and Business Dimensions**
Search Engine Advertising
Examining a profitable side of the long tail of advertising that is not possible under the traditional broadcast advertising model.
By Avi Goldfarb and Catherine Tucker
-
- 25 **Privacy and Security**
A Multidimensional Problem
It's not just science or engineering that will be needed to address security concerns, but law, economics, anthropology, and more.
By Susan Landau
-
- 27 **Legally Speaking**
Quantafying the Value of Patent Exhaustion
Should patents confer power to restrict reuses and redistributions of products embodying the whole or essential parts of inventions?
By Pamela Samuelson
-
- 31 **Education**
Reprogramming College Preparatory Computer Science
The college preparatory computer science education curriculum must be improved, beginning with the earliest phases of the process.
By Joanna Goode

Practice



- 34 **Real-World Concurrency**
What does the proliferation of concurrency mean for the software you develop?
By Bryan Cantrill and Jeff Bonwick
-
- 40 **Software Transactional Memory: Why is it Only a Research Toy?**
The promise of STM may likely be undermined by its overheads and workload applicabilities.
By Călin Cașcaval, Colin Blundell, Maged Michael, Harold W. Cain, Peng Wu, Stefanie Chiras, and Siddhartha Chatterjee
-
- 47 **CTO Roundtable on Virtualization**
Virtualization technology is hot again, but for the right reasons?
By Mache Creeger, Moderator

Contributed Articles

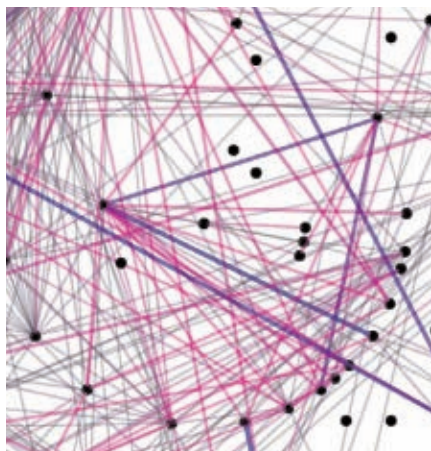


- 54 **A Tribute to Jim Gray**
We knew him as both scholar and friend.
By Michael Stonebraker and David J. DeWitt
-
- 58 **Jim Gray, Astronomer**
How he helped develop the SkyServer, delivering computation directly to terabytes of astronomical data.
By Alexander S. Szalay



About the Cover: The portrait of Jim Gray was done by illustrator Jeffrey Smith, a professor at the Art Center College of Design, Pasadena, CA. His work has been recognized with awards from the New York Society of Illustrators, the Society of Newspaper Design, The Society of Publications Designers, Communication Arts, among others. For more, see <http://jeffreysmithillustrator.com/>.

Review Articles



- 66 **The Convergence of Social and Technological Networks**
Internet-based data on human interaction connects scientific inquiry like never before.
By Jon Kleinberg

Research Highlights

- 74 **Technical Perspective**
The Polaris Tableau System
By Jim Gray
-
- 75 **Polaris: A System for Query, Analysis, and Visualization of Multidimensional Databases**
By Chris Stolte, Diane Tang, and Pat Hanrahan
-
- 85 **Technical Perspective**
Safeguarding Online Information against Failures and Attacks
By Barbara Liskov
-
- 86 **Zyzyva: Speculative Byzantine Fault Tolerance**
By Ramakrishna Kotla, Allen Clement, Edmund Wong, Lorenzo Alvisi, and Mike Dahlin

Virtual Extension

As with all magazines, page limitations often prevent the publication of articles that might otherwise be included in the print edition.

To ensure timely publication, ACM created *Communications'* Virtual Extension (VE).

VE articles undergo the same rigorous review process as those in the print edition and are accepted for publication on their merit. These articles are now available to ACM members in the Digital Library.

SME Strategies: An Assessment of High vs. Low Performers

Daesoo Kim, Terence T. Ow, and MinJoon Jun

A European Perspective of VoIP in Market Competition

Claudio Feijóo, José Luis Gómez-Barroso, and David Rojo-Alonso

Technology Acceptance and ERP Documentation Usability

Judy E. Scott

Conceptual Modeling Windows and Bounds for Space and Time in Database Constraints

Faiz Currim and Sudha Ram

User-Centric Services Provisioning in Wireless Environments

Quan Z. Sheng, Boualem Benatallah, and Zakaria Maamar

Exploring the Dark Side of IS in Achieving Organizational Agility

Dongback Seo and Ariel I. La Paz

Six Strategies for Electronic Medical Records Systems

Srinivasan Venkatraman, Hillol Bala, Viswanath Venkatesh, and Jack Bates

Technical Opinion

Motivational Affordances: Reasons for ICT Design and Use
Ping Zhang



Association for Computing Machinery
Advancing Computing as a Science & Profession



COMMUNICATIONS OF THE ACM

A monthly publication of ACM Media

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 Membership
Lillian Israel
Director, Office of Publications
Mark Mandelbaum
Director, Office of SIG Services
Donna Cappel

ACM COUNCIL

President
Wendy Hall
Vice-President
Alain Chenais
Secretary/Treasurer
Barbara Ryder
Past President
Stuart I. Feldman
Chair, SGB Board
Alexander Wolf
Co-Chairs, Publications Board
Ronald Boisvert, Holly Rushmeier
Members-at-Large
Carlo Ghezzi;
Anthony Joseph;
Mathai Joseph;
Kelly Lyons;
Bruce Maggs;
Mary Lou Soffa;
SGB Council Representatives
Norman Jouppi;
Robert A. Walker;
Jack Davidson

PUBLICATIONS BOARD

Co-Chairs
Ronald F. Boisvert and Holly Rushmeier
Board Members
Gul Agha; Michel Beaudouin-Lafon;
Jack Davidson; Carol Hutchins;
Ee-ping Lim; M. Tamer Ozsu; Vincent Shen;
Mary Lou Soffa; Ricardo Baeza-Yates

ACM U.S. Public Policy Office
Cameron Wilson, Director
1100 Seventeenth St., NW, Suite 507
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
Chris Stephenson
Executive Director
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (800) 401-1799; F (541) 687-1840

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

STAFF

GROUP PUBLISHER
Scott E. Delman
publisher@cacm.acm.org

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

Art Director
Andrij Borys
Associate Art Director
Alicia Kubista
Assistant Art Director
Mia Angelica Balaquiot
Production Manager
Lynn D'Addesio
Director of Media Sales
Jonathan Just
Advertising Coordinator
Graciela Jacome
Marketing & Communications Manager
Brian Hebert
Public Relations Coordinator
Virginia Gold
Publications Assistant
Emily Eng

Columnists
Alok Aggarwal; Phillip G. Armour;
Martin Campbell-Kelly;
Michael Cusumano; Peter J. Denning;
Shane Greenstein; Mark Guzdial;
Peter Harsha; Leah Hoffmann;
Mari Sako; Pamela Samuelson;
Gene Spafford; Cameron Wilson

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

WEB SITE
http://cacm.acm.org

AUTHOR GUIDELINES
http://cacm.acm.org/guidelines

ADVERTISING

ACM ADVERTISING DEPARTMENT
2 Penn Plaza, Suite 701, New York, NY
10121-0701
T (212) 869-7440
F (212) 869-0481

Director of Media Sales
Jonathan M. Just
jonathan.just@acm.org

Media Kit acmmediasales@acm.org

EDITORIAL BOARD

EDITOR-IN-CHIEF
Moshe Y. Vardi
eic@cacm.acm.org

NEWS
Co-chairs
Marc Najork and Prabhakar Raghavan
Board Members
Brian Bershad; Hsiao-Wuen Hon;
Mei Kobayashi; Rajeev Rastogi;
Jeannette Wing

VIEWPOINTS
Co-chairs
William Aspray;
Susanne E. Hambrusch;
J Strother Moore
Board Members
Stefan Bechtold; Judith Bishop;
Peter van den Besselaar; Soumitra Dutta;
Peter Freeman; Seymour Goodman;
Shane Greenstein; Mark Guzdial;
Richard Heeks; Susan Landau;
Carlos Jose Pereira de Lucena;
Helen Nissenbaum; Beng Chin Ooi

PRACTICE
Chair
Stephen Bourne
Board Members
Eric Allman; Charles Beeler;
David J. Brown; Bryan Cantrill;
Terry Coatta; Mark Compton;
Benjamin Fried; Pat Hanrahan;
Marshall Kirk McKusick;
George Neville-Neil
The Practice section of the CACM
Editorial Board also serves as
the Editorial Board of *ACM Queue*.

CONTRIBUTED ARTICLES
Co-chairs
Al Aho and George Gottlob
Board Members
Yannis Bakos; Gilles Brassard; Peter
Buneman; Andrew Chien; Anja Feldmann;
Blake Ives; Takeo Kanade; James Larus;
Igor Markov; Gail C. Murphy; Shree Nayar;
Lionel M. Ni; Sriram Rajamani; Avi Rubin;
Abigail Sellen; Ron Shamir; Larry Snyder;
Wolfgang Wahlster; Andy Chi-Chih Yao;
Willy Zwaenepoel

RESEARCH HIGHLIGHTS
Co-chairs
David A. Patterson and
Stuart J. Russell
Board Members
Martin Abadi; P. Anandan; Stuart K. Card;
Deborah Estrin; Stuart I. Feldman;
Shafi Goldwasser; Maurice Herlihy;
Norm Jouppi; Andrew B. Kahng; Linda
Petzold; Michael Reiter;
Mendel Rosenblum; Ronitt Rubinfeld;
David Salesin; Lawrence K. Saul;
Guy Steele, Jr.; Gerhard Weikum;
Alexander L. Wolf

WEB
Co-chairs
Marti Hearst and James Landay
Board Members
Jason I. Hong; Jeff Johnson;
Greg Linden; Wendy E. MacKay;
Jian Wang



ACM Copyright Notice
Copyright © 2008 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@acm.org or fax (212) 869-0481.

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

Subscriptions
Annual subscription cost is included in the society member dues of \$99.00 (for students, cost is included in \$42.00 dues); the nonmember annual subscription rate is \$100.00.

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://cacm.acm.org/advertising> or by contacting ACM Media Sales at (212) 626-0654.

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

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

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



Printed in the U.S.A.



DOI:10.1145/1400214.1400215

John R. White

On the 10th Anniversary of ACM's Digital Library

When ACM launched its pioneering Digital Library a decade ago, it was one of the first professional societies to offer its members—and the broader computing community—a digital repository of its publications.

At launch, the ACM Digital Library contained the full text of all articles published by ACM from 1991 forward, and the metadata for articles published back to 1985.

ACM's Digital Library proved a success from the outset. The decision to make the metadata for ACM's digital collection freely available (with subscriptions only required for downloading full text) allowed the computing community worldwide to use and benefit from the Digital Library regardless of their relationship with ACM. That decision, combined with extremely affordable pricing for individuals and institutions to access the full text of ACM articles, set the stage for early, enthusiastic engagement of the Digital Library.

Since its launch, ACM has maintained an ongoing commitment in time, talent, and investment to en-

More than anything else, the Digital Library reflects the dimensions, the brand, the quality—the essence—of ACM.

sure the Digital Library continues to flourish and fulfill the needs of the community. Within a year of its debut (and earlier than other professional societies), ACM resolved to capture and host everything the organization had ever published. We also decided to raise the visibility and importance of *The Guide to Computing Literature*, ACM's bibliographic database of the computing citations from a vast array of global publishers. And we elected to extract references from all publications (electronic and scanned) and treat them as first-class metadata. These decisions were at the core of a major reimplementing of the Digital Library released in 2001. That release included reference linking across all ACM publications, citation counts for ACM articles, and a significantly enhanced *Guide*.

The Digital Library, and the role it plays within the computing community, are top priorities for ACM. Throughout the last decade we have made significant investments in its content, features, performance, and worldwide reach. As a result, the Digital Library is now available at over 2,500 institutions around the globe; 26,000 professional members and 14,000 student members hold individual subscriptions. There are over 2.5 million unique visits per month, one million articles downloaded each month, and 75,000 Digital Library searches conducted each day.

ACM will continue investing in the

Digital Library. New search technology recently integrated has dramatically enhanced Digital Library searches and will enable a much richer, guided navigation experience of ACM (and other publishers') content. A major effort has been completed to normalize author and institution names so the community can easily and accurately find the published work of specific authors and institutions. New bibliometrics are now associated with each article and aggregated for authors (and soon institutions). The new ACM Author Page shows the collected works of an author, institutional affiliations, as well as individual and aggregate citation and download counts. With these new features, users can easily see not only who is publishing, but which articles are actually being downloaded (and presumably read).

The Digital Library has become ACM's most significant product and service. More than anything else, the Digital Library reflects the dimensions, the brand, the quality—the essence—of ACM. The success of the Digital Library, however, doesn't stop here. ACM will continue to invest resources and talent to ensure its Digital Library stays at the leading edge, is accessible and affordable to everyone, and remains the premier digital repository for the computing community.

John R. White

CHIEF EXECUTIVE OFFICER, ACM



CALL FOR PAPERS

23rd International Conference on Supercomputing

June 9-11, 2009

IBM T.J. Watson Research Center
Metro New York City Area, USA



Sponsored by ACM/SIGARCH



GENERAL CO-CHAIRS

Michael Gschwind, IBM TJ Watson
Alex Nicolau, UC Irvine

PROGRAM CO-CHAIRS

Valentina Salapura, IBM TJ Watson
José Moreira, IBM TJ Watson

FINANCE CHAIR

Dimitris Nikolopoulos, Virginia Tech

SUBMISSIONS CHAIR

Alex Ramirez, UPC



ICS is the premier international forum for the presentation of research results in high-performance computing systems. Next year the conference will be held at the IBM T.J. Watson Research Center in Yorktown Heights, NY. The Center is located in a scenic area of State of New York and is within easy reach of New York City.

Papers are solicited on all aspects of research, development, and application of high-performance experimental and commercial systems, including:

- Computationally challenging scientific and commercial applications; Application studies and experiences: porting and tuning for performance and scalability
- Architecture and hardware aspects: high-performance and power-aware microarchitectures, multithreading, multicore and multiprocessor systems optimization to exploit different levels of parallelism, interconnection networks, memory organization and parallel storage and I/O
- High-performance computational and programming models; new languages and middleware for high performance computing; autotuners and function-specific code generators
- Hardware and software aspects of computational accelerators for supercomputing, such as Cell, GPGPUs, and FPGAs
- Software aspects: programming models, restructuring and optimizing compilers and runtime systems, kernel and application development and performance tuning tools, novel infrastructures for Internet and Grid computing, operating systems and autonomic components at all levels
- Performance evaluation studies and theoretical underpinnings of any of the above topics
- Large scale installations in the Petaflop era: design, scaling, power, and reliability, including case studies and experience reports

Papers should not exceed 6,000 words, and should be submitted electronically, in PDF format using the ICS'09 submission web site. *Submissions should be blind.* The review process will include a rebuttal period. Please refer to the ICS09 web site for detailed instructions.

Workshop and tutorial proposals are also be solicited and due by January 15, 2009. For further information and future updates, refer to the ICS'09 web site at <http://www.ics-conference.org> or contact the General or Program Chairs.

Important Dates

Abstract submission: January 12, 2009	Paper submission: January 19, 2009	Author notification: March 23, 2009	Final papers: April 24, 2009
---	--	---	--

For more information, please visit the conference web site at <http://www.ics-conference.org>



DOI:10.1145/1400214.1400216

Diane Crawford

Jim Gray: Humble Visionary

It's been almost two years since Jim Gray set sail off the coast of Northern California headed for the Farallon Islands, never to be seen again. While we can only imagine the impact of his loss to family and close friends,

his absence these long months has also resonated deeply within the computer science community.

"Visionary" and "humble" are words often used in the same breath by friends and colleagues to describe Gray. His contributions to computer science are too numerous to tackle on this page; indeed, even in this issue. He received the ACM A.M. Turing Award in 1998 for his contributions to computer science, particularly his work in database technologies and transaction processing that paved the way for today's global e-commerce markets. In later years, up to the time he went missing (Jan. 28, 2007), he would become deeply fascinated and totally immersed in the world of astronomy, lending his scientific acumen to a new area of research and wonder.

His loss has also been felt intensely in the publishing world. A quick search of the ACM Digital Library finds well over 100 articles and papers published under his byline. His contributions to ACM's Special Interest Group on Management of Data (SIGMOD) are legendary throughout the association. Last May, ACM, along with the IEEE Computer Society and the University of California, Berkeley, hosted a tribute to Gray where dozens of friends, family, and colleagues recalled their fondest memories of this gifted yet unassuming computer scientist, pioneer, teacher, astronomer, mentor, and friend. The June 2008 issue of SIGMOD Record (www.sigmod.org/record) included the full collection of talks and tributes from this event; I encourage

you to read how one person can influence so many lives.

Gray's efforts on behalf of ACM's magazines were equally stellar. He wore many hats throughout his career, having been a scientist, scholar, researcher, and practitioner. He was quick to recognize the need for a publication to address the young practitioner entering the IT field. He was a founding board member of ACM's pioneering *Queue* magazine, created to provide young professionals with the information they would need to stay ahead of the learning curve and understand the technologies that would come into focus over the next 12–18 months.

Gray was also instrumental in the initial discussions to reposition *Communications* as a trusted source for research, as well as of practical and trend-setting editorial content. Indeed, he was an early proponent of presenting computing research in a manner that could be appreciated by both scientists in the field and by a broad-based audience. He understood the editorial interests of academics and practitioners intimately, believing there were ways to satisfy them all within ACM's flagship publication. Just months before his disappearance, he lent his support to *Communications'* Research Highlights section, nominating a paper by Chris Stolte, Diane Tang, and Pat Hanrahan on the Polaris system for query, analysis, and visualization of multidimensional databases. So enthusiastic was Gray about the paper, he wrote the origi-

nal Technical Perspective that would ultimately accompany it here; we are pleased to present both pieces in this issue, beginning on p. 74. We also are grateful to David Patterson, co-chair of the Research Highlights Board, for updating the work of his friend and colleague to reflect changes to the system over the past two years.

Also appearing in this issue, we present two especially memorable presentations from the Berkeley tribute. Fellow pioneers in database research Michael Stonebraker and David DeWitt recall Gray as a distinguished computer scientist, listing his multiple contributions to the field of database systems and his memberships in the National Academy of Sciences, National Academy of Engineering, American Academy of Arts and Sciences, European Academy of Science, as well as a fellow of both the ACM and IEEE (p. 54). And Alexander Szalay, a professor in the Department of Physics and Astronomy at Johns Hopkins University, writes of Jim Gray the astronomer (p. 58), noting how his own collaboration with Gray created some of the world's largest astronomy databases, enabling astronomers to test many avant-garde ideas in practice and see the cosmos in ways never before possible.

We hope you enjoy this collection of memories and work of and about Jim Gray. While the man and scientist is sorely missed, his legacy of work lives on in all of us.

Diane Crawford
EXECUTIVE EDITOR

Even Science Would Benefit from Auctions

IN “DESIGNING THE Perfect Auction” (Aug. 2008), Hal R. Varian noted that such auctions have many practical and obvious applications, including in Web advertising, cooperative robotics, digital business ecosystems, digital preservation, and network management. Auctions, by means of complementary community currencies, can also radically shift the way we conceive scientific cooperation. As we advocated in our paper “Selecting Scientific Papers for Publication via Citation Auctions” (*IEEE Intelligent Systems*, Nov./Dec. 2007), replacing peer review with an auction-based approach would benefit science in general. The better a submitted paper, the more complementary scientific currency its author(s) would likely bid to have it published. If the bid would truly reflect the paper’s quality, the author(s) would be rewarded in this new scientific currency; otherwise, the author(s) would lose the currency.

For all scientists, citations are a form of currency available worldwide, unlike the legal national currencies, which are scarce, especially in the third world. Auctions using citations as currency (“citation auctions”) would encourage scientists to better control the quality of their submissions, since those who are careless risk being dropped from the system. Scientists would also likely be more motivated to prepare worthwhile talks concerning their accepted papers and invite discussion of their results by their peers. Scientists would also likely focus on fewer papers and market them better. Citation auctions could thus greatly improve scientific research, helping it shift from peer review as the reigning selection method toward a continuously improving process of selection based on auctions.

Calculating the value of a work of art or historical document is clearly difficult, and projecting that value into the future is even more difficult. The same holds when trying to calculate the current and possible future value of a scientific work. In a sort of back-to-basics movement, like science in the 18th and

19th centuries, that calculation could now be updated through citation auctions. Peer review would continue, though in a more proper place in the scientific production chain—before selection for publication—rather than as the sole selection step.

This distributed-algorithmic mechanism would provide an interesting theoretical framework for incorporating incentives into algorithmic design, with bidding using an uncertain valuation of a work’s quality, senior scientists helping their younger counterparts enter the scientific system, the marketing of scientific work through recommender systems, the avoidance of citation inflation, the creation of banks of citations, and improved auction mechanisms.

**Josep L. de la Rosa and
Boleslaw K. Szymanski**, Troy, NY

Not Only in the U.S.A.

We all know about the internationalization of computer applications, making them easily translatable into a variety of languages, dialects, and currencies. But what about the internationalization of the editorial content of *Communications*?

The recent redesign (beginning July 2008) prompts me to suggest another change to address something that has been niggling at me for years. *Communications* articles often seem to assume that all readers are in the U.S. An example is the otherwise excellent “Envisioning the Future of Computing Research” by Ed Lazowska (Aug. 2008) in which Lazowska referred to such institutions as “the National Science Foundation” and “the National Academy of Engineering.” A couple of tweaks by an editor would have turned it into “the U.S. National Science Foundation” and “the U.S. National Academy of Engineering,” acknowledging that not all readers think of these bodies as their own national institutions. Lazowska also invited participation in the Computing Community Consortium, which is funded by the U.S. NSF, all of whose

current council members appear to be based in the U.S. It would be useful to know whether the invitation extends to all ACM members or just to those in the U.S.

“Internationalizing” *Communications* content would allow all readers to quickly evaluate its articles for personal relevance—yet another benefit from the magazine’s redesign.

Jamie Andrews, London, Ontario, Canada

Moaning About the Dearth of Native Talent

I must take issue with Eric Roberts’s straw-man argument in his “Counterpoint” in the “Viewpoint” “Technology Curriculum for the Early 21st Century” (July 2008). In the real world, Microsoft might hire a candidate from Bangalore, then wait for more candidates from Bangalore, even while whining that there are no qualified candidates in the U.S.

All companies look to control costs, especially fixed ones, even at the expense of short-term return, since, projecting into the future, the marginal return is less likely to stay positive for more highly compensated employees. The desire to control fixed costs also contributes to demand for consultant positions, as they are eliminated more easily.

I know from personal experience how different reality is from the picture Roberts painted. I have no problem with companies trying to find the lowest-cost qualified labor but am disgusted by disingenuous moaning about the dearth of native talent.

Wayne Warren, San Antonio, TX

A Message Even in Knuth’s Typography

I was introduced to Donald E. Knuth’s masterwork *The Art of Computer Programming* in the late 1980s upon my arrival at college, and while I never fully mastered it, I found it to be a handy tool for accomplishing things that just weren’t possible on the PC-based word

processors of the time.

I was struck by the irony of Knuth's quote "I couldn't stand to see my books so ugly" while spelling the name of his software "TeX" as he reminisced in the concluding part of Edward Feigenbaum's interview with him "Donald Knuth: A Life's Work Interrupted" (Aug. 2008). Microsoft products have evolved to the point where it's now possible to render the correct spelling—"T_EX"—with subscript capital E and condensed, kerned character spacing and still manage to email it intact.

Whenever I stumble across "TeX," I recall being scolded about that spelling in the introduction to its instruction manual.

Michael Pelletier, Merrimack, NH

Include These Programming Voices Too

Though Peter J. Denning's take on programming in his "Profession of IT" "Viewpoint" "Voices of Computing" (Aug. 2008) hit the mark, I'd like to acknowledge the importance of two other roles (voices):

Maintainer. Tries to understand, correct, and improve the "product," though years later may pay dearly for poorly designed coding and a lack of documentation; and

Operations manager. Ensures everyday user service based on the availability of programs—the point of producing programs in the first place.

Requiring programmers to serve some of their apprenticeship as maintainers would help them understand what is important in the conception, design, implementation, and operation of programs. College and university courses that more accurately reflect all aspects of a program's lifespan—from conception to decommissioning—would certainly contribute to their professional development.

Brian Kirk, Painswick, U.K.

How to Know When Important Details Are Omitted

Though the issue explored by Mark Guzdial, in "Paving the Way for Computational Thinking" (Aug. 2008) was important, it carries an equally important caveat. There is no reason to assume, a priori, that every important concept in

computation has a natural counterpart in precomputational thinking; some, indeed, do not.

This theme of making thinking about computation more natural has come up many times and, to my knowledge, always carried a tacit assumption that it can be taught in a way that is natural to newcomers. Guzdial's examples of students' propensity to omit an **else** clause in conditional statements illustrate the point. This is a case of giving ambiguous instructions and assuming the instruction-follower will correctly infer and carry out the appropriate action.

This cannot be fixed by making computers better at guessing how to resolve ambiguous or incomplete instructions. Developing the skill to recognize when important details are omitted and make them explicit is an indispensable part of computational thinking. Moreover, it is largely a new concept to students and thus not easily made natural to them.

There are, of course, aspects of computational thinking that can be made more natural, and doing so is a valuable goal when achievable. But any such attempt must be guided by constant vigilance about what can and what cannot be made natural. Otherwise, the results degenerate into just dumbing-down the material, making it easier, perhaps, but also misleading.

Rodney M. Bates, Wichita, KS

.HK Danger 'Under Control'

We were surprised by the McAfee, Inc. research findings reported in the "News" item "Dangerous Web Domains" (Aug. 2008) and would like to add the following:

Old data. McAfee seemed to be describing the situation in 2008 but collected its data in 2007. While it said that 9.9 million Web sites were tested, most of the malicious ones were tested months before and may no longer exist;

New controls. Since March 2007, the Hong Kong Internet Registration Corporation Limited (HKIRC) has worked closely with the Office of the Telecommunications Authority of the Government of the Hong Kong Special Administrative Region, the Hong Kong Police, and the Hong Kong Computer Emer-

gency Response Team Coordination Centre to monitor and control suspicious Web sites using the .hk domain; and

Less phishing. Beginning in 2007, HKIRC adopted measures against suspicious Web sites. The number of reports of phishing and spamvertising using .hk thus decreased 92%, from an average of 38 per day in 2007 to three per day in 2008 (January to May).

In view of these measures, HKIRC deems the situation under control.

Hong Kong Internet Registration Corporation Limited and Hong Kong Domain Name Registration Company Limited; www.hkirc.hk/

The news item said malicious activity that might be associated with the .hk domain doesn't necessarily take place in Hong Kong or China; "The owner of a domain name could theoretically situate his or her business anywhere." McAfee declined to respond.—Ed.

No Best Way to Build a Mental Model

The six-bullet software design process Robert L. Glass outlined as a trial-and-error activity in his "Practical Programmer" column "Software Design and the Monkey's Brain" (June 2008) is better described as a sophisticated analysis-and-design activity that includes a trial-and-error strategy, given that the purpose of the activity is to analyze a problem and create an automated solution for it.

One root of the less-than-optimal progress in software (and software tools) lies in the column's second bullet item—"Build a mental model of a proposed solution to the problem." Nobody knows the one best way to build a mental model of a software solution. Supporting this conclusion are a large number of software strategies and artifacts, like structured programming, object-oriented programming systems, fourth-generation languages, network/hierarchical/relational DBMSs, FORTRAN, COBOL, C, C++, and Java, some of which endure and some of which simply go extinct.

Alex Simonelis, Montreal

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



DOI:10.1145/1400214.1400218

David Roman

A First Look at the Redesigned Site

Here's a first look at the revised *Communications* Web site. The design is mostly complete, and development is in full swing. These images convey the site's energy and abundance, and suggest the site's goal of being the definitive source of computing news and information.

Paths to Enlightenment

A few things deserve special mention. The site's tabular design will provide pathways to *Communications'* latest content, to archived issues, and to a daily stream of news and blogs. The site will also present articles from *Communications* and other ACM publications organized by topic. Though broad, the topics will give readers a shorter path to material that matches their interests. The site will also present career news and resources. Most of these features will be ready when the site launches early next year.



Calling all Volunteers

As for the blogs, some will be syndicated, but one will be created specifically for the *Communications* site and will be managed by a group of volunteer bloggers. We are looking for bloggers to join this team. Each volunteer should be prepared to share the spotlight with other bloggers and to write several posts a week of interest to the CACM community. We're still working on the details, but volunteers

will likely rotate their roles over time. Please email your suggestions to us at cacm-participation@acm.org.

And More...

The site will need *other* volunteers as well. Interested in beta testing the site? Or in moderating content? Get in touch. Like the bloggers, these volunteers will share different responsibilities over time. The plan—and hope—is to keep the work engaging. Get in touch at cacm-participation@acm.org if you'd like to help.

ACM Member News

SIG AWARD WINNERS

A number of distinguished individuals were recently honored for their contributions to the computing field.

At the 27th annual SIGACT-SIGOPS symposium, Baruch Awerbuch, a professor of computer science at Johns Hopkins University, and David Peleg, a professor of computer science at the Weizmann Institute of Science, won the Edsger W. Dijkstra Prize in Distributed Computing.

At SIGGRAPH, Ken Perlin, a professor of computer science at New York University, won the



Maneesh Agrawala

Computer Graphics Achievement Award; Maneesh Agrawala, an assistant professor of electrical engineering and computer science at the University of California at

Berkeley, won the Significant New Researcher Award; and Stephen Spencer, a graphics system engineer at the University of Washington, won the Outstanding Service Award.

Raghu Ramakrishnan, chief scientist for Audience, won the SIGKDD Innovation Award for "his contributions [that] span foundational technical innovation on algorithmic and systems aspects of data mining."

Alexander L. Wolf, a professor of computer science at Imperial College London, and David S. Rosenblum, a professor of software systems at University College London, won the SIGSOFT Impact Award for their paper "A Design Framework for Internet-Scale Even Observation and Notification."

SIGGRAPH ASIA 2008 CONFERENCE

The first ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia will be held in Singapore from Dec. 10–13. The conference will include an art gallery, computer animation festival, educational courses and programs, emerging technologies, and other innovative content and events. For more information, visit www.siggraph.org/asia2008.

Damage Control

The U.S. patent system is overdue for reform, but what needs fixing, and how, is a matter of some dispute.

IN 2006, FACED with the threat of a court-ordered shutdown, Research In Motion (RIM), the Canadian manufacturer of BlackBerry phones, reached a settlement on a prolonged and vicious patent dispute with Virginia-based NTP. Although NTP's threat of injunction was widely viewed as extortion in the IT industry (the company neither makes nor sells any products, and its primary assets are the patents of the late inventor Thomas Campana), RIM nonetheless agreed to pay \$612.5 million in a "full and final settlement of all claims."

That figure sparked both disbelief and outrage among many members of the IT industry. It also increased calls for substantial legislative reform. Patent law in the U.S. has not been substantially updated since 1952, and is frequently thought to be out of sync with modern business practices. However, exactly what needs reforming, and how, is a matter of some dispute.

Most companies and entrepreneurs agree with the principles that underlie the U.S. patent system, which fosters innovation by granting inventors an exclusive, though temporary, right to their creations in exchange for sharing their work. But what constitutes a patentable invention, and how should

it be protected? Critics complain that American patents are too easy to file—the U.S. Patent and Trademark Office (USPTO) grants tens of thousands of patents each year—and too easy to defend in expensive legal suits. (The NTP patents that RIM was found to have infringed upon might never have been granted in many countries.) Companies whose livelihoods depend on revenue from patent licenses, on the

other hand, are loath to support anything that might weaken the value of their portfolio. The Patent Reform Act of 2007, a reform bill introduced by Senators Patrick Leahy (D-VT) and Orrin Hatch (R-UT) and Representatives Howard Berman (D-CA) and Lamar Smith (R-TX), stalled last April as legislators were unable to reconcile these competing interests.

Trolling for Dollars

It is impossible to make a piece of electronic technology without relying on dozens, if not thousands, of individually patented components. In 2003, one computer hardware firm told the U.S. Federal Trade Commis-



The headquarters of Research In Motion, which are almost shut down due to a patent dispute.

sion that more than 90,000 patents, held by some 10,000 parties, were related to a single microprocessor. Most large IT companies, realizing that their products may regularly infringe on the patents of their rivals, and vice versa, have struck an implicit truce to keep themselves out of court. Increasingly, however, they have come under fire from so-called “non-practicing entities” (known in the IT industry by the less-charitable moniker “patent trolls”), companies whose primary line of business is the litigation of patent lawsuits.

Trolls purchase patents from inventors and other sources—such as bankrupt companies that are selling off their assets—then sue for infringement, hoping to cash in on settlements and royalties. It’s not clear how many patent trolls currently exist, but there’s no doubt they’ve had an impact on the IT industry. In 2005, Yahoo! was engaged in four patent-related lawsuits; by 2008, the number had swollen to 22. All are plaintiff-driven cases, and most of them were filed in the Eastern District of Texas (a jurisdiction widely considered friendly to patent holders), in spite of the fact that Yahoo! has neither offices nor servers nearby. According to Joseph Siino, Yahoo!’s vice president of intellectual property, each lawsuit costs the company an estimated \$2 to \$5 million to defend. Other IT firms report similar figures. Terry Alberstein, director of corporate affairs for Cisco, says that each of the 30 active patent-related suits the company currently faces was brought by a non-practicing entity. The cost of litigation: a staggering \$30 to \$50 million a year.

How to fight this dramatic drain

In 2005, Yahoo! was engaged in four patent-related lawsuits; by 2008, the number had swollen to 22. All are plaintiff-driven cases.

on resources? A cross-industry organization called the Coalition for Patent Fairness, with members such as Dell, Hewlett-Packard, and Intel, was created in 2006 to lobby for litigation reform; a handful of other organizations have taken up the cause, as well. Among the most interesting is the newly formed Allied Security Trust, which seeks to fight patent trolls by acquiring patents and granting member companies nonexclusive rights to use them.

One thing reformers hope to see is a tightening of the standards used to determine patentability. The USPTO is overworked and understaffed, and examiners frequently grant patents to inventions that do not actually merit them, according to some critics. But the real battles happen in court, where a company’s best defense against a claim of infringement is usually to argue that the patent under question is invalid. To be patentable, an invention must be novel, useful, and “non-obvious” to an expert in the field. It is

this last requirement that has proven troublesome: what is obvious? Until recently, the courts held an invention to be obvious only if it was explicitly prefigured by prior “teaching, suggestion, or motivation.” On the other hand, as critics point out, when something is obvious, few people write it down.

The stalled Patent Reform Act tried to address the problem through a controversial post-grant review process that would have made it easier to challenge the validity of a patent. Currently, patents can be challenged through special re-examination proceedings at the USPTO, which are widely seen as cumbersome and ineffectual, or by litigation. The Patent Reform Act would have created a three-judge tribunal whose sole purpose would be to consider patent validity in a less costly and more efficient fashion than in court. Though many companies supported the tribunal’s creation, others worried it would subject patents to potentially endless challenges unless temporal limits were applied to its jurisdiction. Just how many months or years after the granting of a patent the tribunal should be able to rule on a patent’s validity was the subject of fierce arguments.

Despite these legislative battles, the courts have begun to reconsider the issue of obviousness on their own. The U.S. Supreme Court reset the bar last spring when it ruled that the combination of two existing technologies was not “non-obvious” and thus did not deserve a patent. Welcomed by many in the IT industry as an important step toward reform, the decision also granted examiners and courts more discretion to use “common sense” when determining the obviousness of

Virtual Reality

Creating ‘Virtual’ Objects With Ultrasound

A team of Japanese researchers has demonstrated a system that uses ultrasound waves to create “virtual” objects in mid-air, BBC News reports. The system, developed by Takayuki Iwamoto and colleagues from the University of Tokyo, uses ultrasonic transducers, which produce ultrasound. As sound is a

pressure wave, once the inaudible sound waves from the transducers interfere, the waves create a focal point that is perceived as a solid object. A camera tracks the position of a user’s hand and shifts the transducers’ output to move the focus around with the movement of the hand. The result: a feeling of tracing the

virtual object’s surface or edges in mid-air with one’s bare hands.

The system is “the first of its kind,” says Stephen Brewster, a haptics researcher at the University of Glasgow. “You can feel it with both hands, rather than having just a single point of contact, and multiple people can use it at the same time.”

Iwamoto’s team is currently adjusting how the transducers are driven in order to create realistic textures and shapes. The team plans to combine their system with 3D modeling software and video games, and has received proposals from several entertainment companies.

an invention. Yet it was a setback for the pharmaceutical industry, which often seeks new patents for the combination of one drug with another.

Enter Big Pharma

Because drugs typically contain only one or two patented compounds, and the pharmaceutical industry relies far more heavily on patents than any tech company does, the battle over patent reform is frequently depicted as pitting IT against Big Pharma. But the IT industry itself is far from unified. Companies like Qualcomm, Tessera, and Rambus are highly dependent on patent revenue, and are therefore deeply suspicious of the reforms proposed by their peers. Another source of disagreement is the apportionment of patent-related damages, which many IT heavyweights complain have recently ballooned in a manner that's disproportionate to the value of infringed inventions.

"If you can't cure the proliferation of questionable patents, you try to reduce their ramifications," says Robert Barr, executive director of the Berkeley Center for Law & Technology.

An injured patent holder is entitled to pursue several different remedies. Injunctive relief prohibits the defendant from continuing to use or sell the infringed invention. Once common, injunctions have become more difficult to obtain since 2006 when the Supreme Court ruled that they could not automatically be issued to non-practicing entities. Lost profits damages, which are difficult to prove and expensive to analyze, have also fallen out of favor. Most plaintiffs thus opt to seek "reasonable royalties" from the defendant. As a matter of convenience, these royalties are often calculated as a percentage of overall product sales. This angers many in the high tech arena, who claim the calculations don't correspond to the specific value of an infringed patent. Consider the earlier example of a microchip, with its thousands of patented components. If a company were to sue for the infringement of a single component and win, it could ask for damages representing a percentage of the sales of the entire chip.

The Patent Reform Act seeks to redefine "reasonable royalties" to reflect

only the economic value of a patent's "specific contribution over prior art" or, as Senator Leahy described it, "the truly new 'thing' that the patent reflects." It was one of the bill's most hotly contested provisions, drawing criticism from both the pharmaceutical industry and certain IT segments. Their chief complaint: the value of a product may not be separable from the value of an individual component.

"It suggests that the whole is divorced from its parts," asserts Brad Ditty, associate general patent counsel at InterDigital Communications. "And it artificially lowers the value of a patent." Ditty and his peers prefer the flexibility of the current system, and they see no need for reform. Nor do they believe an imbalance exists. "There's this notion that we're currently in the midst of a crisis as far as damage awards are concerned," says Ditty. "We just don't see it."

One proposal that remains uncontroversial with the tech community is the Patent Reform Act's third major provision, which would change the way patents are granted from a first-to-invent to a first-to-file system. (Although some individual inventors have complained that this would put them at a disadvantage relative to larger companies, studies have shown that the first person to file for a patent is almost always the first to invent.) Such a change would bring the U.S. system in line with the rest of the world, and would streamline the approval process by eliminating messy debates about who first had an idea. In fact, it is one of the bill's few provisions that the pharmaceutical industry also supports, and industry insiders regret that IT companies have not been able to use it to greater advantage to score concessions on other points.

At press time, there was no schedule for the Patent Reform Act's return. Senator Leahy has said he remains committed to patent reform, but a growing consensus surmises that supporters of the legislation may need to wait until 2009, when there is a new Congress, a new President, and a new head of the USPTO. In the meantime, the battle will continue to be waged, at great expense, in the courts. **Q**

Leah Hoffmann is a Brooklyn, NY-based science and technology writer.

Quantum Computing

Alexei Kitaev Wins MacArthur "Genius" Award



Alexei Kitaev, a professor of theoretical physics and computer science in the departments of

physics and computer science at the California Institute of Technology, is one of 25 recipients of a MacArthur Foundation \$500,000 "genius" award. As a 2008 MacArthur Fellow, Kitaev will receive \$100,000 a year for five years, with no strings attached.

Kitaev said in a statement that he was "very surprised" when he received a call from MacArthur Fellows Program director Daniel Socolow, informing him of his selection.

"I didn't know what the award was at first," said Kitaev. "But then I looked up the names of people who have previously received a MacArthur award and saw that they are very good scientists. I am excited and honored to be in the same group with them."

A physicist, Kitaev was cited by the MacArthur Foundation for his work in the nature of quantum systems and their implications for creating practical uses, such as quantum computers. "Though his work is focused mainly at the conceptual level, he also participates in efforts to develop working quantum computers," the foundation noted. "Through his deep insights into the fundamental nature of quantum physics, Kitaev reveals a rich picture of this unfamiliar world, bringing us closer to the realization of the full potential of quantum computing."

Kitaev conducted his undergraduate and graduate work in Russia, and came to Caltech as a visiting associate and lecturer in 1998 and was named a professor of theoretical physics and computer science in 2002.

Analyzing Online Social Networks

Social network analysis explains why some sites succeed and others fail, how physical and online social networks differ and are alike, and attempts to predict how they will evolve.

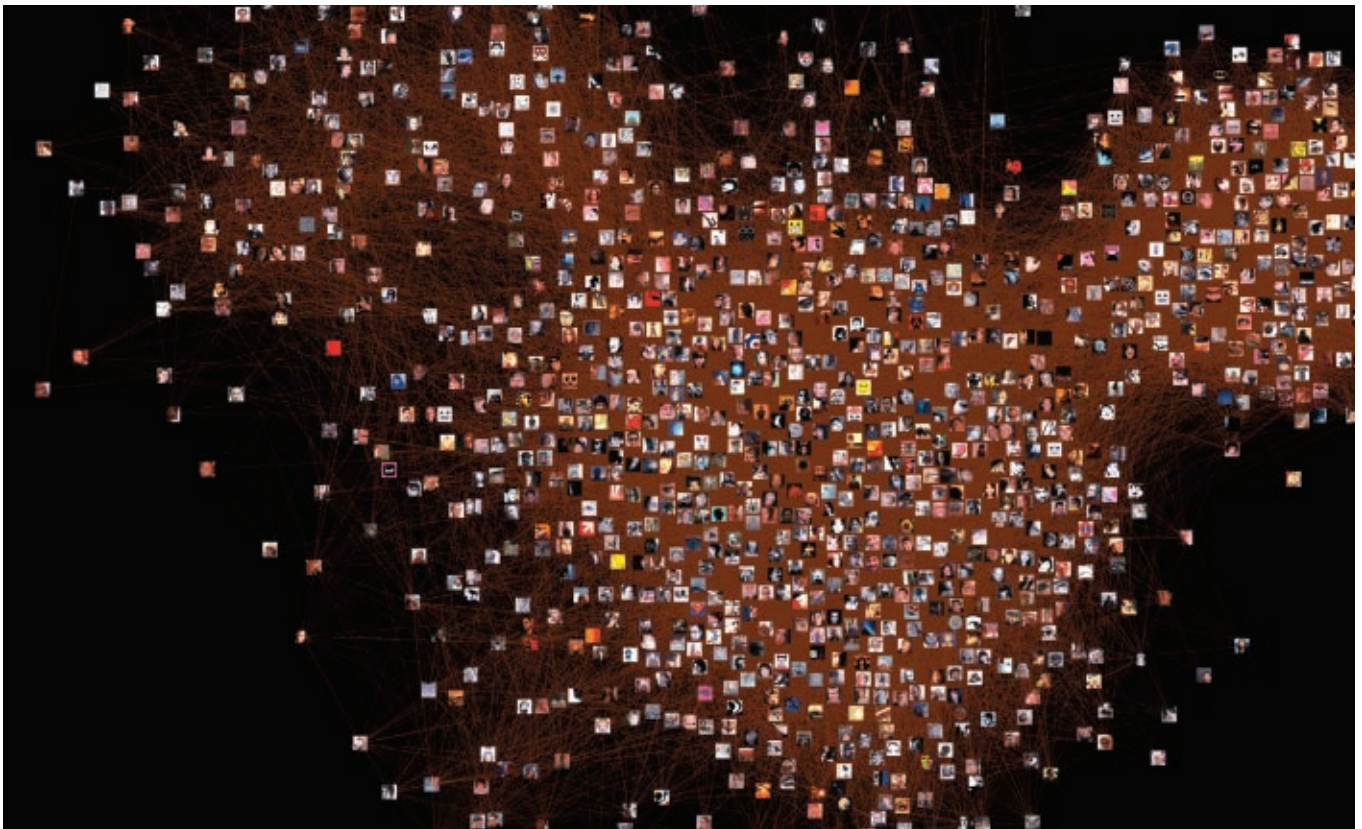
THE ONLINE SOCIAL network seems like a new kid on the online block. Actually, the online social network stretches back years before the dot-com bust. The first major social network site, SixDegrees.com, launched in 1997. The rapid growth has come more recently—MySpace in 2003, Facebook in 2004, and Twitter in 2006—propelled by the ubiquity of broadband and cellular-messaging connections plus the golden touch of yet another Harvard dropout (Mark Zuckerberg of Facebook). Their expansion set off a secondary growth market in analyzing social network sites. Social network analysis (or social networking

analysis, take your pick) helps us understand why Facebook and Flickr succeeded while Friendster didn't; shows how physical and online social networks can be alike and different; and attempts to predict how they'll evolve and, for beneficiaries of the research, how someone might get rich off the next wave. There's also a good deal of research about how honest people are in describing themselves online.

The sites differ in who can join, who can see your profile and how much of it is visible, and their openness to Web crawlers and other applications. The sites also differ in their suitability for use on a cell phone and whether they can be universally accessed among the

multitude of telecom companies. For instance, Twitter, the what-are-you-doing-now site, wouldn't be a big hit if there wasn't a mobile Web.

Online social networks also differ in size. Facebook's magnitude, with 132 million unique visitors in June 2008, seems to fly in the face of the conventional wisdom that too much size makes a social networking site both impersonal and undesirable. (As Yogi Berra quipped, "Nobody goes there anymore; it's too crowded.") More than a few sites evolve in unpredictable ways, sometimes because their infrastructure couldn't handle geometric growth or because their rules annoyed existing members. Some died



A detail from a painting of a Flickr network, consisting only of people with at least 50 mutual contacts, which reveals four distinct clusters.

and others took on second lives. In 2002, Friendster was a dating service, competing against Match.com in the U.S., but it crashed and burned. Now, Friendster has re-emerged as a social network site, but its strongest markets are in Indonesia, Malaysia, the Philippines, and Singapore. Orkut started in the U.S. as a social network site, but flared out; today, 80% of its users reside in Brazil or India.

Social Networking Goes Online

Social network analysis, of course, pre-dates online social networks. Some trace the roots of social network analysis to the early 20th century when sociologist Georg Simmel differentiated between social groups (a group with a specific focus such as a family, neighborhood, or job) and a social network (a looser, larger collection of people and groups with connections among groups). Later, psychologist Abraham Maslow's hierarchy of needs (physiological, safety, love/belonging, esteem, and self-actualization) was used to understand social networks. Research accelerated in the two decades after World War II as the availability of computers allowed the study of social networks with thousands of nodes. It remained for the Internet to provide networks with millions of nodes. As the size of networks grew, it became more difficult to display a network as a plot of dots connected by relationship lines, and the visual description became points or formulas.

Psychologist Stanley Milgram's small world, or six degrees of separation, experiments in the 1960s helped explain some aspects of social networks, including the finding that most pairs of nodes passed through 5.5 nodes to reach the targeted individual. (Don't look for the phrase "six degrees of separation" in Milgram's papers; it was coined by playwright John Guare in his 1992 book of the same name.) While six degrees of separation may be true offline, less than three degrees is more likely online.

The Erdős-Rényi models for generating random graphs, which place connections between pairs of nodes with equal probability, help explain some social networks, but later research indicates that random graph models may not scale to larger online networks.

While six degrees of separation may be true offline, less than three degrees is more likely online.

Work in recent years finds intriguing similarities among social network sites as well as with traditional social networks. In the Barabási-Albert model, networks have power-law, scale-free, growth and exhibit preferential attachment. A physics professor at Notre Dame University, Albert-László Barabási has applied the preferential attachment model to online social networks and found that future gains more often accrue to nodes with more connections. In other words, a rising tide lifts all yachts, oft-cited academic papers are cited even more often, and a newbie to an online community connects more often to a well-known member.

Ravi Kumar, Jasminie Novak, and Andrew Tomkins at Yahoo! Research studied growth patterns at the Flickr photo-sharing site and the Yahoo! 360 social networking site. In both, they found the network density, or the interconnections per person, followed similar patterns: rapid growth through early adopters, decline in the wake of fewer friendships developing relative to network growth, and slow and steady growth where both members and connections grow. The trio segmented the network in three ways: "singletons" who don't take part; a large core of connected users; and a middle region of isolated communities that keep to themselves and display a star structure. The stars make up a third of Flickr users and 10% of Yahoo! 360 users; these communities may have a single charismatic activist linked to other users who have few connections outside the star.

Jure Leskovec of Carnegie Mellon, Lars Backstrom of Cornell, and Ravi Kumar and Andrew Tomkins of Yahoo! Research studied large datasets from

Computer Science

Increasing Network Efficiency

Computer scientists at the University of California, San Diego (UCSD) have developed an algorithm that promises to significantly increase the efficiency of network routing. Known as XL, for approximate link state, the algorithm boosts network routing efficiency by suppressing system updates that force connected networks to continuously recalculate their paths in the Internet.

"Being able to adapt to hardware failures is one of the fundamental characteristics of the Internet," says Kirill Levchenko, a student member of the UCSF team. "Our routing algorithm reduces the overhead of route recomputation after a network change, making it possible to support larger networks. The benefits are especially significant when networks are made up of low-power devices of slow links."

Computer Security

Virus Cinch

Researchers at Tel Aviv University have developed Korset, an open source program designed to halt malware on Linux, the operating system used by the majority of the world's Web and email servers. Instead of waiting for viruses and other malware to begin operating, Korset models the normal behavior of legitimate programs and instantly shuts down any program that veers away from expected activity. Created by Avishai Wool, a professor of computer engineering at Tel Aviv University and his graduate student, Ohad Ben-Cohen, Korset's code has been released at www.korset.org to promote further development of the program. "It is our hope that this becomes mainstream and that this approach is adopted in standing distributions of operating systems," said Wool in an interview with MSNBC.

Flickr, Delicious (social bookmarking), Answers (reference), and LinkedIn (business contacts) to develop a model of network evolution following the preferential attachment model. For all, the number of connections among members drops off exponentially with more degrees of separation, particularly beyond two hops. Two people with a common friend (two hops away) close a triangle and become friends themselves. There were notable differences in new members: Flickr grows exponentially, LinkedIn grows quadratically, Delicious grows superlinearly, and Answers grows sublinearly.

Anthropologist Robin Dunbar has argued a person can sustain about 150 social relationships and that often was the comfortable size of settlements, farming villages, and the tactical unit of the Roman legion, the maniple. Online social networks with millions of users also work to keep human scale in mind.

At Facebook, users strive to mask the immense number of nodes with privacy settings, filters such as People You May Know, and the News Feed that shows on your page what your friends are doing and posting (so you don't have to search dozens or hundreds of individual pages). The News Feed initially set off howls of protest about privacy concerns, but it turned out to be a key element in making Facebook more manageable and fueling its explosive growth. Just as size and density makes cities vibrant and attractive up to a point, Facebook research scientist Jeff Hammerbacher says, "We've noticed that people are more likely to become active users if they enter a dense, active network."

The Facebook network now com-

prises more than 10,000 servers on a Web tier, about 2,000 servers on a MySQL tier, and about 1,000 servers on a MemCache tier. Every second, the site gets 10 million requests, about 500,000 of which are MySQL queries. Data volume was in the tens of gigabytes per day in early 2006, hit 1TB per day by mid-2007, and continues to grow.

"I (almost) look like Brad Pitt"

What man doesn't suck in his gut when a good-looking woman walks by? Online, a user posts his or her best picture, usually in a setting that evokes how the user wants to be perceived, such as placing the Newport Yacht Club or a funky bar in the background. Some users resort to deception. Catalina Toma and Jeffrey Hancock of Cornell University and Nicole Ellison of Michigan State found that when it comes to online profiles on Match.com, Yahoo! Personals, Webdate, and American Singles, 81% of a survey group provided information that deviated from reality. "Deviations tended to be ubiquitous but small in magnitude. Men lied more about their height, and women lied more about their weight, with participants farther from the mean lying more," they noted. "Overall, participants reported being the least accurate about their photographs and the most accurate about their relationship information." The fact that you can update your profile if the misstatement becomes too pronounced may promote deception, although "a record of the presentation is preserved." Because of the asynchronicity of social networking sites, "[Users] can plan, create, and edit their self-presentation, including deceptive elements, much more deliberately than they would in face-to-face first encounters," they noted. "The re-

duction of communication cues, especially nonverbal and visual cues (with the exception of photographs), spares online daters some of the common predicaments faced by traditional daters trying to make a good first impression."

According to Hancock, similar misstatements appear in email communications, too, and they may show similarities in phrasing. "We're looking to see if there are any verbal features that might identify these lies," he says. Which raises the question: Could a future social networking applet be a profile lie detector?

Toma, Hancock, and Ellison found that the online photograph is the information most likely to be less than accurate. The more accurate the photo, the more honest the person is in his or her other profile information. And the more friends who are aware of the online dater's profile, the more accurate the photo. But beware of escalation once the first lie gets told. Hancock says, "There will be elevated lying if people suspect others are, too. Lying will still be constrained even in a 'high-lie environment'—most people do not feel comfortable stating big lies."

Social networks can even make you a fitter, healthier person. Sometimes. Nicole Ellison of Michigan State, Rebecca Heino of Georgetown University, and Jennifer Gibbs of Rutgers University found some respondents to social network and dating sites underreported their weight, then realized they'd better start losing weight to match their ideal self. One woman lost 44 pounds and said, "I can thank online dating for that." Take that, Jenny Craig. ■

Bill Howard writes about science and technology from Westfield, N.J.

Theoretical Astrophysics

The Universe's First Star

For the first time, Japanese and U.S. cosmologists have reliably reproduced the formula of the universe's first star in supercomputer experiments, *Science* reports, and the protostar they produced was the catalyst for a primordial sun that rapidly

expanded to 100 times the mass of our sun.

Led by astrophysicist Naoki Yoshida of Nagoya University and a team of colleagues, the supercomputer simulations of the first primordial stars' formation are partly based on data from NASA's Wilkinson

Microwave Anisotropy Probe. The NASA probe is analyzing the universe's oldest light, which has been traveling across the universe for 13.7 billion years.

Yoshida's team spent nearly eight years on the experiment, and each simulation took a

month of computer time. Even though the theoretical universe exists only as a set of equations operating in a supercomputer, it has provided critical information about the origins of early stars and may help scientists better understand early star formation.

The Limits of Computability

Computational complexity and intractability may help scientists better understand how humans process information and make decisions.

THE THEORY OF modern computing predates by a few years the modern computer itself. In 1936, while studying for his Ph.D. at Princeton University, the British mathematician Alan M. Turing devised an idealized machine that, by executing a series of instructions and storing data, could perform any imaginable computational algorithm. Further work by Turing and others pointed to an intimate connection between computability and calculability in a more conventional sense: If the solution to a problem can be written in the form of a finite mathematical recipe, then it is computable on a Turing machine.

But among computable problems, some (as George Orwell might have put it) are more computable than others. A Turing machine can avail itself of a limitless amount of memory and take as long as it needs to produce an answer. Real computers, on the other hand, have finite storage, and theoretical computability isn't of much practical comfort for an algorithm that takes a long time—longer than the age of the universe, say—to produce an answer. Encryption methods that rely on the difficulty of finding the prime factors of a “key”—a very large number—depend on precisely that point. A guaranteed factorization algorithm is the brute-force strategy of testing all the smaller numbers in sequence until you find an exact divisor, but the size of that task increases exponentially with the number of digits in the key. For large keys, factorization is theoretically feasible but impossible in practice.

Complexity theory arose in the 1960s as a way to classify the hardness of problems. Easy problems belong to complexity class P, for polynomial time, meaning that algorithms exist to produce an answer in a time that rises no faster than some power of the size of the input. Many everyday computational tasks, such as list sorting and optimization by linear programming, belong

to P. A wide variety of more demanding problems do not seem to fall into P, however, but belong to a class labeled NP. The distinguishing characteristic of these problems is that the validity of a proposed solution can be checked in polynomial time, even though there isn't an algorithm for generating that solution in the first place. Factorization falls into this class; if you're presented

degrees of difficulty. The most recalcitrant form a subclass named NP-complete, the salient property of which is that if you could find an efficient (that is, polynomial time) solution to any NP-complete problem, you would in effect have found an efficient solution to all NP problems. That's because there are efficient algorithms that can turn any NP problem into a particular



with two numbers purporting to be the factors of a larger number, it's easy to check the truth of the assertion.

An alternative definition of NP problems is that they can be solved in polynomial time on what's called a nondeterministic Turing machine. This hypothetical machine branches to different computational paths at every step, giving rise to an exponentially growing tree of computations. If one path through that tree arrives at the desired result, the machine has solved the problem. For example, a nondeterministic Turing machine could perform factorization in polynomial time by testing the exponentially large number of possible divisors.

Even within NP, though, there are

instance of an NP-complete problem. Therefore, a general solution to the NP-complete problem automatically includes solutions to NP problems related to it, but a solution to the NP problem solves only some cases of the NP-complete problem.

As his “new favorite example” of an NP-complete problem, Avi Wigderson, a professor in the school of mathematics at the Institute of Advanced Study in Princeton, NJ, offers the familiar Sudoku puzzle. That might come as a surprise to anyone who has solved countless newspaper Sudokus in the time it takes to drink a mug or two of coffee, but Wigderson emphasizes that he is not talking about just the standard 9x9 grids. Sudokus can be constructed on

any doubly square grid—16x16, 25x25, and so on—and the crucial point is that although many Sudokus may be easy to solve, no algorithm exists to solve an arbitrarily large Sudoku in polynomial time.

Showing that a problem is NP-complete means proving that no known algorithm can solve it in polynomial time. But does that mean no such algorithm exists, or that no one has invented one yet? Is the class NP, in other words, truly different from P? So fundamental is this question that it has the distinction of being one of the seven Millennium Problems whose solution the Clay Mathematics Institute in Cambridge, MA, has deemed worthy of a million-dollar prize. The majority opinion among mathematicians and computer scientists is that NP is not identical to P, says Sanjeev Arora, a professor of computer science at Princeton University, which would mean that NP problems are genuinely hard and that in all likelihood no easy method for factorization exists. However, no proof currently exists.

Computational Complexity

Complexity theory has become immensely sophisticated, to the point that there are now close to 500 distinct complexity classes, distinguished by mathematical niceties of the methods that will solve them. This classification is far more than an abstract exercise, however. Understanding a task's computational complexity can provide a novel perspective on real-world processes in which a task shows up.

One "hard" problem not in NP is the computation of so-called Nash equi-

Complexity theory has become sophisticated, to the point there are now close to 500 distinct complexity classes.

libria, which arise in game theory and economics. A Nash equilibrium exists where each participant in a multi-player game is using an optimal strategy, taking into account the strategies used by the other players. That is, no single player can make a unilateral change that will improve his or her outcome. It's recently been proved that the computation of Nash equilibria is complete for its class, known as PPAD, which means that a way to compute them efficiently would also solve other hard problems.

Nash equilibria crop up in the study of distributed decisionmaking, where parts of a system act as independent agents that determine the behavior of the system as a whole. As such, they are of great interest to economists, but their computational intractability raises the question of whether and how real markets actually compute Nash equilibria, or whether the types of Nash equilibria that arise in economics are of a simpler and more tractable kind.

These notions of hardness rest on the assumption that the Turing machine is a universal model for all methods of computation. That seemed like a safe bet until quantum computing came along. The bits in a quantum computer, known as qubits, can exist in superpositions of many states at once, as the strange rules of quantum mechanics allow. The operation on a single qubit can, in effect, perform many computations at once, and in 1994, MIT mathematics professor Peter Shor made use of this property to devise a quantum computer algorithm for factoring numbers in polynomial time.

It remains uncertain whether it is feasible to build quantum computers of sufficient size to do interesting factorizations. Moreover, Wigderson says, a quantum computer is not a nondeterministic Turing machine, because purposeful manipulations of qubits are not equivalent to unrestricted exponential branching. Shor's factorization algorithm, which relies on certain properties of prime numbers in order to select the desired answer from the multiplicity of superposed computations, may be a special case. "There's no hint that quantum computers can solve any NP-complete problem," Wigderson adds.

Beyond its significance for computation in the formal sense, Wigderson says that complexity theory sheds light on our intuitive idea of what it means to call a problem hard. He suggests that the characteristics of a NP problem—a solution cannot be obtained by any routine method, but when found is easily checked—are reminiscent of what happens when a scientist hits on a theoretical innovation that

Information Technology

China Strives for Petaflop Supercomputer

China will significantly increase production of its Godson microprocessor and plans to build its first petaflop supercomputer in 2010, according to *PC World*.

China decided against investing in microprocessor development in the late 1980s, but changed course in 2001. Its chip technology currently lags behind that of AMD, IBM, and

Intel, but China multiplied its investment in chip production in 2006 and now has four different Godson processors. A deal was struck with STMicroelectronics last year to manufacture and sell the chips under the commercial name of Loongson, and the chips are being used by 40 companies in laptops, set-top boxes, and other devices.

Due in 2009, the Godson 3

chip will be China's first multicore chip, with four general-purpose cores and four specialized cores for tasks such as scientific computing, advanced technologies, and national security. China plans to use the Godson 3 chip in its proposed one-petaflop supercomputer. If China succeeds, the Godson 3-powered supercomputer will match the performance of the

IBM Roadrunner system, which leads this year's Top500 list of the world's fastest supercomputers. Built for the U.S. Department of Energy's National Nuclear Security Administration, the Roadrunner system has a processing speed of one quadrillion calculations per second and is used for research into astronomy, climate change, energy, and nuclear weapons simulations.

explains some mystifying experimental fact or when a sculptor suddenly grasps how to complete a work of art. Both experience an “aha!” moment that tells them they’ve found what they want, even if they can’t explain where it came from.

Wigderson regards this analogy as more than a clever metaphor. If the brain is a computer—a fantastically complex computer, but a computer nonetheless—then leaps of intuition, creativity, and aesthetic judgments must ultimately be the results of reasonably efficient computations. Understanding these mysterious processes in algorithmic terms remains a far-off goal, but computer scientists, neuroscientists, and evolutionary biologists are beginning to use the notions of computational complexity and intractability to understand how humans and other animals process information and make decisions.

For example, Adi Livnat, a Miller Postdoctoral Fellow at the University of California at Berkeley, and Nicholas Pippenger, a mathematics professor at Harvey Mudd College, have developed a model that uses a sort of inverse of the Nash equilibrium to define conflict in neural systems trying to deal with warring impulses. An animal that has to endure electric shocks in order to obtain food, for instance, must balance desire for food against fear of physical harm. Regarding these two urges as separate agents within the animal’s brain, conflict arises, Livnat says, because each agent tells the other, “I want you to do something else.”

Evolutionary theorists have argued that the brain ought to develop systems to resolve warring impulses in an orderly manner, yet the fact that humans and other animals can be rendered helpless by indecision suggests this isn’t so. Livnat and Pippenger mathematically demonstrated that conflicting agents in the brain, each independently pursuing its own goal, can nevertheless contrive to produce a beneficial compromise. Crucial to their model was that computing resources were limited: in that case, it turns out that internal conflict can produce an adequate solution where a perfect solution is beyond the system’s computational capacity.

In general, however, understanding neural systems from a computational perspective is a formidable task. As

Luis von Ahn, a professor of computer science at Carnegie Mellon University, points out, many of the tasks that computers can’t do are so easy for humans that we don’t think of them as requiring mental effort—recognizing the letter *A* no matter what style it is written in, for example, or being asked to say whether a picture has a cat in it.

As an empirical way to explore the types of computation that are involved in processes such as image perception, researchers have built neural networks or other adaptable computer systems that can “learn” from a set of training examples. While these efforts can produce improvements in task performance, understanding that improvement in direct algorithmic terms is difficult. Training tends to produce systems that are “completely incomprehensible” in algorithmic terms, says von Ahn. Knowing their structure doesn’t mean you know what they’re doing.

It’s far from clear, adds neuroscientist Larry Abbott of Columbia University, that knowing how computers cope with a task like pattern recognition would necessarily help in understanding how the brain does it. Abbott doesn’t doubt that the human brain works in an essentially algorithmic way, but “the hardware is very different.” Compared to a computer, he says, the brain is “slow and unreliable,” yet it seems far more internally active in that it constantly tries out different ways of processing information rather than sticking to a preordained routine.

Although some philosophers and other researchers disagree, computer scientists and neuroscientists almost universally believe the brain does what does by performing extraordinarily elaborate computations. Millions of years of evolution have trained neural systems in a more or less ad hoc fashion to perform numerous tasks. Neuroscientists are trying to tease out the essential features of these systems by comparing them to computational models and algorithms. The day might come, says Abbott, when it is possible to understand an organism’s behavior in terms of its neural circuitry. **Q**

David Lindley is a science writer and author based in Alexandria, VA. Richard M. Karp and Christos Papadimitriou, both of the University of California at Berkeley, contributed to the development of this article.

Telecommunications

Teens Who Text

For many teenagers, texting is replacing talking on cell phones, according to a new online poll of 2,089 U.S. teenagers conducted by CTIA, the international wireless telecommunications association, and Harris Interactive. Teens say they spend nearly as much time talking as they do texting, and prefer texting as it “is all about multitasking, speed, privacy, and control,” says Joseph Porus, a Harris Interactive vice president. How pervasive is texting? Of the respondents, 42% say they can text blindfolded and 47% say that without texting their social life would deteriorate or simply end.

Teens’ suggestions for future cell phones include phones in the form of sunglasses and jewelry; a “dream device” that is software based and would enable “the user’s fingerprint to turn anything” into a mobile device; and a LEGO-like design so a user can “build just what he wants for every occasion.”

As for landline phones, 40% of the teens say a cell phone is the only phone they will ever need.

Materials Science

A Paper Transistor

Elvira Fortunato, a professor of materials science at the New University of Lisbon, has developed a paper-based transistor that might be suitable for disposable electronics, such as RFID tags and smart labels, *New Scientist* reports. Fortunato built the transistor by covering both sides of a sheet of paper with metal oxides before applying aluminum contacts. The transistor acts as a flexible substrate and as an integral part of a semiconductor by helping to amplify the electrical current that passes through the transistor. The transistors are vulnerable to tears or wetness, but both problems can be overcome by laminating the device.

ACM, Uniting the World's Computing Professionals, Researchers, Educators, and Students

Dear Colleague,

At a time when computing is at the center of the growing demand for technology jobs worldwide, ACM is continuing its work on initiatives to help computing professionals stay competitive in the global community. ACM delivers resources that advance computing as a science and profession.

As a member of ACM, you join nearly 90,000 other computing professionals and students worldwide to define the largest educational, scientific, and professional computing society. Whether you are pursuing scholarly research, building systems and applications, or managing computing projects, ACM offers opportunities to advance your interests.

MEMBER BENEFITS INCLUDE:

- A subscription to the completely redefined **Communications of the ACM**, ACM's flagship monthly magazine
- The option to subscribe to the full **ACM Digital Library**, with improved search functionalities and **Author Profile Pages** for almost every author in computing
- The **Guide to Computing Literature**, with over one million bibliographic citations
- Access to ACM's **Career & Job Center** offering a host of exclusive career-enhancing benefits
- **Free e-mentoring services** provided by MentorNet®
- **Full and unlimited access to over 3,000 online courses** from SkillSoft
- **Full and unlimited access to 1,100 online books**, featuring 500 from Books24x7®, and 600 from Safari® Books Online, including leading publishers such as O'Reilly (Professional Members only)
- The option to connect with the **best thinkers in computing** by joining **34 Special Interest Groups** or **hundreds of local chapters**
- **ACM's 40+ journals and magazines** at special member-only rates
- **TechNews**, ACM's tri-weekly email digest delivering stories on the latest IT news
- **CareerNews**, ACM's bi-monthly email digest providing career-related topics
- **MemberNet**, ACM's e-newsletter, covering ACM people and activities
- **Email forwarding service & filtering service**, providing members with a free acm.org email address and high-quality **Postini spam filtering**
- And much, much more!

ACM's worldwide network ranges from students to seasoned professionals and includes many of the leaders in the field. ACM members get access to this network, and enjoy the advantages that come from sharing in their collective expertise, all of which serves to keep our members at the forefront of the technology world.

I invite you to share the value of ACM membership with your colleagues and peers who are not yet members, and I hope you will encourage them to join and become a part of our global community.

Thank you for your membership in ACM.

Sincerely,



John R. White
Executive Director and Chief Executive Officer
Association for Computing Machinery



Association for
Computing Machinery

Advancing Computing as a Science & Profession



Association for
Computing Machinery

Advancing Computing as a Science & Profession

membership application & digital library order form

Priority Code: ACACM29

You can join ACM in several easy ways:

Online
<http://www.acm.org/join>

Phone
+1-800-342-6626 (US & Canada)
+1-212-626-0500 (Global)

Fax
+1-212-944-1318

Or, complete this application and return with payment via postal mail

Special rates for residents of developing countries:
<http://www.acm.org/membership/L2-3/>

Special rates for members of sister societies:
<http://www.acm.org/membership/dues.html>

Please print clearly

Name _____

Address _____

City _____ State/Province _____ Postal code/Zip _____

Country _____ E-mail address _____

Area code & Daytime phone _____ Fax _____ Member number, if applicable _____

Purposes of ACM

ACM is dedicated to:

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

I agree with the Purposes of ACM:

Signature _____

ACM Code of Ethics:
<http://www.acm.org/serving/ethics.html>

choose one membership option:

PROFESSIONAL MEMBERSHIP:

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

STUDENT MEMBERSHIP:

- ACM Student Membership: \$19 USD
- ACM Student Membership plus the ACM Digital Library: \$42 USD
- ACM Student Membership PLUS Print CACM Magazine: \$42 USD
- ACM Student Membership w/Digital Library PLUS Print CACM Magazine: \$62 USD

All new ACM members will receive an
ACM membership card.
For more information, please visit us at www.acm.org

Professional membership dues include \$40 toward a subscription to *Communications of the ACM*. Member dues, subscriptions, and optional contributions are tax-deductible under certain circumstances. Please consult with your tax advisor.

RETURN COMPLETED APPLICATION TO:

Association for Computing Machinery, Inc.
General Post Office
P.O. Box 30777
New York, NY 10087-0777

Questions? E-mail us at acmhelp@acm.org
Or call +1-800-342-6626 to speak to a live representative

Satisfaction Guaranteed!

payment:

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc. in US dollars or foreign currency at current exchange rate.

- Visa/MasterCard American Express Check/money order
- Professional Member Dues (\$99 or \$198) \$ _____
- ACM Digital Library (\$99) \$ _____
- Student Member Dues (\$19, \$42, or \$62) \$ _____
- Total Amount Due** \$ _____

Card # _____

Expiration date _____

Signature _____



DOI:10.1145/1400214.1400222

Avi Goldfarb and Catherine Tucker

Economic and Business Dimensions

Search Engine Advertising

Examining a profitable side of the long tail of advertising that is not possible under the traditional broadcast advertising model.

THE LONG TAIL is no secret to Internet users or vendors. Both Amazon and Cafepress, for example, have shown that it is possible to profit from providing many niche or unusual products that would take up expensive shelf space offline. Less appreciated is how the long tail alters advertising strategies. In a recent research paper,¹ we explored how, and how much, search engines can profit from providing advertising for niche or unusual popular products. The best way to determine this turns out to be an analysis of Internet advertising purchased by lawyers.

Industry Background

Search platforms enjoy a huge advantage over other online advertising platforms, because they can use a customer's own search terms to match customers' interests with advertisers. Ads are displayed only when a Web user enters a search term, and the advertiser only pays when the user clicks on the ad and is redirected to the advertiser's Web site. Offline companies find this

kind of tailoring much more difficult, so Google can charge higher prices for these ads than for conventional non-tailored ads.

For example, lawyers specializing in computer crime lawsuits can use search engine advertising to ensure their ads are displayed only to people searching for "computer crime lawyer."

Their alternatives would be either: ordinary broadcast-style media like newspapers, magazines, television, and banner ads; or contacting people directly via telephone, postal mail, email, or in person.

Using broadcast media would waste eyeballs: Many of the people who saw the ad would have no interest in find-



ILLUSTRATION BY JOHN HERSEY

ing a computer crime lawyer. Identifying the right people to contact directly is much more expensive than using a search engine. There has been no real competitive response to the new phenomenon of search engine advertising by the broadcast media. All this means that Google and other search engines can offer a uniquely strong value proposition.

Google was not the first company to price and display ads based on search terms or “keywords” and to hold continuous automated auctions to price the advertising of these keywords (that was Goto.com in 1998). However, Google was the first large search engine to use this model, and Yahoo and MSN eventually followed. Google assigns top ad placements to top bidders, but also uses a “quality score” to sort out undesirable advertisers. Besides ads based on keywords and automated auctions, a third unusual feature of search engine advertising is that advertisers pay only when a user clicks on their advertisement. This is not original to Google either—it was used by several different online publishers from 1996 on—but again this is a much more targeted and measurable advertising method than traditional broadcast media advertising where advertisers pay merely to be seen.

Our Methodology

In our research paper, we use data on lawyers’ advertising decisions to illustrate that Google, Yahoo, and other search engines’ ability to capitalize on the long tail of advertising depends on how easy it is to make a match between advertiser and potential customer. This in turn depends on two things: The number of potential customers (fewer customers = more wasted eyeballs) and the difficulty of targeting niche customers using traditional broadcast media or banner ads.

To examine how match difficulty affects advertiser decisions, we use data from a lawyer Web site portal. The data shows the ad prices that lawyers looking to advertise on Google would see for 139 different law-related keywords in April 2007. A typical example of the variance in prices is that one click on an ad in a top spot on Google for “Truck accident attorney” in Boise, ID costs the advertiser about \$12.

There has been no real competitive response to the new phenomenon of search engine advertising from the broadcast media.

In contrast, the same string of words costs about \$32 in Montgomery, AL. “Employment attorney” costs about \$17 in Boise, but just \$5 in Montgomery. We combine this data with information about the popularity of the search terms and about restrictions on lawyer behavior across locations.

We find that search engine advertising generates value through both of the drivers of match difficulty: the number of potential buyers and the difficulty of targeting niche customers. In particular, we show that, controlling for location and search term, firms bid more for keywords when there are fewer customers and therefore a greater need for targeting. This result suggests a possible reason for the price differences between Boise and Montgomery: when there are fewer searches for a string, firms want a way to find these niche customers and they bid the price of that string higher.

Establishing Causation

Correlation does not establish causation. Higher global temperatures are correlated with a lower incidence of piracy on the high seas, but that doesn’t mean that building pirate ships will reduce global warming. For empirical economists, one of their biggest challenges is working out how to establish causation from correlations in the data.

We found a raw correlation in the data between a lack of searches and higher prices, but of itself this does not establish a causal link between match difficulty and bids for search engine ads. To establish a causal link, we exploit a “natural experiment”

that results from the fact that lawyers’ activities are regulated at the state level, by their state’s bar association. These state regulations mean that you can have two similar states, but in one state lawyers are permitted to do something, while in another state they are not.

Specifically, many state bar associations prohibit “ambulance chasing” behavior. In 15 states, lawyers are not permitted to contact clients directly (electronically or in writing) for a period following “personal injury or wrongful death.” This behavior is permitted in the other 35 states. This restriction affects personal injury lawyers, but it does not affect others. When direct solicitation is banned, it is more difficult for personal injury lawyers to target clients. In the absence of search engine advertising, these lawyers could only use broadcast media to find clients. A key point is that the restrictions do not affect the ability of other lawyers to find clients; only personal injury lawyers are affected.

We show that when personal injury lawyers are forbidden from directly contacting clients, they are willing to pay significantly more for search engine advertising, all else equal. By “all else equal,” we mean we control for the fact the keyword string “truck accident attorney” generally costs more than the keyword string “employment attorney” and for the fact that search engine ads in Montgomery cost slightly more than search engine ads in Boise. Our analysis therefore compares whether personal injury keywords in locations with restrictions cost more than other law keywords in those locations, compared to locations without restrictions. If we exclude all other reasonable explanations for the price difference, then we can legitimately attribute the cost difference to the presence or absence of restrictions on lawyer behavior. We will spare you the complicated statistical details—you can read the full paper for those—but we worked out a way we could legitimately connect the restrictions and keyword prices.

We find that personal injury keywords cost approximately \$1 more per click (or 11%) in places where directly contacting clients is prohibited. By making targeting more difficult, the value of search engine advertising in-

ACM Journal on Computing and Cultural Heritage



JOCCH publishes papers of significant and lasting value in all areas relating to the use of ICT in support of Cultural Heritage, seeking to combine the best of computing science with real attention to any aspect of the cultural heritage sector.



www.acm.org/jocch
www.acm.org/subscribe



Association for
Computing Machinery

Search engine advertising is most valuable when firms have just a few hard-to-reach customers.

creases substantially. Interestingly, we find the solicitation restrictions only matter when it is especially difficult to locate clients. These restrictions only affect prices when there are relatively few searches for that keyword in that location. If there are many people searching for that string, and therefore many potential customers, then the restriction on offline targeting does not matter. Search engine advertising is most valuable when firms have just a few hard-to-reach customers. Therefore, it is match difficulty that is driving the relationship between ad prices and the regulation.

Checking our Results

To ensure our results were reliable, we performed a battery of statistical tests. We tried different definitions of personal injury. Also, it could be that the results are driven by the underlying, unobserved condition of commerce of a particular state, and not by people's real advertising choices. For example, the state's personal injury attorneys could be systematically more aggressive at pursuing customer leads than regular attorneys, which could lead both to their valuing leads more and their being regulated more by that state. So, to make sure our result is real, we performed a "falsification test."

We chose a type of law that was similarly motivated, but that would not increase the price paid for ads. This was a type of law that set limits on lawyers taking cases on a contingency fee basis. In states with contingency fee limits, personal injury lawyers paid relatively less for personal injury keywords compared to other legal keywords. So, the falsification test reassures us that there was not something about states

that enact lawyer regulations that can provide an alternative explanation of our results.

What it Means

We show that targeting generates the most value in small markets, where the ability to target using traditional direct response media is limited. We provide clear empirical evidence of the extent to which advertisers value context-based advertising's ability to target very narrow markets. This enables a "long tail of advertising" that is not feasible under the traditional broadcast advertising model. Whether customers are difficult to find because there are few of them or because it is costly to communicate with them, search engine advertising helps firms overcome these challenges and therefore generates considerable value to firms, customers, and (of course) the search engines themselves.

The profitability of search markets for search engines is highly dependent on the availability of alternative marketing communications channels both online and offline. It is therefore not clear that extending electronic auctions to other advertising networks without context-based advertising in place will necessarily be profitable. For example, it is not clear that Google's plans to bring online auctions to TV advertising and conduct these auctions on the basis of "daypart, geography and [...] demographic," will prove as successful as its prior online search auctions that are conducted using specific context-based pricing and extreme micro-targeting.

We have illustrated this process on lawyers because it is convenient to do so, but there is nothing to suggest the results are unique to legal advertising. We believe we will see similar spreads in prices for local market services in which online advertising permits a high level of targeting. ■

Reference

1. Goldfarb, A. and Tucker, C. *Search Engine Advertising: Pricing Ads to Context*. NET Institute Working Paper #07-23, 2008.

Avi Goldfarb (agoldfarb@rotman.utoronto.ca) is an assistant professor of marketing at the Joseph L. Rotman School of Management at the University of Toronto.

Catherine Tucker (cetucker@mit.edu) is the Douglas Drane Career Development Professor in IT and Management and an assistant professor of marketing at the MIT Sloan School of Management.



Privacy and Security A Multidimensional Problem

It's not just science or engineering that will be needed to address security concerns, but law, economics, anthropology, and more.

WHEN THOSE OF US who are now editors of this magazine were in graduate school, it was easy to believe that with the inevitable exception of automation, social implications of computing technology could be ignored with impunity. Yes, before the public Internet, there was discussion of social impact—Joe Weizenbaum's ELIZA, the Department of Health, Education, and Welfare report, *Records, Computers, and the Rights of Citizens*,^a the establishment in Europe and Canada of data commissioners, the "I am a [person]. Please don't bend, fold, spindle or mutilate me," joke that made the rounds in the 1970s,^b the role of computers in President Reagan's Star Wars program—but it was easy to maintain the fiction that the machines we built and the code we wrote had as much social impact as the designers of John Deere tractors have on the migratory patterns of cliff swallows: minimal and not really significant.

Tom Lehrer once sarcastically characterized a famous astronautics engineer, "Once the rockets are up, who cares where they come down? That's



not my department,' says Wernher von Braun."³ But while the view that scientists bear responsibility for the social impact of their work was perhaps radical when it was espoused by Joseph Rotblat (a nuclear physicist who later won a Nobel Peace Prize for his work on nuclear disarmament) in the decade after Hiroshima and Nagasaki, this expectation is no longer unusual. It is also no less true for technologists now than for scientists.

This is part of the ACM code. The original ACM Code of Ethics and Professional Conduct stated, "An ACM member shall consider the health, privacy and general welfare of the public in the performance of the mem-

ber's work." It went on to say that, "An ACM member, when dealing with data concerning individuals, shall always consider the principle of individual privacy and seek the following: To minimize the data collection; To limit authorized access to the data; To provide proper security for the data; To determine the required retention period of the data; To ensure proper disposal of the data." (The current ACM code of ethics contains a similar set of principles, though it omits the requirement regarding proper disposal of data.) But observing current computer privacy and security practices leads one to question whether this code is honored more often in the breach.

Each week brings yet another news story of a major security breach, of the ability to do a cross-site scripting attack on the new favorite mailer, of the polymorphic virus code that changes its signature to evade detection. We aren't getting privacy and security right.

We aren't even asking the right questions. A recent U.S. Department of Defense (DoD) effort to develop an Iraqi ID database of bad guys is one such example. The database includes not just names, but biometric identifiers: fingerprint records and iris scans; its purpose is to maintain records on the people who keep turning up in an area soon after an explosion has occurred.² As any developer knows, of course, this database will not be used only in this way. One such likely use will be at checkpoints—and currently in Iraq, it can be

a This report, which recommended legislation supporting Fair Information Practices for automated data systems, was highly influential in both Europe and the United States; see <http://aspe.hhs.gov/DATACNCL/1973privacy/tocprefacemembers.htm>.

b This was a takeoff on IBM's instructions for the handling of punch cards.

quite dangerous to be a Sunni at a Shiite checkpoint (and vice versa). Now, to its credit, the Defense Science Board, an independent board advising the DoD, recommended that the military “engage responsible advocates of privacy early in the design and application of identity management systems,”¹ yet somehow this database system was developed for use in a place in which a name of the wrong ethnicity can lead to being murdered. Technologists did not stop to consider “once the rockets are up, where will they come down?”

One reason for our failure of cyber privacy and security is that these problems are difficult to resolve. Yes, over 30 years ago we had the ideas of Multics and the Orange Book, but such solutions have little traction in the current environment, especially when (almost) all users seek to mount their newest untrusted device on their (less than fully protected) systems. In the rush toward releasing a product, there is little economic incentive to spend the time properly designing privacy and security into systems.

We don’t ask: What system design for highway toll collection gives appropriate privacy protections? Do we really need to store the toll records any longer than a month after billing? Should we passively collect any data on a user as he or she visits an e-government site? How sensitive is an IP address? (Does it reveal any information about the user?) Is our organization’s system for managing passwords usable? (Or are users finding an insecure workaround?) Is there a way that the digital-rights system can find cheating users without compromising everyone else’s privacy? What are the security risks of that CCTV surveillance system? Can this database system really help us find the bad guys, or does it risk the safety of ordinary citizens? As technologists, we have a responsibility to investigate such issues before we build—not after.

No company wants to appear on the front page of the *New York Times* or in front of the Article 29 Data Protection Working Party of the EU Commission explaining how its system failed to protect important health care/financial/personal data. But while there may be breach laws that require notification in the case of data exposure, there have

Solutions for computer privacy and security are not mathematical theorems, but instead lie in the complexity of human behavior.

been precious few liability suits against the companies whose technologies allowed the problem to occur in the first place. Legal and policy systems simply haven’t kept up with technology. Meanwhile our technology keeps evolving at an ever-increasing pace. Our networked, interconnected systems pose new threats and present new challenges; we need to find new ways of working.

The right technical answers are not always obvious; because the problems involve societal concerns, often the solutions are less than clear-cut. What is the way out of this mess? The sorry state of computer privacy and security is a state for which technologists bear part of the responsibility. We can—and must—be part of the solution. Yet there is another part of this story, namely that computer privacy and security are both technical concerns and social ones.

Solutions for computer privacy and security are not mathematical theorems, but instead lie in the complexities of human behavior. One person’s good identity management scheme may violate another person’s view of adequate control of personal data; another person’s method for securing a network may be simply too restrictive to permit appropriately private access by the network’s users. It is not just science that will enable us to solve these problems, or engineering, or business acumen, or even anthropologic studies of what makes users tick. It will be a combination of all of these, and more.

Communications will publish articles on computer privacy and security in the Practice, Contributed, and Research sections of the magazine. This column will present peoples’ opinions on privacy and security concerns—and their possible solutions. Because the

problems are not only technical, this column will present a diversity of viewpoints on the issues, soliciting responses from lawyers, economists, political scientists—and computer scientists.

We will also seek geographic diversity. The Internet knows no physical boundaries. As we know, its privacy and security breaches don’t either—consider the ILUVU virus that apparently originated in the Philippines, the Nigerian 419 scam^c that can as easily originate in Russia as Nigeria, and a breach in a system designed in Mountain View, CA can cause serious problems in Melbourne, Australia. People are as concerned about data retention in Korea as they are in Europe (and apparently more so than they are in the U.S.). To solve the problems of computer privacy and security, we must look at the issues globally.

Protecting the privacy and security of data in networked computer systems is a major challenge of our age. The challenge of this column is to present ideas that stimulate the critical thinking needed to develop solutions to this multifaceted problem. Yours is to read, ruminate, and change the system—and systems—that currently harbor such poor protections of privacy and security. Change is slow, and changes of this order of magnitude are very difficult. If this column has even a minor impact on improving the privacy and security of computer systems, it will have succeeded in its mission. ■

References

1. Defense Science Board, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics. *Report of the Defense Science Board Task Force on Defense Biometrics*, March 2007, 71.
2. Frank, T. U.S. is building database on Iraqis. *USA Today*, (July 21, 2007); www.usatoday.com/news/world/iraq/2007-07-12-iraq-database_N.htm.
3. Lehrer, T. *Too Many Songs* by Tom Lehrer. Pantheon Books, New York, 1981, 124–125.

Susan Landau (Susan.Landau@sun.com) is a Distinguished Engineer at Sun Microsystems Laboratories in Burlington, MA.

^c This is a scam in which victims are offered large amounts of money from someone who has unexpectedly died (typically in a plane crash) leaving no will or known next of kin. In order to participate, the victims must first demonstrate their seriousness by funding efforts to access the money. It is called a “419” scam after the part of the Nigerian Criminal Code that deals with obtaining property through false pretenses.



Legally Speaking

Quantifying the Value of Patent Exhaustion

Should patents confer power to restrict reuses and redistributions of products embodying the whole or essential parts of inventions?

OWNERS OF PATENTS and copyrights have the right to control the manufacture and sale of products embodying their creations, and that is as it should be. But do they have the right to control all downstream uses, reuses, and transactions as to products embodying their intellectual property (IP)?

Common sense tells us that when we buy a patented machine or a copyrighted book, we have the right to use it for whatever purposes we choose and to resell it. The law backs up this expectation by providing that the first authorized sale of products in the marketplace “exhausts” IP rights in those products.

Quantifying the full value to society of IP exhaustion rules is not easy, but exhaustion clearly makes commerce flow more freely, reduces transaction costs, facilitates follow-on innovation, and promotes competition in primary and secondary markets.

In June 2008 the U.S. Supreme Court in *Quanta Computer, Inc. v. LG Electronics, Inc.* unanimously affirmed the continuing vitality of the exhaustion doctrine, reversing as erroneous a decision by the Court of Appeals for the Federal Circuit (CAFC) that had eroded this rule’s application.

Quanta held (1) that the exhaustion rule applies to method as well as apparatus claims and (2) because LGE’s license with Intel authorized the latter to sell components to customers such as Quanta, exhaustion shielded Quan-

ta from patent liability even though LGE’s license with Intel sought to limit uses that Intel’s customers could make of products purchased from Intel.

Quanta was an important victory for the IT industry, as it thwarts efforts to claim royalties from all downstream users of patented inventions. In this column I will explain LGE’s theory about the exhaustion doctrine and why the Supreme Court rejected LGE’s analysis. I will also consider some implications of *Quanta* for contractual restrictions on uses, reuses, and redistributions of products embodying patented inventions.

The LGE v. Quanta Dispute

Intel and LGE entered into cross-licensing agreements as to their patent portfolios. The LGE license allowed Intel to manufacture and sell microprocessors and chipsets that, when combined with other components, implemented some of LGE’s inventions. LGE’s license allowed Intel to pass on the benefits of the LGE license to customers who purchased Intel-made components, but not to those who mixed Intel and non-Intel components in their systems. LGE’s license obliged Intel to notify its customers about this license limitation. After *Quanta* purchased Intel products and combined them with non-Intel components, LGE sued *Quanta* for patent infringement.

The LGE patents in *Quanta* involved: a method for ensuring that only the most current data would be retrieved

from main memory; a method for coordinating read and write requests within computer systems; and a method for managing data traffic on a bus connecting computer components so that no one device could monopolize the bus.

The microprocessors and chipsets that *Quanta* bought from Intel did not fully embody or practice LGE’s patented inventions for the obvious reason that the methods could not be practiced without combining the Intel products with other components (for example, main memory) capable of carrying out the processes.

LGE claimed that because the Intel products did not and could not embody the patented inventions, the patent exhaustion doctrine did not apply. In its view, patent rights are only exhausted when the patentee has authorized the making and selling of a product that fully embodies the invention.

The CAFC agreed with LGE that the Intel products were only components designed for use in the patented processes, not implementations of those processes. More generally, the CAFC opined that method claims, by their very nature, were not subject to the exhaustion rule. The CAFC thus gave LGE a green light to sue all of Intel’s microprocessor customers for patent infringement unless they purchased Intel-only components for their systems.

Why Quanta Won

Quanta won its appeal to the Supreme Court in large part because history was



ACM
Transactions on
Reconfigurable
Technology and
Systems

ACM Transactions on
Reconfigurable Technology
and Systems

SPECIAL SECTION ON THE 15TH INTERNATIONAL
SYMPOSIUM ON FIELD-PROGRAMMABLE
DEVICES

Articles 1-10 pages Introduction
1-10 pages M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule

Articles 11-20 pages Synthesis of Periodic Logic Networks in FPGA
using Multiple Configurations
11-20 pages M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule

Articles 21-30 pages Statistical Analysis and Process Variation-Aware Routing and
Flow Assignment for FPGA
21-30 pages M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule

Articles 31-40 pages A Design Compiler with a Reconfigurable Processor
31-40 pages M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule, M. J. Heule

Association for
Computing Machinery
Advancing Computing and Science Education

◆ ◆ ◆ ◆ ◆

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

on its side. Indeed, the opening sentence of the *Quanta* decision states: “For over 150 years, this Court has applied the doctrine of patent exhaustion to limit the patent rights that survive the initial authorized sale of a patented item.” The Court reviewed several cases in which exhaustion defenses had succeeded, including the *Univis* case, which, like *Quanta*, involved method claims.

Univis owned patents pertaining to eyeglass lenses. It licensed certain firms to make blank lenses; it also licensed wholesalers to grind the blanks to make finished lenses and retailers to sell the finished lenses to consumers. U.S. anti-trust authorities challenged Univis’ effort to control the downstream market as an unreasonable restraint on trade. In 1942, the Supreme Court ruled that Univis’ patent rights had been exhausted by its authorization of the manufacture and sale of blank lenses. Even though the blanks did not embody the whole of Univis’ invention, they embodied essential features of the invention and the intended use of the blanks was to practice the invention. This sufficed to exhaust Univis’ patent rights.

The Court in *Quanta* observed: “Just as the lens blanks in *Univis* did not fully practice the patents at issue because they had not been ground into finished lenses...the Intel products cannot practice the LGE patents—or indeed function at all—until they are combined with memory and buses in a computer system.” All that stood between the Intel components and completion of LGE’s patented processes was the addition of standard components, such as main memory.

The Court observed that the exhaustion doctrine would be a “dead letter unless it is triggered by the sale of components that essentially, even if not completely, embody an invention.” The “dead letter” danger was especially keen because of how simple it would be for “[p]atentees seeking to avoid the patent exhaustion doctrine [to] simply draft their patent claims to describe a method rather than an apparatus.”

The CAFC’s ruling would allow patentees to intrude deeply into the stream of commerce and upset settled expectations about the consequences of purchasing authorized products from licensed manufacturers. The Supreme

Court could not accept this “end run” around the exhaustion doctrine.

Implications of *Quanta*

Quanta will certainly protect downstream customers from LGE-like attempts to enforce license restrictions as to goods embodying the whole or material parts of patented inventions. However, the Supreme Court in *Quanta* “express[ed] no opinion on whether contract damages might be available [to patent owners] even though exhaustion operates to eliminate patent damages.”

Left unresolved was an important set of questions as to whether (or to what extent) contractual restrictions on customer uses or distributions of products embodying the invention are consistent with patent law’s exhaustion doctrine.

Of particular significance is whether conditional sales of goods embodying patented inventions fall outside the exhaustion rule, as the CAFC opined in *Mallinkrodt, Inc. v. Medipart, Inc.* Mallinkrodt’s patents were for inventions embodied in medical devices. It sold these devices inscribed with a legend that they were “for single use only.” Ignoring this legend, certain customers (hospitals) contracted with Medipart to prepare the devices for reuse. When Mallinkrodt sued Medipart for patent infringement, Medipart asserted that the patent exhaustion doctrine shielded it from liability.

In *Mallinkrodt*, the CAFC treated exhaustion as a default rule that could be, and had been, overridden by sales made conditional on the purchaser’s acceptance of single-use-only terms. (The CAFC has also enforced restric-

Quanta won its appeal to the Supreme Court in large part because history was on its side.

tive legends on bags of seeds that purport to forbid purchasers to plant second-generation seeds derived from purchased seeds.) Failure to abide by such restrictive legends, in the CAFC's view, gives rise not only to liability for breach of contract, but also for patent infringement.

Certain statements in *Quanta* suggest the patent exhaustion ruling in *Mallinkrodt* is wrong. Among other things, *Quanta* states that “[t]he long-standing doctrine of patent exhaustion provides that the initial authorized sale of a patented item terminates all patent rights to that item.” Thus, when Mallinkrodt sold its medical devices to hospitals, its patent rights were exhausted because these were “initial authorized sale[s]” under *Quanta*.

It is, of course, a separate question whether Mallinkrodt could sue the hospitals for breach of a single-use-only term of its sales contracts. Some Supreme Court and other precedents cast doubt on the notion that putting a restrictive legend on a product creates a contractual obligation to abide by the restriction. Yet, some cases have enforced restrictive terms of non-negotiated mass market licenses.

Even so, Medipart should not be liable for breach of contract, as it was not a party to the sale between Mallinkrodt and the hospitals. Contracts only bind those who entered into them, not all others in the world.

License Versus Sale?

It is also a separate question whether exhaustion applies if patentees “license” rather than “sell” their products to customers. Negotiated license restrictions will likely be enforceable as between a patentee and its licensees, but should IP rights be exhausted as to mass-market software just because developers characterize their contractual arrangements with customers as “licenses” rather than as “sales”?

The case law thus far is mixed about how much deference courts should give to such designations. *Vernor v. Autodesk, Inc.* is the most recent U.S. case to consider this issue. The lawsuit arose because Autodesk objected to Vernor's efforts to sell used copies of Autodesk software on eBay. Vernor sought a declaratory judgment that his sale of the software on eBay was


It is a separate question whether exhaustion applies if patentees “license” rather than “sell” their products to customers.

protected by copyright's exhaustion doctrine.

Autodesk moved to dismiss Vernor's lawsuit by arguing that the exhaustion doctrine didn't apply because Autodesk doesn't sell its software to customers, but only licenses it on terms that expressly forbid retransfer of the software. (Remember, there must be “an initial authorized sale” to exhaust IP rights.) Autodesk further claimed that Vernor's sale of the software infringed Autodesk's copyright as an unauthorized distribution of its software to members of the public.

The court in *Vernor* took note of Autodesk's characterization of the transaction as a “license,” but did not consider this designation to be controlling. Instead, it examined the nature of Autodesk's commercial dealings in this software and decided they were more aptly characterized as sales than licenses. Hence, exhaustion protected Vernor from copyright liability. Autodesk has appealed this ruling.

Conclusion

Quanta was an important step in preservation of IP exhaustion rules. The decision provides a sound basis for thwarting other end runs around IP exhaustion rules such as efforts to impose conditional sales and mass market license restrictions on customers. It remains to be seen whether the CAFC and other courts will recognize that the logic of *Quanta* should produce this result. 

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

Calendar of Events

November 13–16

Koli '08: 8th Koli Calling International Conference on Computing Education Research, Finland,
Contact: Arnold N. Pears,
Phone: +46-1847-11066,
Email: arnold.pears@it.uu.se

November 13–16

4th International Conference on Collaborative Computing, Orlando, FL,
Contact: Calton Pu,
Phone: 404-385-1106,
Email: calton.pu@cc.gatech.edu

November 14–15

CHI+IT '08: Computer Human Interaction for the Management of Information Technology, San Diego, CA,
Contact: Eser Kandogan,
Phone: 650-694-7974,
Email: kandogan@cs.umd.edu

November 15–21

SC Conference on High Performance Computing Networking, Storage and Analysis, Austin, TX,
Sponsored: SIGARCH,
Contact: Patricia J Teller PhD,
Phone: 915-747-5939,
Email: pteller@utep.edu

November 16–20

Formal Methods in Computer Aided Design, Portland, OR,
Contact: Robert B Jones,
Phone: 971-214-1770,
Email: robert.b.jones@comcast.net

November 17–19

RISE/EFTS Joint International Workshop on Software Engineering for REsilient SystEms, Newcastle Upon Tyne United Kingdom,
Contact: Alexander Romanovsky
Email: Alexander.romanovsky@ncl.ac.uk

November 17–19

The 4th International Wireless Internet Conference, Maui, HI,
Contact: Jiang Xie,
Phone: 704-687-8413,
Email: jxie1@unc.edu

Join us in San Diego...



Symposium on Computer Human Interaction for Management of Information Technology

November 14/15, 2008 | San Diego, CA

A Turning Point in IT

General Chairs:

Aileen Frisch, Exponential
Eser Kandogan, IBM Research

Program Chairs:

Wayne Lutters, UMBC
Jim Thornton, PARC
Mustapha Mouloua, UCF

Steering Committee:

Stephen Barley, Stanford
David Blank-Edelman,
Northeastern
Jack Carroll, Penn State
Alva Couch, Tufts
Patricia Jones, NASA Ames
Rob Kolstad
Paul Maglio, IBM Research
Tom Sheridan, MIT

Publicity

George Engelbeck, Microsoft
Nate Gunderson, Microsoft

October 13

Advance Registration Ends

November 13

Web Registration Ends

CHIMIT is the leading forum for discussing topics on IT management with a focus on people, business, and technology. Join the discussion on issues, solutions, and research drawing upon fields such as human-computer interaction, human factors, computer systems, and management and service sciences.

Workspace Studies

- Ethnographic studies of IT work in context

Processes and Practices

- Development and use of processes in IT

Organizational Knowledge

- Studies of collaboration and coordination

Plenary Talks

I Got My Jet Pack and I'm Still Not Happy

David Blank-Edelman, Northeastern

Human-Centered Design: Finding the Sweet Spot Among the Many Stakeholders in the Design of a Complex System

William B. Rouse, Tennenbaum Institute, Georgia Institute of Technology

Panels

Designing for Complexity: New Approaches to System Administration UI's
Leading UI architects and designers from IBM, Microsoft, HP, Salesforce.com, Oracle, and BMC.

Design

- Design of human-centered IT systems

Tools and Techniques

- Visualizations of complex system behavior

Automation

- Automation/Policy languages
- Human interfaces to automation



In cooperation with

USENIX

www.chimit08.org

Microsoft



IBM

SIGCHI



Education

Reprogramming College Preparatory Computer Science

The college preparatory computer science education curriculum must be improved, beginning with the earliest phases of the process.

IN EARLY APRIL, the College Board announced the cancellation of the Advanced Placement (AP) Computer Science “AB” course, the more advanced of two AP computing courses that enable students to study college-level content while still in high school. Citing low participation in the “AB” course, the College Board’s communication to AP teachers declared its increased commitment to the Computer Science “A” course, stating, “Appropriate College Board committees will focus their efforts on improving and supporting the AP Computer Science A program, which will be enhanced during the next five years to better represent a full-year, entry-level college computer science sequence.” This attention toward rethinking college preparatory computer science education calls critical attention to the educational crisis in this field.

This announcement should not have come as a surprise to those who have been following computer science education. High school computing courses have shown signs of distress for the past several years. Even for the more popular “A” exam, student participation has declined 15% since the peak enrollment in 2002. Though these participation rates have flattened out over the past two years, the number of exam-takers in AP Computer Science has failed to mirror the increasing number of high school students taking AP exams in other sub-



jects. In fact, since 2002, the average number of students taking AP exams across all subjects has increased by 58%.

Part of the problem of low student enrollment in AP Computer Science can be attributed to the unique challenges teachers encounter in building and sustaining this course. As a former AP Computer Science teacher in a diverse urban high school, I experienced a sense of isolation in teaching a subject with little collegial support and a steep learning curve. As a social science researcher, I have studied the obstacles in creating and maintaining rigorous computer science courses in complex school structures. Since 2004, I have led professional development programs for Los Angeles AP Computer Science teachers and have

encountered numerous challenges to the recruitment and retention of teachers who possess the requisite knowledge to teach this course. Over the past 10 years, I have witnessed the official AP Computer Science programming language change from Pascal to C++ in 1999 and from C++ to Java in 2004. Last year, the case study accounting for up to 25% of the questions on the AP exam changed from the Marine Biology Simulation to the GridWorld Simulation.

For any high school teacher, even those with adequate foundational knowledge and collegial support, keeping up with these modifications is quite a challenge. Few other subjects in the AP program, or in any high school course for that matter, encounter this level of fluctuation that has

ACM Transactions on Accessible Computing



This quarterly publication is a quarterly journal that publishes refereed articles addressing issues of computing as it impacts the lives of people with disabilities. The journal will be of particular interest to SIGACCESS members and delegates to its affiliated conference (i.e., ASSETS), as well as other international accessibility conferences.

www.acm.org/taccess
www.acm.org/subscribe



Association for
Computing Machinery

been demanded from AP Computer Science teachers. As a result, many teachers cannot or will not continue to teach this course and drop off as each adjustment to the course content is revealed. Without course and instructor availability, fewer students have the opportunity to learn any foundational knowledge of computer science.

Another cause of this low participation can be attributed to the low numbers of females and minority students who enroll in the course. Only 17% of exam-takers in the two 2007 AP Computer Science exams were females, representing the lowest rate of female participation in any AP course. Additionally, only a combined 11% of exam-takers were African Americans, American Indians, or Latinos. Clearly, it is difficult to maintain courses that attract such a low representation of the student body.

My line of research has determined that these low participation rates can be attributed to a misunderstanding of the computer science discipline by students, parents, and educators alike; a minimal number of computer science role models who are females or minorities; a representation of the course as difficult and boring; a set of teacher and counselor belief systems that make assumptions about who would do well in this course; a deficiency of student support outside of the classroom; a shortage of qualified teachers; a lack of availability of the course in high-minority and high-poverty schools; and weak and even disengaging pedagogical approaches in the classroom setting. Until we begin addressing these issues, the lack of diversity in computer science courses will continue to impact enrollment and limit the creativity that shapes the computing discipline.

A third cause of this low enrollment in the AP Computer Science program concerns the content of the courses themselves. The “A” course, for example, has focused almost exclusively on object-oriented programming methodology, algorithms, data structures, and abstraction. Though these topics are certainly at the core of many first-year college courses, they are not necessarily the most attractive topics to students who experience more exciting applications of computing in their

recreational and academic domains. The current AP program fails to make explicit the connections between computer science and modern technologies that students are familiar with. Except for the most technologically engaged students, the AP Computer Science course falls short in capturing the excitement of this discipline for 21st century youth. However, the College Board’s aim is to duplicate the most common type of introductory computer science content in higher education. Thus, university faculty must address this issue in their own curriculum to drive changes in the AP course and draw more students into the K–university computing pipeline.

Of course, other non-AP computing courses are regularly offered in high schools, such as Web design, animation, robotics, and desktop publishing. Students show more interest in these courses due to the easier entry points and the ability to integrate their own interests into the course content by designing Web pages, animations, and other creations of their own choosing. However, extensive qualitative research conducted by my colleagues and I demonstrates that the design of these courses typically focuses more on skill development and less on the theoretical underpinnings of computing. Correspondingly, these courses are often located in the vocational education department.

These research findings, detailed in our recently published book, *Stuck in the Shallow End: Education, Race, and Computing*,^a reveal how examples of assignments in these courses include utilizing illustration software to duplicate yellow-page advertisements, creating simple animated characters using drawing programs, and creating static Web pages with basic Web development software layout templates. Rather than learning about the science that underlies the technology, students are directed to become users of preexisting software applications. As a result, these courses rarely qualify as college-preparatory electives, so few college-bound students enroll.

Recently, other promising courses

a J. Margolis et al., *Stuck in the Shallow End: Education, Race, and Computing*. MIT Press, Cambridge, MA, 2008.

have emerged that provide rigorous computing experiences using Alice software, media computation, and other innovative approaches to teaching computer science. Though these courses are very promising, most have not yet become institutionalized as part of district or state college-preparatory curriculum.

In Los Angeles, my colleagues and I, with support from the National Science Foundation (NSF), have spent the past several years committed to broadening the participation in computing among high school students, particularly minorities and female students. Due to the rigorous nature of AP Computer Science, and its status as a college-preparatory course, we originally organized our professional development programs and student outreach efforts around this course. The initial results of our strategy were rapid and dramatic; in three years, the participation of girls tripled, the participation of Latinos quadrupled, and the districtwide overall enrollment in the course doubled. But, due to the ongoing challenges with the AP course discussed previously, we are now changing directions and developing a new college-preparatory computer science course, “Exploring Computer Science.” This effort, also supported by the NSF, presents a more engaging introduction to major computing concepts.

Building upon the curricular topics recommended by the *ACM Model Curriculum for K–12 Computer Science*, “Exploring Computer Science” blends major concepts relating to “Computer Science in the Modern World” (Level II) and “Computer Science as Analysis and Design” (Level III). This course includes units on human-computer interaction, problem-solving methodologies, Web design, programming with Scratch software, data modeling, and robotics. The curriculum adopts an inquiry-based instructional approach and will engage students in unit projects so they can apply their emerging computing knowledge to real-world problems. Diverse representations of computing concepts and computer scientists are integrated throughout the course. In addition, career options that utilize major concepts will be highlighted to address concerns

We must provide students an engaging curriculum that goes beyond programming and represents the imaginative, creative, collaborative, and complex character of computing.


about the nature of the computer science job market.

Rather than attracting only a small subset of students, this course will be integrated into the college-preparatory curriculum across district high schools as an academic elective and will thus enroll a much larger group of students than the AP course. Ongoing professional development for teachers will accompany this new course and formative and summative research will document the strengths and weaknesses of this curriculum in introducing computing topics. We believe this comprehensive approach will help students and teachers understand and appreciate the multiple facets of computing instead of equating computer science solely with computer programming. It is essential to point out that this course development would not have been possible without a strong district/university partnership with Los Angeles Unified School District, particularly the district Director of Secondary Science and his staff.

This new course will also help prepare students who are interested in enrolling in AP Computer Science. I am hopeful the committee selected to redesign the AP Computer Science “A” course will also broaden its conception of what topics and pedagogical approaches should be integrated into the revised course outline in an effort to attract more students, particularly traditionally underrepresented groups of students, to the redesigned course. Maintaining the rigor of AP

Computer Science is important, but the course should also be made relevant, meaningful, and engaging for a diverse body of students. However, as noted earlier, this committee’s ability to reform the content of this AP course is constrained by the results of the College Board’s survey of college and universities first-year course curriculum.

Strengthening college preparatory computer science courses is essential for the health of the computer science discipline at the college level and beyond. The number of newly declared college majors in computer science dropped 44% from 2000 to 2006, likely due to a lack of representative exposure to the field before college and misinformation regarding computing careers. Although the numbers of students studying computer science in high school and college has decreased over several years and only recently flattened out, the Bureau of Labor Statistics lists computer science as the fastest-growing professional sector in the next 10 years. In fact, of the six fastest-growing professions that rely on a college education, five require computer science degrees.

Given the importance of computer science to academic, economic, and security sectors globally, it is imperative that we begin rethinking the computer science educational opportunities provided to students at the beginning of the computing pipeline—in middle school and high school. We must provide students an engaging curriculum that goes beyond programming and represents the imaginative, creative, collaborative, and complex character of computing. This will likely increase overall enrollment, attract more diverse students to the field, and provide a much-needed image makeover to what it means to study computer science. However, this is certainly something that we cannot expect K–12 educators to do on their own. Reprogramming the computer science curriculum will require strong K–12/university partnerships, working the entire pipeline of computer science education. 

Joanna Goode (goodej@uoregon.edu) is an assistant professor in the Department of Teacher Education at the University of Oregon, Eugene, OR.

What does the proliferation of concurrency mean for the software you develop?

BY BRYAN CANTRILL AND JEFF BONWICK

Real-World Concurrency

SOFTWARE PRACTITIONERS TODAY could be forgiven if recent microprocessor developments have given them some trepidation about the future of software. While Moore's Law continues to hold (that is, transistor density continues to double roughly every 18 months), due to both intractable physical limitations and practical engineering considerations, that increasing density is no longer being spent on boosting clock rate, but rather on putting multiple CPU cores on a single CPU die. From the software perspective, this not a revolutionary shift, but rather an evolutionary one: multicore CPUs are not the birthing of a new paradigm, but rather the progression of an old one (multiprocessing) into more widespread deployment. From many recent articles and papers on the subject, however, one might think that this blossoming of concurrency is the coming of the apocalypse that "the free lunch is over."¹⁰

As practitioners who have long been at the coal face of concurrent systems, we hope to inject some calm reality (if not some hard-won wisdom) into a discussion that has too often descended into hysterics. Specifically, we hope to answer the essential question: what does the proliferation of concurrency mean for

the software that you develop? Perhaps regrettably, the answer to that question is neither simple nor universal—your software's relationship to concurrency depends on where it physically executes, where it is in the stack of abstraction, and the business model that surrounds it. And given that many software projects now have components in different layers of the abstraction stack spanning different tiers of the architecture, you may well find that even for the software that you write, you do not have one answer but several: some of your code may be able to be left forever executing in sequential bliss, and some of your code may need to be highly parallel and explicitly multithreaded. Further complicating the answer, we will argue that much of your code will not fall neatly into either category: it will be essentially sequential in nature but will need to be aware of concurrency at some level. While we will assert that less—much less—code needs to be parallel than some might fear, it is nonetheless true that writing parallel code remains something of a black art. We will also therefore give specific implementation techniques for developing a highly parallel system. As such, this article will be somewhat dichotomous: we will try to both argue that most code can (and should) achieve concurrency without explicit parallelism, and at the same time elucidate techniques for those who must write explicitly parallel code. Indeed, this article is half stern lecture on the merits of abstinence and half Kama Sutra.

Some Historical Context

Before discussing concurrency with respect to today's applications, it is helpful to explore the history of concurrent execution: even by the 1960s—when the world was still wet with the morning dew of the computer age—it was becoming clear that a single central processing unit executing a single instruction stream would result in unnecessarily limited system performance. While computer designers experimented with different ideas to circumvent this limi-



ILLUSTRATION BY ANDY GILMORE

tation, it was the introduction of the Burroughs B5000 in 1961 that captured the idea that ultimately proved to be the way forward: disjoint CPUs concurrently executing different instruction streams, but sharing a common memory. In this regard (as in many) the B5000 was at least a decade ahead of its time. But it was not until the 1980s that the need for multiprocessing became clear to a wider body of researchers, who over the course of the decade explored cache coherence protocols (for example, the Xerox Dragon and DEC Firefly), prototyped parallel operating systems (for example, multiprocessor Unix running on the AT&T 3B20A), and developed par-

allel databases (for example, Gamma at the University of Wisconsin).

In the 1990s, the seeds planted by researchers in the 1980s bore the fruit of practical, shipping systems, with many computer companies (for example, Sun, SGI, Sequent, Pyramid) placing big bets on symmetric multiprocessing. These bets on concurrent hardware necessitated corresponding bets on concurrent software: if an operating system cannot execute in parallel, not much else in the system can either. These companies came to the realization (independently) that their operating systems must be rewritten around the notion of concurrent execution. These rewrites took place in

the early 1990s and the resulting systems were polished over the decade. In fact, much of the resulting technology can today be seen in open source operating systems like OpenSolaris, FreeBSD, and Linux.

Just as several computer companies made big bets around multiprocessing, several database vendors made bets around highly parallel relational databases; upstarts like Oracle, Teradata, Tandem, Sybase and Informix needed to use concurrency to achieve a performance advantage over the mainframes that had dominated transaction processing until that time.⁵ As in operating systems, this work was conceived in the

late 1980s and early 1990s, and incrementally improved over the course of the decade.

The upshot of these trends was that by the end of the 1990s, concurrent systems had displaced their uniprocessor forebears as high-performance computers. When the Top500 list of supercomputers was first drawn up in 1993, the highest-performing uniprocessor in the world was just #34, with over 80% of the Top 500 being multiprocessors of one flavor or another. By 1997, uniprocessors were off the list entirely. Beyond the supercomputing world, many transaction-oriented applications scaled with CPU, allowing users to realize the dream of expanding a system without revisiting architecture.

The rise of concurrent systems in the 1990s coincided with another trend: while CPU clock rate continued to increase, the speed of main memory was not keeping up. To cope with this relatively slower memory, microprocessor architects incorporated deeper (and more complicated) pipelines, caches and prediction units. Even then, the clock rates themselves were quickly becoming something of a fib: while the CPU might be able to execute at the advertised rate, only a slim fraction of code could actually achieve (let alone surpass) the rate of one cycle per instruction—most code was mired spending three, four, five (or many more) cycles per instruction. Many saw these two trends—the rise of concurrency and the futility of increasing clock rate—and came to the logical conclusion: instead of spending transistor budget on “faster” CPUs that weren’t actually yielding much in terms of performance gains (and had terrible cost in terms of power, heat, and area), why not take advantage of the rise of concurrent software and use transistors to effect multiple (simpler) cores per die? That it was the success of concurrent software that contributed to the genesis of chip multiprocessing is an incredibly important historical point, and bears reemphasis: there is a perception that microprocessor architects have—out of malice, cowardice, or despair—inflicted concurrency on software.⁷ In reality, the opposite is the case: it was the maturity of concurrent software that led architects to consider concurrency on the die. (Readers are referred to one of the earliest chip multiprocessors—

DEC’s Piranha—for a detailed discussion of this motivation.¹) Were software not ready, these microprocessors would not be commercially viable today. So if anything, the “free lunch” that some decry as being “over” is in fact, at long last, being served—one need only be hungry and know how to eat!

Concurrency is for Performance

The most important conclusion from our foray into the history of concurrency is that concurrency has always been employed for one purpose: to improve the performance of the system. This seems almost too obvious to make explicit. Why else would we want concurrency if not to improve performance? And yet for all its obviousness, concurrency’s *raison d’être* is seemingly forgotten, as if the proliferation of concurrent hardware has awakened an anxiety that all software must use all available physical resources. Just as no programmer felt a moral obligation to eliminate pipeline stalls on a superscaler microprocessor, no software engineer should feel responsible for using concurrency simply because the hardware supports it. Rather, concurrency should be considered and used for one reason only: because it is needed to yield an acceptably performing system.

There are three fundamental ways in which concurrent execution can improve performance: to reduce latency (that is, to make a unit of work execute faster); to hide latency (that is, to allow the system to continue doing work during a long latency operation); or to increase throughput (that is, to make the system able to perform more work).

Using concurrency to reduce latency is highly problem-specific in that it requires a parallel algorithm for the task at hand. For some kinds of problems—especially those found in scientific computing—this is straightforward: work can be divided *a priori*, and multiple compute elements set on the task. But many of these problems are often so parallelizable they do not require the tight coupling of a shared memory—and they are often able to more economically execute on grids of small machines instead of a smaller number of highly concurrent ones. Further, using concurrency to reduce latency requires that a unit of work be long enough in its execution to amortize the substantial

costs of coordinating multiple compute elements: one can envision using concurrency to parallelize a sort of 40 million elements—but a sort of a mere 40 elements is unlikely to take enough compute time to pay the overhead of parallelism. In short, the degree to one can use concurrency to reduce latency depends much more on the problem than those endeavoring to solve it—and many important problems are simply not amenable to it.

For long-running operations that cannot be parallelized, concurrent execution can instead be used to perform useful work while the operation is pending. In this model, the latency of the operation is not reduced, but it is *hidden* by the progression of the system. Using concurrency to hide latency is particularly tempting when the operations themselves are likely to block on entities outside of the program—for example, a disk I/O operation or a DNS lookup. Tempting though it may be, one must be very careful when considering using concurrency to merely hide latency: having a parallel program can become a substantial complexity burden to bear for just improved responsiveness. Further, concurrent execution is *not* the only way to hide system-induced latencies: one can often achieve the same effect by employing non-blocking operations (for example, asynchronous I/O) and an event loop (for example, the `poll()/select()` calls found in Unix) in an otherwise sequential program. Programs that wish to hide latency should therefore consider concurrent execution as an option, not as a foregone conclusion.


When problems resist parallelization or have no appreciable latency to hide, there is a third way to use concurrent execution to improve performance: concurrency may also be used to increase the throughput of the system. That is, instead of using parallel logic to make a single operation faster, one can employ multiple concurrent executions of sequential logic to be able to accommodate more simultaneous work. Importantly, a system using concurrency to increase throughput need *not* consist exclusively (or even largely) of multithreaded code. Rather, those components of the system that share no state can be left entirely sequential, with the system executing multiple instances

of these components concurrently. The sharing in the system can then be offloaded to components explicitly designed around parallel execution on shared state, which can be ideally reduced to those elements already known to operate well in concurrent environments: the database and/or the operating system. To make this concrete, in a typical Model/View/Controller application, the View (typically implemented in environments like JavaScript, PHP, or Flash) and the Controller (typically implemented in environments like J2EE or Ruby on Rails) can consist purely of sequential logic and still achieve high levels of concurrency provided that the Model (typically implemented in terms of a database) allows for parallelism. Given that most don't write their own database (and virtually no one writes their own operating system), it is possible to build (and indeed, many have built) highly concurrent, highly scalable MVC systems without explicitly creating a single thread or acquiring a single lock; it is concurrency by architecture instead of by implementation.


illuminating the Black Art

But what if you are the one developing the operating system or database or some other body of code that must be explicitly parallelized? If you count yourself among the relative few who need to write such code, you presumably do not need to be warned that writing multithreaded code is difficult. In fact, this domain's reputation for difficulty has led some to (mistakenly) conclude that writing multithreaded code is simply impossible: "no one knows how to organize and maintain large systems that rely on locking" reads one recent (and typical) assertion.⁹ Part of the difficulty of writing scalable and correct multithreaded code is the scarcity of written wisdom from experienced practitioners: oral tradition in lieu of formal writing has left the domain shrouded in mystery. So in the spirit of making this domain less mysterious for our fellow practitioners (if not to also demonstrate that some of us actually *do* know how to organize and maintain large lock-based systems), we present some of our collective tricks for writing multithreaded code.

Know your cold paths from your hot paths. If there is one piece of advice to



Tempting though it may be, one must be very careful when considering using concurrency to merely hide latency: having a parallel program can become a substantial complexity burden to bear for just improved responsiveness.



dispense to those that must develop parallel systems, it is to know which paths through your code you want to be able to execute in parallel (the "hot paths") versus which paths can execute sequentially without affecting performance (the "cold paths"). In our experience, much of the software we write is bone-cold in terms of concurrent execution: it is only executed when initializing, in administrative paths, when unloading, and so on. It is not only a waste of time to make such cold paths execute with a high degree of parallelism, it is dangerous: these paths are often among the most difficult and error-prone to parallelize. In cold paths, keep the locking as coarse-grained as possible. Don't hesitate to have one lock that covers a wide range of rare activity in your subsystem. Conversely, in hot paths—in those paths that must execute concurrently to deliver highest throughput—you must be much more careful: locking strategies must be simple and fine-grained, and you must be careful to avoid activity that can become a bottleneck. And what if you don't know if a given body of code will be the hot path in the system? In the absence of data, err on the side of assuming that your code is in a cold path, and adopt a correspondingly coarse-grained locking strategy, but be prepared to be proven wrong by the data.

Intuition is frequently wrong—be data intensive. In our experience, many scalability problems can be attributed to a hot path that the developing engineer originally believed (or hoped) to be a cold path. When cutting new software from whole cloth, some intuition will be required to reason about hot and cold paths. However, once your software is functional in even prototype form, the time for intuition is over: your gut must defer to the data. Gathering data on a concurrent system is a tough problem in its own right. It requires you have a machine that is sufficiently concurrent in its execution to be able to highlight scalability problems. Once you have the physical resources, it requires you put load on the system that resembles the load you expect to see when your system is deployed into production. And once the machine is loaded, you must have the infrastructure to be able to dynamically instrument the system to get to the root of any scalability problems.

The first of these problems has his-

torically been acute: there was a time when multiprocessors were so rare that many software development shops simply didn't have access to one. Fortunately, with the rise of multicore CPUs, this is no longer a problem: there is no longer any excuse for not being able to find at least a two-processor (dual core) machine, and with only a little effort, most will be able (as of this writing) to run their code on an eight-processor (two socket, quad core) machine.

But if the physical situation has improved, the second of these problems—knowing how to put load on the system—has worsened: production deployments have become increas-

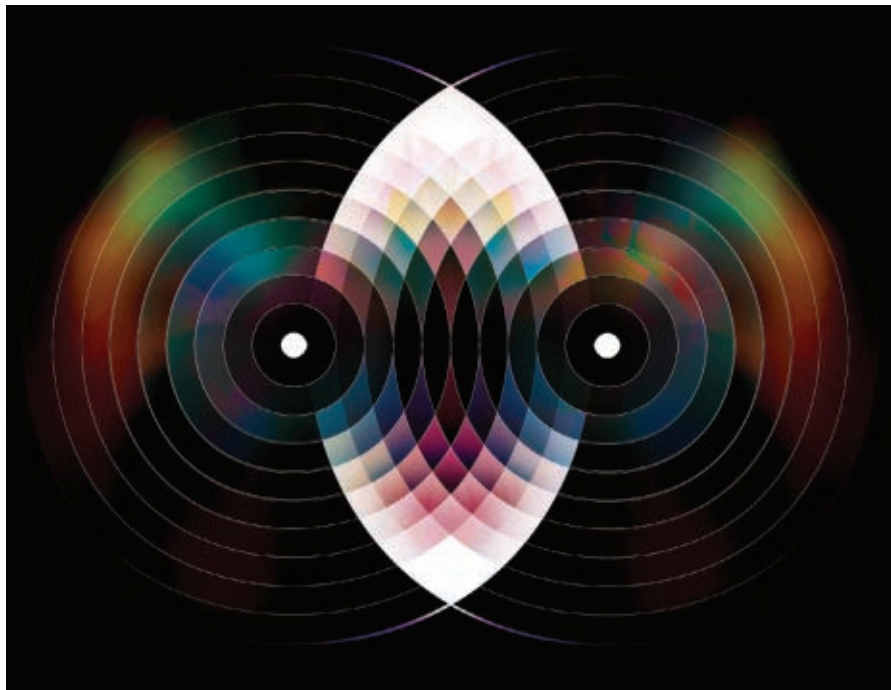
ingly complicated, with loads that are difficult and expensive to simulate in development. It is essential that load generation and simulation be treated as a first-class problem; the earlier this development problem is tackled, the earlier you will be able to get critical data that may have tremendous implications for the software. And while a test load should mimic its production equivalent as closely as possible, timeliness is more important than absolute accuracy: the absence of a perfect load simulation should not prevent you from simulating load altogether, as it is much better to put a multithreaded system under the wrong kind of load than under no load whatsoever.

development or in production—it is useless to software development if the impediments to its scalability can't be understood. Understanding scalability inhibitors on a production system requires the ability to safely dynamically instrument its synchronization primitives, and in developing Solaris, our need for this was so historically acute that it led one of us (Bonwick) to develop a technology to do this (lockstat) in 1997. This tool became instantly essential—we quickly came to wonder how we ever resolved scalability problems without it—and it led the other of us (Cantrill) to further generalize dynamic instrumentation into DTrace, a system for nearly arbitrary dynamic

rearchitecting a subsystem to make it more parallel, we always strive to have the data in hand indicating the subsystem's lack of parallelism is a clear inhibitor to system scalability!

Know when—and when not—to break up a lock. Global locks can naturally become scalability inhibitors, and when gathered data indicates a single hot lock, it is reasonable to want to break up the lock into per-CPU locks, a hash table of locks, per-structure locks, and so on. This might ultimately be the right course of action, but before blindly proceeding down that (complicated) path, carefully examine the work done under the lock: breaking up a lock is not the only way to reduce contention, and contention can be (and often is) more easily reduced by reducing the hold time of the lock. This can be done by algorithmic improvements (many scalability improvements have been had by reducing execution under the lock from quadratic time to linear time!) or by finding activity that is needlessly protected by the lock. As a classic example of this latter case: if data indicates that you are spending time (say) deallocating elements from a shared data structure, you could dequeue and gather the data that needs to be freed with the lock held, and defer the actual deallocation of the data until after the lock is dropped. Because the data has been removed from the shared data structure under the lock, there is no data race (other threads see the removal of the data as atomic), and lock hold time has been reduced with only a modest increase in implementation complexity.

Be wary of readers/writer locks. If there is a novice error when trying to break up a lock, it is this: seeing that a data structure is frequently accessed for reads and infrequently accessed for writes, it can be tempting to replace a mutex guarding the structure with a readers/writer lock to allow for concurrent readers. This seems reasonable, but unless the hold time for the lock is long, this solution will scale no better (and indeed, may well scale worse) than having a single lock. Why? Because the state associated with the readers/writer lock must itself be updated atomically, and in the absence of a more sophisticated (and less space-efficient) synchronization primitive, a readers/writer lock will use a single word of memory to store the num-



instrumentation of production systems that first shipped in Solaris in 2004, and has since been ported to many other systems including FreeBSD and MacOS.³ (The instrumentation methodology in lockstat has been reimplemented to be a DTrace provider, and the tool itself has been reimplemented to be a DTrace consumer.)

Today, dynamic instrumentation continues to provide us with the data we need to not only to find those parts of the system that are inhibiting scalability, but to gather sufficient data to understand which techniques will be best suited to reduce that contention. Prototyping new locking strategies is expensive, and one's intuition is frequently wrong; before breaking up a lock or

rearchitecting a subsystem to make it more parallel, we always strive to have the data in hand indicating the subsystem's lack of parallelism is a clear inhibitor to system scalability!

rearchitecting a subsystem to make it more parallel, we always strive to have the data in hand indicating the subsystem's lack of parallelism is a clear inhibitor to system scalability!

ber of readers. Because the number of readers must be updated atomically, acquiring the lock as a reader requires the same bus transaction—a read-to-own—as acquiring a mutex, and contention on that line can hurt every bit as much. There are still many situations where long hold times (for example, performing I/O under a lock as reader) more than pay for any memory contention, but one should be sure to gather data to make sure that it is having the desired effect on scalability. And even in those situations where a readers/writer lock is appropriate, an additional note of caution is warranted around blocking semantics. If, for example, the lock implementation blocks new readers when a writer is blocker (a common paradigm to avoid writer starvation), *one cannot recursively acquire a lock as reader*: if a writer blocks between the initial acquisition as reader and the recursive acquisition as reader, deadlock will result when the recursive acquisition is blocked. None of this is to say that readers/writer locks shouldn't be used, just that they shouldn't be romanticized.

Know when to broadcast—and when to signal. Virtually all condition variable implementations allow threads waiting on the variable to be awoken either via a signal (in which case one thread sleeping on the variable is awoken) or via a broadcast (in which case all threads sleeping on the variable are awoken). These constructs have subtly different semantics: because a broadcast will awaken all waiting threads, it should generally be used to indicate *state change* rather than *resource availability*. If a condition broadcast is used when a condition signal would have been more appropriate, the result will be a *thundering herd*: all waiting threads will wake up, fight over the lock protecting the condition variable and (assuming that the first thread to acquire the lock also consumes the available resource) sleep once again when they discover that the resource has been consumed. This needless scheduling and locking activity can have a serious effect on performance, especially in Java-based systems, where `notifyAll()` (that is, broadcast) seems to have entrenched itself as a preferred paradigm; changing these calls to `notify()` (or `signal`) has been known to result in substantial performance gains.⁶

Design your systems to be composable. Among the more galling claims of the detractors of lock-based systems is the notion that they are somehow uncomposable: “Locks and condition variables do not support modular programming,” reads one typically brazen claim, “building large programs by gluing together smaller programs: locks make this impossible.”⁸ The claim, of course, is incorrect; for evidence one need only point at the composition of lock-based systems like databases and operating systems into larger systems that remain entirely unaware of lower-level locking.

There are two ways to make lock-based systems completely composable, and each has its own place. First (and most obviously) one can make locking entirely internal to the subsystem. For example, in concurrent operating systems, control never returns to user-level with in-kernel locks held; the locks used to implement the system itself are entirely behind the system call interface that constitutes the interface to the system. More generally, this model can work whenever there is a crisp interface between software components: as long as control flow is never returned to the caller with locks held, the subsystem will remain composable.

Secondly, (and perhaps counterintuitively), one can achieve concurrency and composability by having no locks whatsoever. In this case, there must be no global subsystem state; all subsystem state must be captured in per-instance state, and it must be up to consumers of the subsystem to assure that they do not access their instance in parallel. By leaving locking up to the client of the subsystem, the subsystem itself can be used concurrently by different subsystems and in different contexts. A concrete example of this is the AVL tree implementation that is used extensively in the Solaris kernel. As with any balanced binary tree, the implementation is sufficiently complex to merit componentization, but by not having any global state, the implementation may be used concurrently by disjoint subsystems—the only constraint is that manipulation of a single AVL tree instance must be serialized.

The Concurrency Buffet

It's difficult to communicate over a decade of accumulated wisdom in a single article, and space does not permit us ex-

ploration of the more arcane (albeit important) techniques we have used to deliver concurrent software that are both high-performing and reliable. Despite our attempt to elucidate some of the important lessons that we have learned over the years, concurrent software remains, in a word, difficult. Some have become fixated on this difficulty, viewing the coming of multicore computing as cataclysmic for software. This fear is unfounded, for it ignores the fact that relatively few software engineers need to actually write multithreaded code: for most, concurrency can be achieved by standing on the shoulders of those subsystems that are highly parallel in implementation. Those practitioners implementing a database or an operating system or a virtual machine will continue to sweat the details of writing multithreaded code, but for everyone else, the challenge is not how to implement those components but rather how to best use them to deliver a scalable system. And while lunch might not be exactly free, it *is* practically all-you-can-eat. The buffet is open. Enjoy! □

References

1. Barroso, L. A., Gharachorloo, K., McNamara, R., Nowatzky, A., Qadeer, S., Sano, B., Smith, S., Stets, R., and Verghese, B. Piranha: A scalable architecture based on single-chip multiprocessing. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*. ACM, NY, 282–293, 2000.
2. Cantrill, B. Postmortem object type identification. In *Proceedings of the 5th International Workshop on Automated Debugging*, 2003.
3. Cantrill, B. Hidden in plain sight. *Queue* 4, 1 (Feb. 2006), 26–36.
4. Cantrill, B. A spoonful of sewage. A. Oram and G. Wilson, Eds. *Beautiful Code*. O'Reilly, 2007.
5. DeWitt, D. and Gray, J. Parallel database systems: The future of high performance database systems. *Commun. ACM* 35, 6 (June 1992), 85–98.
6. McKusick, K. A conversation with Jarod Jenson. *Queue* 4, 1 (Feb. 2006), 16–24.
7. Oskin, M. The revolution inside the box. *Commun. ACM* 51, 7 (July 2008), 70–78.
8. Peyton-Jones, S. Beautiful concurrency. A. Oram and G. Wilson, Eds. *Beautiful Code*. O'Reilly, 2007.
9. Shavit, N. Transactions are tomorrow's loads and stores. *Commun. ACM* 51, 8 (Aug. 2008), 90–90.
10. Sutter, H. and Larus, J. Software and the concurrency revolution. *Queue* 3, 7 (Sept. 2005), 54–62.

Bryan Cantrill is a Distinguished Engineer at Sun Microsystems, where he works on concurrent systems. Along with colleagues Mike Shapiro and Adam Leventhal, Bryan developed DTrace, a facility for dynamic instrumentation of production systems that was directly inspired by Bryan's frustration in understanding the behavior of concurrent systems.

Jeff Bonwick is a Fellow at Sun Microsystems, where he works on concurrent systems. Best known for inventing and leading the development of Sun's Zettabyte Filesystem (ZFS), Bonwick has also written (or rather, rewritten) many of the most parallel subsystems in the Solaris kernel, including the synchronization primitives, kernel memory allocator, and thread-blocking mechanism.

The promise of STM may likely be undermined by its overheads and workload applicabilities.

BY CĂLIN CAȘCAVAL, COLIN BLUNDELL, MAGED MICHAEL, HAROLD W. CAIN, PENG WU, STEFANIE CHIRAS, AND SIDDHARTHA CHATTERJEE

Software Transactional Memory: Why is it Only a Research Toy?

TRANSACTIONAL MEMORY (TM)¹³ is a concurrency control paradigm that provides atomic and isolated execution for regions of code. TM is considered by many researchers to be one of the most promising solutions to address the problem of programming multicore processors. Its most appealing feature is that most programmers only need to reason locally about shared data accesses, mark the code region to be executed transactionally, and let the underlying system ensure the correct concurrent execution. This model promises to provide the scalability of fine-grained locking while avoiding common pitfalls of lock composition such as deadlock. In this article, we explore the performance of a highly optimized STM

and observe the overall performance of TM is much worse at low levels of parallelism, which is likely to limit the adoption of this programming paradigm.

Different implementations of transactional memory systems make tradeoffs that impact both performance and programmability. Larus and Rajwar¹⁶ present an overview of design trade-offs for implementations of transactional memory systems. We summarize some of the design choices here:

- ▶ Software-only (STM)^{7, 10, 12, 14, 18, 23, 25} is the focus here. While offering flexibility and no hardware cost, it leads to overhead in excess of most users' tolerance.

- ▶ Hardware-only (HTM)^{2, 4, 9, 13, 19, 20, 35} suffers from two major impediments: high implementation and verification costs lead to design risks too large to justify on a niche programming model; hardware capacity constraints lead to significant performance degradation when overflow occurs, and proposals for managing overflows (for example, signatures⁵) incur false positives that add complexity to the programming model. Therefore, from an industrial perspective, HTM designs have to provide more benefits for the cost, on a more diverse set of workloads (with varying transactional characteristics) for hardware designers to consider implementation.^a

- ▶ Hybrid^{1, 6, 24, 28} is the most likely platform for the eventual adoption of TM by a wide audience, although the exact mix of hardware and software support remains unclear.

A special case of the hybrid systems are hardware-accelerated STMs. In this scenario, the transactional semantics are provided by the STM, and hardware primitives are only used to speed up critical performance bottlenecks in the STM. Such systems could offer an attractive solution if the cost of hardware primitives is modest and may be further amortized by other uses in the system.

Independent of these implementa-

^a Reuse of hardware for other purposes can also justify its inclusion, as the case may be for Sun's implementation of Scout Threading in the Rock processor.³²

Figure 1: STM operations.

```
STM_BEGIN()
read global version number /* gv# */
```

(a) Pseudo-code for STM begin

```
STM_VALIDATE()
read global version number /* gv# */
if global version number changed /* gv# */
for each read set entry
if metadata changed return FALSE
return TRUE
```

(b) Pseudo-code for STM validate

```
STM_READ(A)
if already written goto written path
read metadata of A
if metadata is locked goto conflict path
log A and its metadata in the read set
read value at A
if ! STM_VALIDATE() goto conflict path
return val
```

(c) Pseudo-code for STM read barrier

```
STM_END()
lock metadata for write set
if already locked goto conflict path
if ! STM_VALIDATE() goto conflict path
/* Success guaranteed */
increment global version number /* gv# */
execute writes
update/unlock metadata for write set
```

(d) Pseudo-code for STM end

tion decisions, there are transactional semantics issues that break the ideal transactional programming model for which the community had hoped. TM introduces a variety of programming issues that are not present in lock-based mutual exclusion. For example, semantics are muddled by:

- Interaction with non-transactional codes, including access to shared data from outside of a transaction (tolerating weak atomicity) and the use of locks inside a transaction (breaking isolation to make locking operations visible outside transactions);

- Exceptions and serializability: how to handle exceptions and propagate consistent exception information from within a transactional context, and how to guarantee that transactional execution respects a correct ordering of operations;

- Interaction with code that cannot be transactionalized, due to either communication with other threads or a requirement barring speculation;

- Livelock, or the system guarantee that all transactions make progress even in the presence of conflicts.

In addition to the intrinsic semantic issues, there are also implementation-specific optimizations motivated by high transactional overheads, such as programmer annotations for exclud-

ing private data. Furthermore, the non-determinism introduced by aborting transactions complicates debugging—transactional code may be executed and aborted on conflicts, which makes it difficult for the programmer to find deterministic paths with repeatable behavior. Both of these dilute the productivity argument for transactions, especially software-only TM implementations.

Given all these issues, we conclude that TM has not yet matured to the point where it presents a compelling value proposition that will trigger its widespread adoption. While TM can be a useful tool in the parallel programmer's portfolio, it is our view that it is not going to solve the parallel programming dilemma by itself. There is evidence that it helps with building certain concurrent data structures, such as hash tables and binary trees. In addition, there are anecdotal claims that it helps with workloads; however, despite several years of active research and publication in the area, we are disappointed to find no mentions in the research literature of large-scale applications that make use of TM. The STAMP³⁰ and Lonestar¹⁷ benchmark suites are promising starts, but have a long way to go to be representative of full applications.

We base these conclusions on our work over the past two years building a

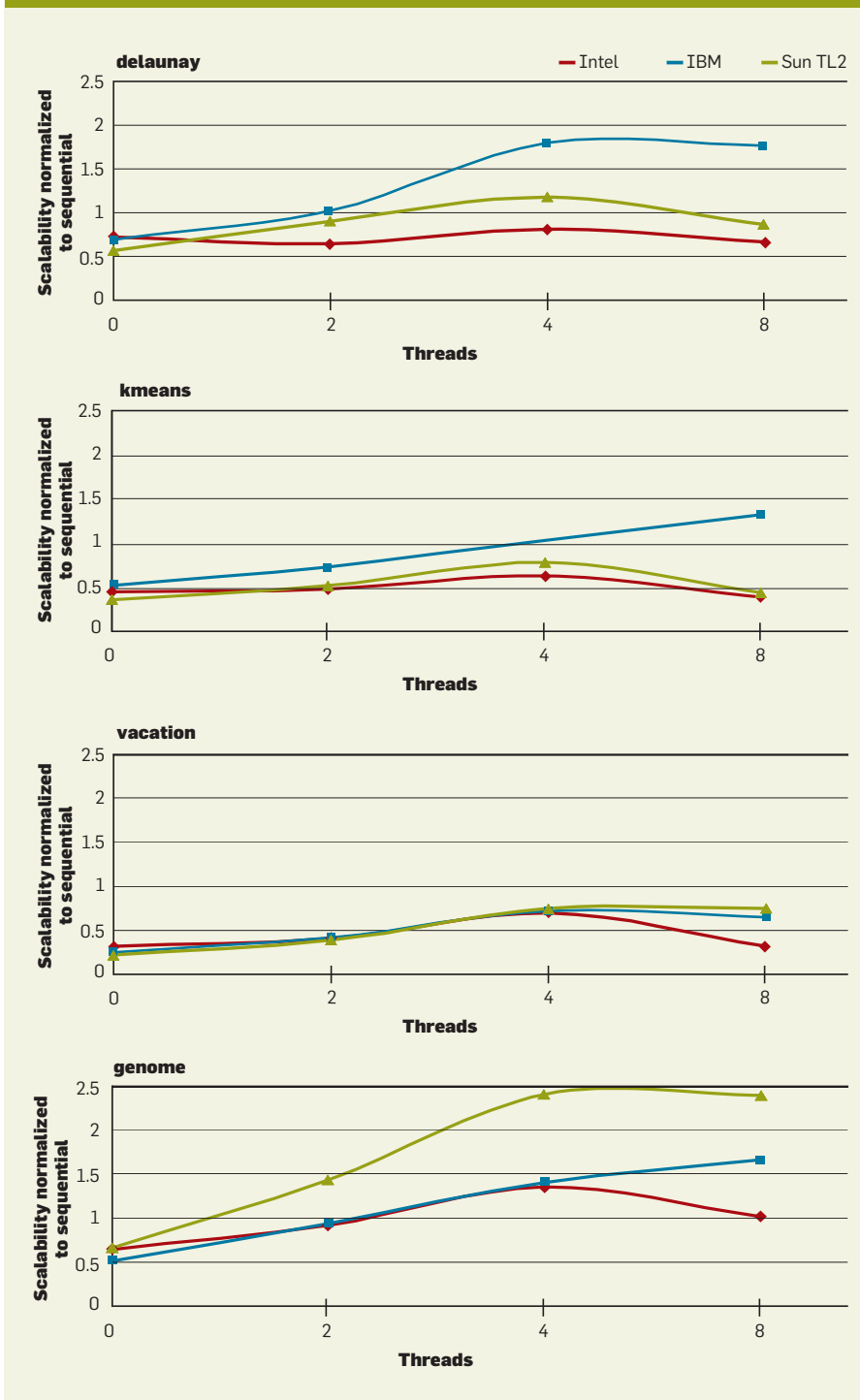
state of the art STM runtime system and compiler framework, the freely available IBM STM.³¹ Here, we describe this experience, starting with a discussion of STM algorithms and design decisions. We then compare the performance of this STM with two other state of the art implementations (the Intel STM¹⁴ and the Sun TL2 STM⁷) as well as dissect the operations executed by the IBM STM and provide a detailed analysis of the performance hotspots of the STM.

Software Transactional Memory

STM implements all the transactional semantics in software. That includes conflict detection, guaranteeing the consistency of transactional reads, preservation of atomicity and isolation (preventing other threads from observing speculative writes before the transaction succeeds), and conflict resolution (transaction arbitration). The pseudo-code for the main operations executed by a typical STM is illustrated in Figure 1. We show two STM algorithms, one that performs full validation and one that uses a global version number (the additional statements marked with the *gv#* comment).

The advantage of an STM for system programmers is that it offers flexibility in implementing different mechanisms and policies for these operations. For

Figure 2: Scalability results for three STM runtimes on a quad-core Intel Xeon server: IBM, Intel STM v2, and Sun TL2.



end users, the advantage of an STM is that it offers an environment to transactionalize (that is, porting to TM) their applications without incurring extra hardware cost or waiting for such hardware to be developed.

Conversely, an STM entails nontrivial drawbacks with respect to performance and programming semantics:

► **Overheads:** In general, STM results

in higher sequential overheads than traditional shared-memory programming or HTM. This is the result of the software expansion of loads and stores to shared mutable locations inside transactions to tens of additional instructions that constitute the STM implementation (for example, the STM_READ code in Figure 1c). Depending on the transactional characteristics of a workload,

these overheads can become a high hurdle for STM to achieve performance. The sequential overheads (that is, conflict-free overheads that are incurred regardless of the actions of other concurrent threads) must be overcome by the concurrency-enabling characteristics of transactional memory.

► **Semantics:** In order to avoid incurring high STM overheads, non-transactional accesses (such as loads and stores occurring outside transactions) are typically not expanded. This has the effect of weakening—and hence complicating—the semantics of transactions, which may require the programmer to be more careful than when strong transactional semantics are supported. The following are some of the weakened guarantees that are usually associated with such STMs:

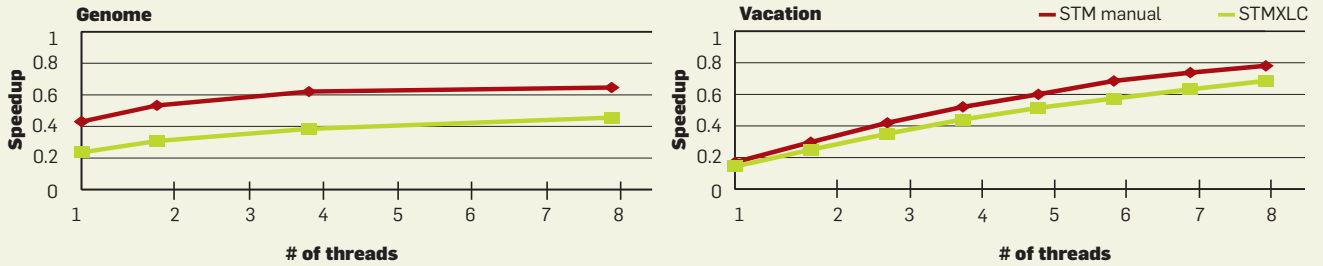
► **Weak atomicity:** Typically the STM runtime libraries cannot detect conflicts between transactions and non-transactional accesses. Thus, the semantics of atomicity are weakened to allow undetected conflicts with non-transactional accesses (referred to as weak atomicity³), or equivalently put the burden on the programmer to guarantee that no such conflicts can possibly take place.

► **Privatization:** Some STM designs prohibit the seamless privatization of memory locations, that is, the transition from being accessed transactionally to being accessed privately—or non-transactionally in general, by using locks. For some STM designs, once a location is accessed transactionally, it must continue to be accessed transactionally. With some STM designs, the programmer can ease the transition by guaranteeing that the first access to the privatized location—such as after the location is no longer accessible by other threads—is transactional.

► **Memory reclamation:** Some STM designs prohibit the seamless reclamation of the memory locations accessed transactionally for arbitrary reuse, such as using `malloc` and `free`. With such STM designs, memory allocation and deallocation for locations accessed transactionally are handled differently from other locations.

► **Legacy binaries:** STM needs to observe all memory activities of the transactional regions to ensure atomicity and isolation. STMs that achieve this observation by code instrumentation gener-

Figure 3: Scalability results for manual and compiler instrumented benchmarks on AIX PowerPC with IBM XLCSTM compiler.



ally cannot support transactions calling legacy codes that are not instrumented (for example, third-party libraries) without seriously limiting concurrency, such as by serializing transactions.

Evaluation

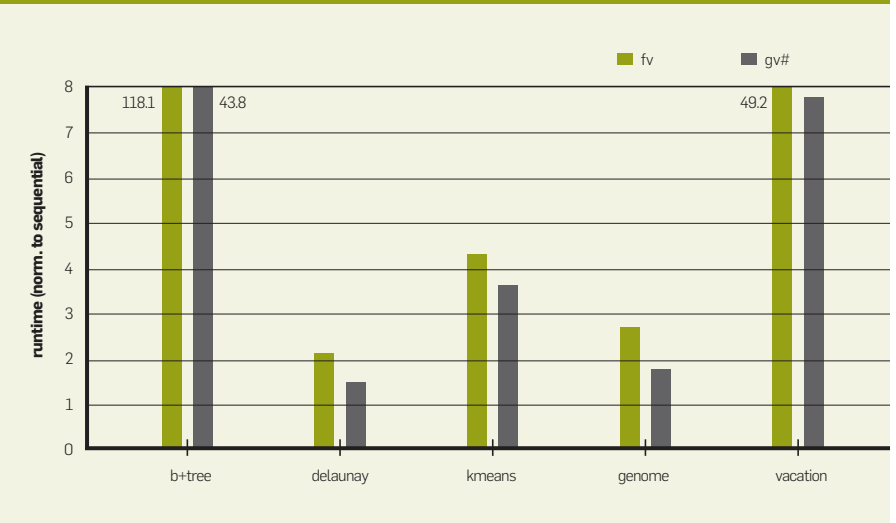
Here we use the following set of benchmarks:

- *b+tree* is an implementation of database indexing operations on a b-tree data structure for which the data is stored only on the tree leaves. This implementation uses coarse-grain transactions for every tree operation. Each b+ tree operation starts from the tree root and descends down to the leaves. A leaf update may trigger a structural modification to rebalance the tree. A rebalancing operation often involves recursive ascent over the child-parent edges. In the worst case, the rebalancing operation modifies the entire tree. Our workload inserts 2,048 items in a b+tree of order 20. For this code we have only a transactional version that is not manually instrumented, therefore experimental results are presented only in configurations where we can use our compiler to provide instrumentation;

- *delaunay* implements the Delaunay Mesh Refinement algorithm described in Kulkarni et al.¹⁵ The code produces a guaranteed quality Delaunay mesh. This is a Delaunay triangulation with the additional constraint that no angle in the mesh be less than 30 degrees. The benchmark takes as input an unrefined Delaunay triangulation and produces a new triangulation that satisfies this constraint. In the TM implementation of the algorithm, multiple threads choose their elements from a work-queue and refine the cavities as separate transactions.

- *genome*, *kmeans*, and *vacation* are part of the STAMP benchmark suite¹⁹

Figure 4: Single-threaded overhead of the STM algorithms.



version 0.9.4. For a detailed description of these benchmarks see STAMP.³⁰

Baseline Performance. In Figure 2 we present a performance comparison of three STMs: the IBM,^{31, 34} Intel,¹⁴ and Sun's TL2⁷ STMs. The runs are on a quad-core, two-way hyperthreaded Intel Xeon 2.3GHz box running Linux Fedora Core 6. In these runs, we used the manually instrumented versions of the codes that aggressively minimize the number of barriers for the IBM and TL2 STMs. Since we do not have access to low-level APIs for the Intel STM, the curves for the Intel STM are from codes instrumented by its compiler, which incur additional barrier overheads due to compiler instrumentation.³⁶ The graphs are scalability curves with respect to the serial, non-transactionalized version. Therefore a value of 1 on the y-axis represents performance equal to the serial version. The performance of these STMs is mostly on par, with the IBM STM showing better scalability on *delaunay* and TL2 obtaining better scalability on *genome*. However, the overall performance obtained is very low: on *kmeans* the IBM

STM barely attains single thread performance at 4 threads, while on *vacation* none of the STMs actually overcome the overhead of transactional memory even with 8 threads.

Compiler Instrumentation. The compiler is a necessary component of an STM-based programming environment that is to be adopted by mass programmers. Its basic role is to eliminate the need for programmers to manually instrument memory references to STM read- and write-barriers. While offering convenience, compiler instrumentation does add another layer of overheads to the STM system by introducing redundant barriers, often due to conservativeness of compiler analysis, as also observed in Yoo.³⁶

Figure 3 provides another baseline: the overhead of compiler instrumentation. The performance is measured on a 16-way POWER5 running AIX 5.3. For the STMXLC curve, we use the uninstrumented versions of the codes and annotate transactional regions and functions using the language extensions provided by the compiler.³¹

Figure 5: Percentage of time spent in different STM operations.

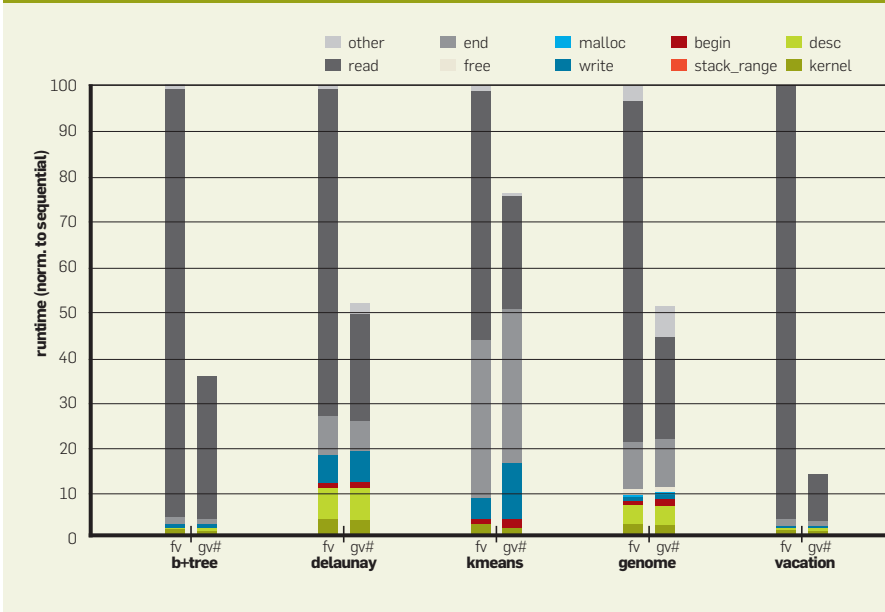
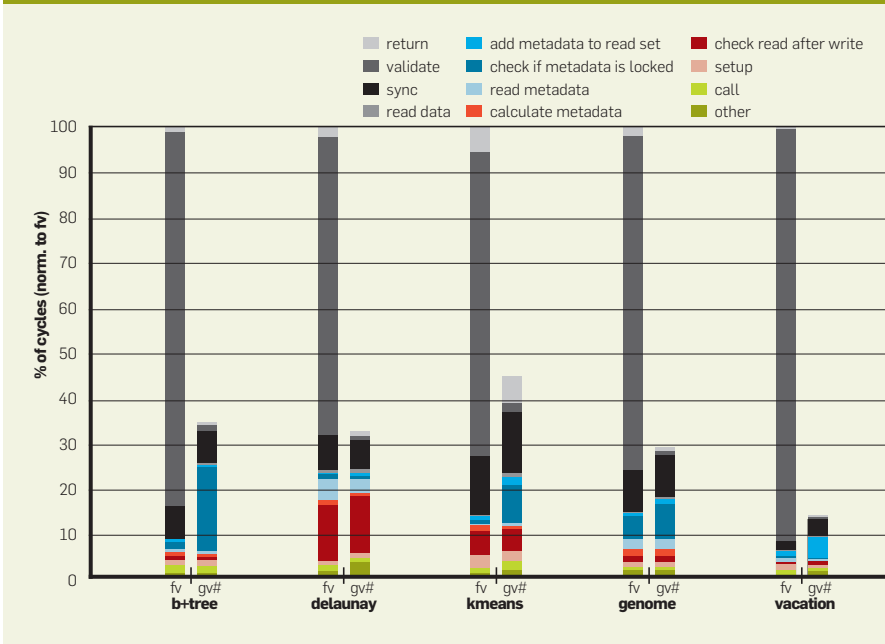


Figure 6: Percentage of time spent in STM read sub-operations.



Compiler over-instrumentation is more pronounced in traditional, unmanaged languages, such as C and C++, where a compiler instrumentation without interprocedural analysis may end up instrumenting every memory reference in the transactional region (except for stack accesses). Indeed, our compiler instrumentation more than doubled the number of dynamic read barriers in *delaunay*, *genome*, and *kmeans*. Interprocedural analysis can help improve the tightness of compiler instrumentation for some cases, but is generally limited by the accuracy of global analysis.

STM Operations Performance. Given this baseline, we now analyze in detail which operations in the STM cause the overhead. For this purpose, we use a cycle-accurate simulator of the PowerPC architecture that provides hooks for instrumentation. The STM operations and suboperations are instrumented with these simulator hooks. The reason for this environment is that we want to capture the overheads at instruction level and eliminate any other non-determinism introduced by real hardware. The simulator eliminates all other bookkeeping operations introduced by

instrumentation and provides an accurate breakdown of the STM overheads.

We study the performance of two STM algorithms: one that fully validates (“fv”) the read set after each transactional read and one that uses a global version number (“gv#”) to avoid the full validation, while maintaining the correctness of the operations. The fv algorithm provides more concurrency at a much higher price. The gv# is deemed as one of the best trade-offs for STM implementations.

Figure 4 presents the single-threaded overhead of these algorithms over sequential runs, illustrating again the substantial slowdowns that the algorithms induce. Figure 5 breaks down these overheads into the various STM components. For both algorithms, the overhead of transactional reads dominates due to the frequency of read operations relative to all other operations. The effectiveness of the global version number in reducing overheads is shown in the lower read overhead of “gv#.”

Figure 6 gives a fine-grain breakdown of the overheads of the transactional read operation. As expected, the overhead of validating the read set dominates transactional read time in the “fv” configuration. For both algorithms, the sync operations (necessary for ordering the metadata read and data read as well as the data read and validation) form a substantial component. In applications that perform writes before reads in the same transaction (*delaunay*, *kmeans*), the time spent checking whether a location has been written by prior writes in the same transaction forms a significant component of the total time. Interestingly, reading the data itself is a negligible amount of the total time, indicating the hurdles that must be overcome for the performance of these algorithms to be compelling.

Figure 7 gives a similar breakdown of the transactional commit operation. As before, the “fv” configuration suffers from having to validate the read set. Other dominant overheads for both configurations are that of having to acquire the metadata for the write set (which involves a sequence of load-linked/store-conditional operations) and the sync operations that are necessary for ordering the metadata acquires, data writes, and metadata releases. Once again, the data writes themselves form a small

component of the total time.

Overhead Optimizations. There have been many proposals on reducing STM overheads through compiler or runtime techniques, most of which are complementary to STM hardware acceleration.

► *Redundant barrier elimination.* One technique is to eliminate barriers to thread-local objects through escape analysis. Such analysis is typically quite effective identifying thread-local accesses that are close to the object allocation site. It can eliminate both read- and write-barriers, but is often more effective on write-barriers. For example, we observe that an intra-procedural escape analysis can eliminate 40–50% of write barriers in *vacation*, *genome*, and *b+tree*. However, its impact on performance is more limited: from negligible to 12%. To target redundant read-barriers, a whole-program analysis called Not-Accessed-In-Transaction analysis²⁷ eliminates some barriers to read-only objects in transactions;

► *Barrier strength reduction.* These optimizations do not eliminate barriers, but identify at runtime special locations that require only lightweight barrier processing, such as dynamic tracking of thread-local objects^{11,27} and runtime filtering of stack references and duplicate references;¹¹

► *Code generation optimizations.* One common technique is to inline the fast path of barriers. It has the potential benefit of reducing function call overhead, increasing ILP, and exposing reuse of common sub-barrier operations. In our experiments, compiler inlining achieved less than 2% overall improvement across our benchmark suite;

► *Commit sequence optimizations.* Eliminating unnecessary global version number updates³⁷ improves the overall performance of several micro-benchmarks by up to 14%.

Such optimizations have a positive impact on STM performance. However, the results presented here indicate how much further innovation is needed for the performance of STMs to become generally appealing to users.

Related Work

The first STM system was proposed by Shavit and Touitou²⁶ and is based on object ownership. The protocol is static, which is a significant shortcoming that has been overcome by subsequently pro-

posed STM systems.⁷ Conflict detection is simplified significantly by the static nature because conflicts can be ruled out already when ownership records are acquired (at transaction start).

DSTM¹² is the first dynamic STM system; the design follows a per-object runtime organization (locator object). Variables (objects) in the application heap refer to a locator object. Unlike in a design with ownership records (for example, Harris and Fraser¹⁰), the locator does not store a version number but refers to the most recently committed version of the object. A particularity of the DSTM design is that objects must be explicitly ‘opened’ (in read-only or read-write mode) before transactional access; also DSTM allows for early release. The authors argue that both mechanisms facilitate the reduction of conflicts.

The design principles of the RSTM¹⁸ system are similar to DSTM in that it associates transactional metadata with objects. Unlike DSTM however, the system does not require the dynamic allocation of transactional data but co-locates it with the non-transactional data. This scheme has two benefits: first, it facilitates spatial access locality and hence fosters execution performance and transaction throughput. Second, the dynamic memory management of transactional data (usually done through a garbage collector) is not necessary and

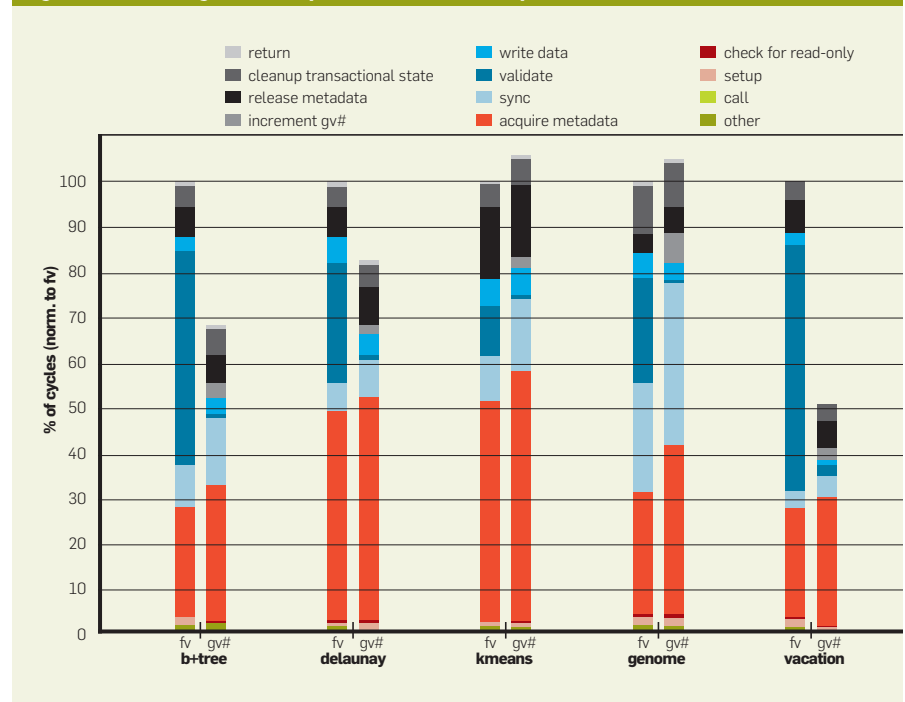
hence this scheme is amenable for use in environments where memory management is explicit.

Recent work explored algorithmic optimizations and/or alternative implementations of the basic STM algorithms described here. Riegel et al. propose the use of real-time clocks to enhance the STM scalability using a global version number.²² JudoSTM²¹ and RingSTM²⁹ reduce the number of atomic operations that must be performed when committing a transaction at the cost of serializing commit and/or incurring spurious aborts due to imprecise conflict detection. Several proposals have been made for STMs that operate via dynamic binary rewriting in order to allow the usage of STM on legacy binaries.^{8,21,33}

Yoo et. al³⁶ analyze the overhead in the execution of Intel’s STM.^{14,23} They identify four major sources of overhead: over-instrumentation, false sharing, amortization costs, and privatization-safety costs. False sharing, privatization-safety, and over-instrumentation are implementation artifacts that can be eliminated by either using finer granularity bookkeeping, more refined analysis, or user annotations. Amortization costs are inherent overheads in an STM that, as we demonstrated here, are not likely to be eliminated.

A large amount of research effort has been spent in analyzing the opera-

Figure 7: Percentage of time spent in STM end sub-operations.




tions in TM systems. Recent software optimizations have managed to accelerate STM performance by 2%–15%. We believe such analysis is a good practice that should be extended to every piece of system software, especially open source. However, the gains are only a minor dent in the overheads we observed, indicating the challenge that lies before the community in making STM performance compelling.

Conclusion

Based on our results, we believe that the road ahead for STM is quite challenging. Lowering the overheads of STM to a point where it is generally appealing is a difficult task and significantly better results have to be demonstrated. If we could stress a single direction for further research, it is the elimination of dynamically unnecessary read and write barriers—possibly the single most powerful lever toward further reduction of STM overheads. However, given the difficulty of similar problems explored by the research community such as alias analysis, escape analysis, and so on, this may be an uphill battle. And because the argument for TM hinges upon its simplicity and productivity benefits, we are deeply skeptical of any proposed solutions to performance problems that require extra work by the programmer.

We observed that the TM programming model itself, whether implemented in hardware or software, introduces complexities that limit the expected productivity gains, thus reducing the current incentive for migration to transactional programming, and the justification at present for anything more than a small amount of hardware support.

Acknowledgments

We would like to thank Pratap Pattnaik for his continuous support, Christoph von Praun for numerous discussions, work on benchmarks and runtimes, and Rajesh Bordawekar for the B+tree code implementation. 

References

- Baugh, L., Neelakantam, N., and Zilles, C. Using hardware memory protection to build a high-performance, strongly-atomic hybrid transactional memory. In *Proceedings of the 35th International Symposium on Computer Architecture*. IEEE Computer Society, Washington, DC, 2008, 115–126.
- Blundell, C., Devietti, J., Lewis, E.L., Martin, M.M.K. Making the fast case common and the uncommon case simple in unbounded transactional memory. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*. ACM, NY, 2007.
- Blundell, C., Lewis, C., and Martin, M.M.K. Subtleties of transactional memory atomicity semantics. *IEEE TCCA Computer Architecture Letters* 5, 2 (Nov 2006).
- Bobba, J., Goyal, N., Hill, M.D., Swift, M.M., and Wood, D.A. TokenTM: Efficient execution of large transactions with hardware transactional memory. In *Proceedings of the 35th International Symposium on Computer Architecture*. IEEE Computer Society, Washington, D.C., 2008, 127–138.
- Ceze, L., Tuck, J., Cascaval, C., Torrellas, J. Bulk disambiguation of speculative threads in multiprocessors. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*. ACM, NY, 2006, 237–238.
- Damron, P., Federova, A., Lev, Y., Luchangco, V., Moir, M., and Nussbaum, D. Hybrid transactional memory. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 2006.
- Dice, D., Shalev, O., and Shavit, N. Transactional Locking II. *DISC*, Sept. 2006, 194–208.
- Felber, P., Fetzer, C., Mueller, U., Riegel, T., Suesskraut, M., and Sturzhelm, H. Transactifying applications using an open compiler framework. In *Proceedings of the ACM SIGPLAN Workshop on Transactional Computing*, Aug. 2007.
- Hammond, L., Wong, V., Chen, M., Carlstrom, B.D., Davis, J.D., Hertzberg, B., Prabhu, M.K., Wijaya, H., Kozyrakis, C., and Olukotun, K. Transactional memory coherence and consistency. In *Proceedings of the 31st Annual International Symposium on Computer Architecture*. IEEE Computer Society, June 2004, 102.
- Harris, T. and Fraser, K. Language support for lightweight transactions. In *Proceedings of Object-Oriented Programming, Systems, Languages, and Applications*. Oct. 2003, 388–402.
- Harris, T., Plesko, M., Shinnar, A., and Tarditi, D. Optimizing memory transactions. In *Proceedings of the Programming Language Design and Implementation Conference*. 2003, 388–402.
- Herlihy, M., Luchangco, V., Moir, M., and Scherer III, W.N. Software transactional memory for dynamic-sized data structures. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing*. July 2003, 92–101.
- Herlihy, M. and Moss, J.E.B. Transactional memory: Architectural support for lock-free data structures. In *Proceedings of the 20th Annual International Symposium on Computer Architecture*. May 1993.
- Intel C++ STM compiler, prototype edition 2.0.; <http://softwarecommunity.intel.com/articles/eng/1460.htm/> (2008).
- Kulkarni, M., Pingali, K., Walter, B., Ramnarayanan, G., Bala, K., and Chew, P.L. Optimistic parallelism requires abstractions. In *Proceedings of the PLDI 2007*. ACM, NY, 2007, 211–222.
- Larus, J.R., and Rajwar, R. *Transactional Memory*. Morgan Claypool, 2006.
- The Lonestar benchmark suite; <http://iss.ices.utexas.edu/lonestar/> (2008).
- Marathe, V.J., Spear, M.F., Heriot, C., Acharya, A., Eisenstat, D., Scherer III, W.N., and Scott, M.L. Lowering the overhead of software transactional memory. Technical Report TR 893, Computer Science Department, University of Rochester, Mar 2006. Condensed version submitted for publication.
- Minh, C.C., Trautmann, M., Chung, J., McDonald, A., Bronson, N., Casper, J., Kozyrakis, C., and Olukotun, K. An effective hybrid transactional memory system with strong isolation guarantees. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*. ACM, NY, 2007, 69–80.
- Moore, K.E., Bobba, J., Moravan, M.J., Hill, M.D., and Wood, D.A. LogTM: Log-based transactional memory. In *Proceedings of the 19th ACM Symposium on High Performance Computer Architecture*, Feb 2006.
- Olszewski, M., Cutler, J., Steffan, J.G. Judostm: A dynamic binary-rewriting approach to software transactional memory. In *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*. 2007. IEEE Computer Society, Washington D.C., 365–375.
- Riegel, T., Fetzer, C., and Felber, P. Time-based transactional memory with scalable time bases. In *Proceedings of the 19th ACM Symposium on Parallelism in Algorithms and Architectures*, 2007.
- Saha, B., Adl-Tabatabai, A.R., Hudson, R.L., Minh, C.C., and Hertzberg, B. Mrcrt-stm: A high performance software transactional memory system for a multi-core runtime. In *Proceedings of the 11th ACM Symposium on Principles and Practice of Parallel Programming*, Mar. 2006, ACM, NY, 187–197.
- Saha, B., Adl-Tabatabai, A.R., and Jacobson, Q. Architectural support for software transactional memory. In *Proceedings of the 39th Annual International Symposium on Microarchitecture*. Dec. 2006, 185–196.
- Shavit, N., and Touitou, D. Software Transactional Memory. In *Proceedings of the ACM Symposium of Principles of Distributed Computing*. ACM, 1995.
- Shavit, N. and Touitou, D. Software transactional memory. In *Proceedings of the 14th ACM Symposium on Principles of Distributed Computing*. ACM, NY, 1995.
- Shpeisman, T., Menon, V., Adl-Tabatabai, A-R., Balensiefer, S., Grossman, D., Hudson, R., Moore, K.F., and Saha, B. Enforcing isolation and ordering in STM. In *Proceedings of Proceedings of the Programming Language Design and Implementation Conference*. ACM, 2007, 78–88.
- Shriraman, A., Spear, M.F., Hossain, H., Marathe, V.J., Dwarakadas, S., and Scott, M.L. An integrated hardware-software approach to flexible transactional memory. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*. ACM, NY, 2007, 104–115.
- Spears, M.T., Michael, M.M., and von Praun, C. Ringstm: Scalable transactions with a single atomic instruction. In *Proceedings of the 20th ACM Symposium on Parallelism in Algorithms and Architectures*. ACM, NY, 275–284.
- STAMP benchmark; <http://stamp.stanford.edu/> (2007).
- (IBM) XL C/C++ for Transactional Memory for AIX; <http://www.alphaworks.ibm.com/tech/xlccstm/> (2008).
- Tremblay, M. and Chaudhry, S. A third generation 65nm 16-core 32-thread plus 32-scout-thread CMT. In *Proceedings of the IEEE International Solid-State Circuits Conference*. Feb. 2008.
- Wang, C., Chein, W-Y, Wu, Y., Saha, B., and Adl-Tabatabai, A.R. Code generation and optimization for transactional memory constructs in an unmanaged language. In *Proceedings of International Symposium on Code Generation and Optimization*. 2007, 34–48.
- Wu, P., Michael, M.M., von Praun, C., Nakaïke, T., Bordawekar, R., Cain, H.W., Cascaval, C., Chatterjee, S., Chiras, S., Hou, R., Mergen, M., Shen, X., Spear, M.F., Wang, H.Y., and Wang, K. Compiler and runtime techniques for software transactional memory optimization. To appear in *Concurrency and Computation: Practice and Experience*, 2008.
- Yen, L., Bobba, J., Marty, M.M., Moore, K.E., Volos, H., Hill, M.D., Swift, M.M., and Wood, D.A. LogTM-SE: Decoupling hardware transactional memory from caches. In *Proceedings of the 13th International Symposium on High-Performance Computer Architecture*. Feb 2007.
- Yoo, R.M., Ni, Y., Welc, A., Saha, B. Adl-Tabatabai, A-R. and Lee, H-H.S. Kicking the tires of software transactional memory: why the going gets tough. *Proceedings of the 20th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, 2008.
- Zhang, R., Budimlic, Z. and Scherer III, W.N. Commit phase in timestamp-based STM. In *Proceedings of the 20th Annual Symposium on Parallelism in Algorithms and Architectures*. ACM, NY, 326–335.

Çâlin Caşcaval (cascaval@us.ibm.com) is a Research Staff Member and Manager of Programming Models and Tools for Scalable Systems at IBM TJ Watson Research Center, Yorktown Heights, NY.

Colin Blundell is a member of the Architecture and Compilers Group, Department of Computer and Information Science, University of Pennsylvania.

Maged Michael is a Research Staff Research Member at IBM TJ Watson Research Center, Yorktown Heights, NY.

Trey Cain is a Research Staff Member at IBM TJ Watson Research Center, Yorktown Heights, NY.

Peng Wu is a Research Staff Member at IBM TJ Watson Research Center, Yorktown Heights, NY.

Stefanie Chiras is a manager in IBM's Systems and Technology Group.

Siddhartha Chatterjee is director of the Austin Research Laboratory, IBM Research, Austin, TX.

**Virtualization technology is hot again,
but for the right reasons?**

BY MACHE CREEGER, MODERATOR

CTO Roundtable on Virtualization

THIS MONTH WE present the second in a series of ACM CTO Roundtable forums. Overseen by the ACM Professions Board, the goal of the forums is to provide high-powered expertise to practicing IT managers to help inform their decisions when investing in new architectures and technologies.

The topic of this forum is virtualization. When investing in virtualization technologies, IT managers must know what is considered standard practice and what is considered too leading-edge and risky for near-term deployment. For this forum we've assembled five leading experts on virtualization to discuss what those best practices should be. While the participants might not always agree with each other, we hope their insights will help IT managers navigate the virtualization landscape and make informed decisions on how best to use the technology. Next month we will present Part II of this forum, discussing such topics as clouds and virtualization, using virtualization to streamline desktop delivery, and how to choose appropriate virtual machine platforms and management tools.

Participants

Mache Creeger (Moderator): Creeger is a longtime technology industry veteran based in Silicon Valley. Along with being an *ACM Queue* columnist, he is the principal of Emergent Technology Associates, marketing and business development consultants to technology companies worldwide.

Tom Bishop is CTO of BMC Software. Prior to BMC, Bishop worked at Tivoli, both before and after their initial public offering and acquisition by IBM, and also at Tandem Computers. Earlier in his career Bishop spent 12 years at Bell Labs' Naperville, IL facility and then worked for UNIX International. He graduated from Cornell University with both bachelor's and master's degrees in computer science.

Simon Crosby is the CTO of the Virtualization Management Division

at Citrix. He was one of the founders of XenSource and was on the faculty of Cambridge University, where he earned his Ph.D. in computer science. Crosby grew up in South Africa and has master degrees in applied probability and computer science.

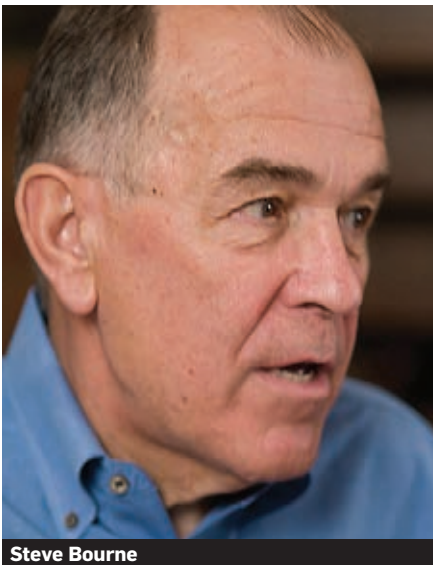
Gustav. This is a pseudonym due to the policies of his employer, a large financial services company where he runs distributed systems. Early in his career, Gustav wrote assembler code for telephone switches and did CAD/CAM work on the NASA space station *Freedom*. He later moved to large system design while working on a government contract and subsequently worked for a messaging and security

alization, virtualization management, and application virtualization. Stewart is a Microsoft Certified Architect and is on the Board of Directors of the Microsoft Certified Architect Program.

Steve Herrod is the CTO of VMware, where he's worked for seven years. Before VMware, Herrod worked in Texas for companies such as EDS and Bell Northern Research. Earlier in his career Herrod studied with Mendel Rosenblum, the founder of VMware, at Stanford and then worked for TransMeta, a computer hardware and software emulation company.

Steve Bourne is chair of the ACM Professions Board. He is also a past president of the ACM and Editor-in-

SIMON CROSBY: The power-savings issue is a big red herring because the CPU is a small portion of the total power consumption compared to spinning all those disk drives in a storage array. I'll be the first one to say that free, ubiquitous CPU virtualization is just an emergent property of Moore's Law, just part of the box. Memory is another major power consumer and memory architectures are definitely not keeping up. When you're talking about virtualizing infrastructure, you should be talking about what bits of it you virtualize and how: CPU, storage, and/or memory. You have to look at the whole thing. As for showing lower overall power consumption, I have yet



Steve Bourne



Mache Creeger



Allen Stewart

startup company in Silicon Valley, taking it public in the mid-1990s. After starting his own consulting firm, he began working at his first large financial firm. Seven or eight years later, he landed at his current company.

Allen Stewart is a Principle Program Manager Lead in the Windows Server Division at Microsoft. He began his career working on Unix and Windows operating systems as a system programmer and then moved on to IBM, where he worked on Windows systems integration on Wall Street. After IBM, Stewart joined Microsoft, where for the first six years he worked as an architect in the newly formed Financial Services Group. He then moved into the Windows Server Division Engineering organization to work on Windows Server releases. His primary focus is virtualization technologies: hardware virtu-

Chief of the *ACM Queue* editorial advisory board. A fellow alumnus with Simon Crosby, Bourne received his Ph.D. from Trinity College, Cambridge. Bourne held management roles at Cisco, Sun, DEC, and SGI and currently is CTO at El Dorado Ventures, where he advises the firm on their technology investments.

MACHE CREEGER: Virtualization is a technology that everyone is talking about, and with the increased cost of energy the server consolidation part of the value proposition has become even more compelling. Let's take that as a given and go beyond that. How do we manage large numbers of virtualized servers and create an integral IT architecture that's extensible, scalable, and meets all the criteria that reasonable people can agree on?

to see a good calculation for that.

GUSTAV: I support virtualization for a number of reasons, but cost savings isn't one of them. What I typically see is that the server guy makes a decision to reduce his costs, but that significantly impacts storage and the network, making their costs go up.

To put eight software services on a single machine, instead of buying the \$3,000 two-socket 4GB 1U blade, I bought the four-socket, 16GB system for \$20,000. While that calculation provides an obvious savings, because I want to use VMotion I have to purchase an additional storage array that can connect to two servers. The result is that I paid more money than the traditional architecture would cost to support the same service set.

That's why you see a large interest in virtualization deployment followed

by this trough of disillusionment. Once deployed, people find out that a) it's hard and b) oddly, they are spending all this money on storage.

SIMON CROSBY: I agree with that. Solving storage is the hardest problem of all. It's the big bear in the room.

From an infrastructural cost perspective, you have to consider the human cost: The number of administrators still grows linearly with the number of VMs (virtual machines). Server-consolidation costs are very real. However, as virtualization addresses the mainstream production workload, costs are still going to be driven by the number of administrators. Unless the administrator problem is solved, IT has not re-

threaded has taken off at all.

TOM BISHOP: I think that is the elephant in the room. We as an industry haven't built the correct abstraction which can fungibly apply computing to a domain and allow dynamic apportioning of computing capacity to the problems to be solved. Storage is a piece of it, but ultimately it's more complex than just that.

If you take virtual memory as an example, we spent approximately 20 years trying to figure out the correct abstraction for memory. One can define memory consumption using one abstraction and build a set of automated mechanisms to dynamically allocate a much smaller physical resource

amount of demand, and I want to satisfy that demand with my capacity, using an affordable cost equation in a way that is moment-to-moment optimal for heterogeneous systems.

GUSTAV: The beauty of having a good working abstraction of an underlying function is that the operating system can give you less than you asked for and still provide acceptable results. Virtual memory is a good example.

MACHE CREEGER: So how does this apply to the poor guy who's struggling in a small to medium-size company? What is he supposed to take away from all this?

SIMON CROSBY: The single-server notion of virtualization—server con-



Tom Bishop



Simon Crosby



Steve Herrod

ally been transformed.

GUSTAV: Fully using all the cores on the die is one of the biggest things that drive the need for virtualization. In the not-too-distant future we are going to be forced to buy eight-core CPUs. Because our computational problem has not grown as fast as the ability to throw CPU resources at it, we will not be able to keep all those cores fully utilized.

It's not an issue of power savings, as Intel will nicely power down the unused cores. The real benefit of multi-core CPUs is to be able to address the huge legacy of single-threaded processes in parallel.

SIMON CROSBY: I think that should be the high-order bit. There are 35 million servers out there, all of which are single threaded. So let's just admit that, run one VM per core, and have a model that actually works. I don't think multi-

to a much larger virtual demand and still achieve optimal performance moment-to-moment. We have not found that same abstraction for the core IT mission of delivering application services.

SIMON CROSBY: But isn't determining the biting constraint the fundamental problem with any one of these things? I don't care about optimizing something that is not a biting constraint. It may be memory, CPU, storage, and/or power—I have no clue. Different points of the operating spectrum might have different views on what is or is not a biting constraint. The grid guys, the network guys—all have different views and needs.

TOM BISHOP: I predict we will develop a workable abstraction for representing the basic IT mission: *I've got a certain amount of capacity, a certain*

solidation—is very well established and basically free. It will be part of every server, an emergent property of Moore's Law, and multiple vendors will give it to you.

The orchestrated assignment of resources across the boundaries of multiple servers or even multiple different resources is a major problem and I don't think we really have begun to understand it as yet.

TOM BISHOP: Regarding vendor lock-in, I think there is a body of experience, certainly in the companies I speak to, that says the heck with vendor lock-in. If my chance of getting a solution I can live with is better if I limit myself to a single vendor, I'm prepared to accept that trade-off.

GUSTAV: If you have a relatively small number of servers, there is no problem that you have that virtualization

can't solve. However, there will be new things that you may choose to spend money on that in the past were not a problem, such as doing live migration between servers. But if you just want to run a shop that has up to around 20 servers, unless you're doing something really weird, you should do it. It is easy and readily available from any of the major vendors. This addresses the relatively easy problem of "Can I put three things on one server?"

If you then realize you have new problems, meaning "Now that I have three things on one server, I want that server to be more available or I want to migrate stuff if that server fails," this is a level of sophistication that the market is only beginning to address. Different vendors have different definitions.

SIMON CROSBY: Once you achieve basic virtualization, the next big issue is increasing overall availability. If I can get higher availability for some key workloads, that transforms the business.

STEVE HERROD: I agree. In fact, we currently have a large number of customers that buy one VM per box first and foremost for availability and second for provisioning.

TOM BISHOP: About two years ago, the best session at a conference I attended was "Tales from the Front, Disaster Recovery Lessons Learned from Hurricane Katrina." A large aerospace company had two data centers, one just south of New Orleans and another one about 60 miles away in Mississippi. Each center backed the other up and both ended up under 20 feet of water.

The lesson they learned was to virtualize their data center. In response to that experience they built a complete specification of their data center where it could be instantiated instantaneously and physically anywhere in the world.

MACHE CREEGER: Our target IT manager is trying to squeeze a lot out of his budget, to walk the line between what's over the edge and what's realistic. Are you saying that all this load balancing, dynamic migration, that the marketing literature from Citrix, VMware, and Microsoft defines as the next big hurdle and the vision for where virtualization is going are not what folks should be focusing on?

SIMON CROSBY: Organizations today build organizational structures around current implementations of technology, but virtualization changes all of it. The biggest problem we have right now is that we have to change the architecture of the IT organization. That's people's invested learning and their organizational structure. They're worried about their jobs. That's much harder than moving a VM between two servers.

MACHE CREEGER: A while back I did a consulting job for a well-known data-center automation company and they brought up this issue as well. When you change the architecture of the data center you blow up all the traditional boundaries that define what people do for a living—how they develop their careers, how they got promoted, and everything else. It's a big impediment for technology adoption.

SIMON CROSBY: One of the reasons I think that cloud-based IT is very interesting is none of the cloud vendors have invested in the disaster of today's typical enterprise IT infrastructure. It is horrendously expensive because none of it works together; it's unmanageable except with a lot of people. Many enterprise IT shops have bought one or more expensive proprietary subsystems that impose significant labor-intensive requirements to make it all work.

Clouds are way cheaper to operate because they build large, flat architectures that are automated from the get-go, making the cost for their infrastructure much lower than most companies' enterprise IT. If I'm Amazon Web Services and I want to offer a disaster recovery service, the numbers are in my favor. I need only provide enough additional capacity to address the expected failure rate of my combined customer set, plus a few spares and, just like an actuary, determine the risks and cost. A very simple and compelling business model.

ALLEN STEWART: The thing that challenges the cloud environment and most enterprise data centers is the heterogeneity of the shop and the types of applications they run. To take advantage of the cloud, you have to develop an application model that suits disconnected state and applications. I think that challenges enterprise IT

shops, because they look out and see a completely dissimilar range of applications without a common development framework.

GUSTAV: I just built two data centers and fully populated them. If I look at the stereotypical cloud case right now, EC2 (www.amazon.com/ec2) is about \$0.80 per hour per eight-CPU box. My cost, having bought the entire data center, is between \$0.04 and \$0.08.

Having bought the entire data center, I have the budget and scale to blow away that \$0.80 EC2 pricing. SMBs (small- and medium-size businesses) probably do not have that option. The cloud guys can produce tremendous margin for themselves by producing the scale of an entire data center and selling parts of it to SMBs.

TOM BISHOP: The model that's going to prevail is exactly the way the power companies work today. Every company that builds power-generation capacity has a certain model for their demand. They build a certain amount of capacity for some base level demand and then they have a whole set of very sophisticated provisioning contracts.

MACHE CREEGER: Or reinsurance.

TOM BISHOP: Reinsurance to basically go get electricity off the grid when they need it. So the power we get is a combination of locally generated capacity and capacity bought off the grid.

SIMON CROSBY: As a graduate student, I read a really interesting book on control theory that showed mathematically that arbitrage is fundamental to the stability of a market and the determination of true market price. Based on that statement, virtualization is just an enabler of a relatively efficient market for data center capacity; it's a provisioning unit of resource.

Virtualization allows for late binding, which is generally considered to be a good thing. Late binding means I can lazily (that is, just-in-time) compose my workload (a VM) from the OS, the applications, and the other relevant infrastructural components. I can bind them together at the last possible moment on the virtualized infrastructure, delaying the resource commitment decision as long as possible to gain flexibility and dynamism. Virtualization provides an abstraction that allows us to late bind on resources.

STEVE HERROD: The opportunity to

have a VM and to put the policy around that VM for late binding is pretty powerful. You create your application or your service, which might be a multi-machine service, and you associate with it the security level you want, the availability level you want, and the SLAs (service level agreements) that should go with it. The beauty of this bubble, which is the workload and the policy, is it can move from one data center to another, or to an offsite third party, if it satisfies the demands that you've wrapped around it.

GUSTAV: Our administrative costs generally scale in a nonlinear fashion, but the work produced is based on the number of operating-system instances more than the number of hardware instances. The number of servers may drive some capital costs, but it doesn't drive my support costs.

TOM BISHOP: What you're really managing is state. The more places you have state in its different forms, the more complex your environment is and the more complex and more expensive it is to manage.

SIMON CROSBY: I'll disagree. You're managing bindings. The more bindings that are static, the worse it is, the more they are dynamic, the better it is.

We have a large financial services customer that has 250,000 PCs that need to be replaced. They want to do it using VDI (virtual desktop infrastructure) running desktop OSes as VMs in the data center to provide a rich, remote desktop to an appliance platform.

Following the "state" argument, we would have ended up with 250,000 VMs consuming a lot of storage. By focusing on bindings, given that they only support Windows XP or Vista, we really need only two VM images for the base OS. Dynamically streaming in the applications once the user has logged in allows us to provide the user with a customized desktop, but leaves us with only two golden-image VM templates to manage through the patch-update cycle.

Steve Herrod, Mike Neil from Microsoft, and I have been working on an emerging standard called OVF to define a common abstraction to package applications into a container. Under this definition, an application is some number of template VMs, plus all the



STEVE HERROD

The opportunity to have a VM and to put the policy around that VM for late binding is pretty powerful. You create your application or your service. The beauty of this bubble, which is the workload and the policy, is it can move from one data center to another, or to an offsite third party, if it satisfies the demands that you've wrapped around it.



metadata about how much resource they need, how they're interconnected, and how they should be instantiated.

We started working on it because there was the potential for a "VHS versus Betamax" virtual-hard-disk format war and none of us wanted that to happen. It started out as a portable virtual-machine format but is now emerging into more of an application description language. The container contains one instance of every component of the application, but when you roll it out at runtime you may request multiple copies. I think that's a very important step forward in terms of standardization.

STEVE HERROD: Virtualization breaks up something that's been unnaturally tied together. However, allowing late binding introduces new problems. If you cannot be more efficient with virtualization, then you shouldn't be using it.

We do surveys every single year on the number of workloads per administrator. Our numbers are generally good but it is because we effectively treat a server like a document and apply well-known document-management procedures to gain efficiencies. This approach forces you to put processes around things that did not have them before. For smaller companies that don't have provisioning infrastructure in place, it allows you much better management control. It's not a substitute for the planning part, but rather a tool that lets you wrap these procedures in a better way.

MACHE CREEGER: So how do people decide whether to choose VMware, Citrix, or Microsoft? How are people going to architect data centers with all the varying choices? Given that the vendors are just starting to talk about standards and that no agreements on benchmarking exist, on what basis are people expected to make architectural commitments?

GUSTAV: I think this is a place where the technology is ready enough for operations, but there are enough different management/software theories out there that I fully expect to have VMware, Microsoft, and Xen in different forms in my environment. That doesn't concern me nearly as much as having both SuSE and RedHat in my environment.

TOM BISHOP: Every customer we talk

to says they'll have at least three.

MACHE CREEGER: As a large enterprise customer, aren't you worried about having isolated islands of functionality?

GUSTAV: No, I have HP and Dell. That's a desirable case.

MACHE CREEGER: But that's different. They have the x86 platform; it's relatively standardized.

SIMON CROSBY: It's not. You'll never move a VM between AMD and Intel—not unless you're foolhardy. They have different floating point resolution and a whole bunch of other architectural differences.

People tend to buy a set of servers for a particular workload, virtualize the lot, and run that workload on those newly virtualized machines. If we treated all platforms as generic, things would break. AMD and Intel cannot afford to allow themselves to become undifferentiated commodities, and moreover, they have a legitimate need to innovate below the "virtual hardware line."

MACHE CREEGER: So you are saying that I'm going to spec a data center for a specific workload—spec it at peak, which is expensive—and keep all those assets in place specifically for that load. Doesn't that fly in the face of the discussions about minimizing capital costs, flexibility, workload migration, and high-asset utilization?

TOM BISHOP: You're making an assumption that every business defines risk in the same way. Gustav defines risk in a particular way that says "The cost of excess capacity is minuscule compared to the risk of not having the service at the right time."

MACHE CREEGER: In financial services, that's true, but there are other people that can't support that kind of value proposition for their assets.

SIMON CROSBY: That's an availability argument, where the trade-off is between having the service highly available on one end of the line, and lower capital costs, higher asset utilization, and lower availability at the other end. Virtualization can enhance availability.

GUSTAV: You will tend to use the VM, because while there are differences now at the hypervisor level, those differences are converging relatively rapidly and will ultimately disappear.

If you're worried about the long-



ALLEN STEWART

The thing that challenges the cloud environment and most enterprise data centers is the heterogeneity of the shop and the types of applications they run. To take advantage of the cloud, you have to develop to an application model that suits disconnected state and applications.



term trend of hypervisors, you're worried about the wrong thing. Choose the VM that is most compatible today to the application you are going to run. If you're doing desktop virtualization, you're probably going to use CXD (Citrix Xen Desktop). If you're doing Windows server virtualization, you're going to use either Veridian or, depending on what you're trying to do regarding availability management, VMware.

The first question to ask is "What are you used to?" That's going to determine what you're likely VM is. The second question is "What is the problem you're trying to solve?" The more complex the management problem the more attractive an integrated tool suite from VMware becomes. If you are saying "I don't have complex problems now but I'm going to have complex problems in three or four years," the more attractive Microsoft becomes. If you are going to build it on your own and/or have your own toolsets to integrate, which is most of the enterprise, you're going to find the Xen/Citrix option more attractive. If you're coming from the desktop side, you're at the other side of Citrix, and that is back to Xen. Where you're coming from is going to determine your VM product selection much more than where you're going to because they're all heading to the same place.

SIMON CROSBY: Looking at Microsoft Hyper-V/System Center and VMware VSX and Virtual Center (VC), both of these are complete architectures. Neither of them has a well-established ISV ecosystem significantly limiting customer choices. That said, I think the ecosystem around VMware is now starting to emerge due to the adoption of standards-based APIs.

What worries me is whether the missing functionality in any vendor's product needs to be developed by the vendor or whether the customer is OK with a solution composed of a vendor product and ISV add-ons. Both Stratus and Marathon offer fault-tolerant virtual machine infrastructure products using Citrix XenServer as an embedded component. That's because they focus on how to build the world's best fault tolerance whereas Citrix, VMware, and Microsoft do not. We have an open architecture, and that allows

the world's best talent to look at how to extend it and build solutions beyond our core competence. This is a very powerful model.

From an architectural perspective, I am absolutely passionate about the fact that virtualization should be open because then you get this very powerful model of innovation.

I have an ongoing discussion with one of the major analyst organizations because virtualization in their brains is shaped like VMware's products are shaped today. They think of it as a thing called ESX Server. So if VMware's ESX Server is viewed as a fully integrated car, then Xen should be viewed as a single engine. I would assert that because we don't know how virtualization is going to be in five years, you do not want to bind your consumption of virtualization to a particular car right now. As technology innovation occurs, virtualization will take different shapes. For example, the storage industry is innovating rapidly in virtualization and VMware cannot take advantage of it with their (current) closed architecture. Xen is open and can adapt: Xen runs on a 4,096 CPU supercomputer from SGI and it runs on a PC. That is an engine story; it is not a car story.

It's really critical we have an architecture that allows independent innovation around the components of virtualization. Virtualization is just a technology for forcing separation as far down the stack as you can—on the server, separated by the hypervisor, in the storage system—and then let's see how things build. I'm not in favor of any architecture which precludes innovation from a huge ecosystem.

STEVE HERROD: I actually agree on several parts. Especially for the middle market, the number-one thing that people need is something easy to use. I think there's a reasonable middle road which can provide a very nice framework or a common way of doing things, but also have tie in to the partner ecosystem as well. Microsoft has done this very well for a long time.

STEVE BOURNE: These bindings may be ABIs or they may not be, but they sound like the analogue of the ABIs. ABIs are a pain in the neck. So are these bindings a pain in the neck?

SIMON CROSBY: Bindings are a very hot

area. The hottest one for us right now is the VM you run on XenServer will run on Microsoft Hyper-V. This is a virtual hardware interface, where, when you move a virtual machine from one product to the other, the VM will still think it has the same hardware underneath it.

Right now if you take a VM from VMware and try to run on Citrix you will get a blue screen. It's just the same as if you took a hard disk out of a server and put it in another server and expected the OS to boot correctly. VMware and XenSource actually had discussions on how to design a common standard hardware ABI, but we couldn't get other major vendors to play.

If we actually were able to define an industry standard virtual hardware ABI, the first guys who'd try to break it would be Intel and AMD. Neither of those companies can afford for that line to be drawn because it would render all their differentiation meaningless, making their products undifferentiated commodities. Even if you move everything into the hardware, the ABIs would still be different.

In the ABI discussion there are two things that count. There's "Will the VM just boot and run?" Then it's "If the VM is up and running can I manage it using anybody's management tool?" I think we're all in the same position on standards-based management interfaces—DMTF (Distributed Management Task Force) is doing the job.

MACHE CREEGER: Let's take a moment to summarize.

Server consolidation should not be a focus of VM deployment. One should architect their data center around the strengths of virtualization such as availability and accessibility to clouds.

An IT architect should keep the operating application environment in mind as he makes his VM choices. Each of the VM vendors has particular strengths and one should plan deployments around those strengths.

In discussing cloud computing we said the kind of expertise resident in large enterprises may not be available to the SMB. Virtualization will enable SMBs and others to outsource data center operations rather than requiring investment in large, in-house facilities. They may be more limited in the types of application services

available, but things will be a lot more cost effective with a lot more flexibility than would otherwise be available. Using their in-house expertise, large enterprises will build data centers to excess, and either sell that excess computing capacity like an independent power generator or not, depending on their own needs for access to quick capacity.

GUSTAV: The one point we talked around, that we all have agreement on, is that server administrators will have to learn a lot more about storage and a lot more about networks than they were ever required to do before. We are back to the limiting constraint problem. The limiting constraint used to be the number of servers you had and given their configuration, how they were limited in what they could do. Now with virtualized servers the limiting constraint has changed.

With cheap gigabit Ethernet switches a single box only consumes 60 to 100 megabits. Consolidate that box and three others into a single box supporting four servers and suddenly I'm well past the 100 megabit limit. If I start pushing toward the theoretical limits with my CPU load, which is 40-to-1 or an average of 2%, suddenly I've massively exceeded GigE. There is no free lunch. Virtualization pushes the limiting constraint to either the network or to storage; it's one of those two things. When we look at places that screw up virtualization, they generally over consolidate CPUs, pushing great demands on network and/or storage.

You shouldn't tell your management that your target is 80% CPU utilization. Your target should be to utilize the box most effectively. When I have to start buying really, really, high-end storage to make this box consolidatable, I have a really big problem. Set your target right. Think of it like cycle scavenging, not achieving maximum utilization. When you start by saying "I want 100% CPU utilization," you start spending money in storage and networks to get there that you never needed to spend. That is a very bad bargain. ■

Mache Creeger (mache@creeger.com) is the principal of Emergent Technology Associates, marketing and business development consultants.

© 2008 ACM 0001-0782/08/1100 \$5.00



DOI:10.1145/1400214.1400230

We knew him as both scholar and friend.

BY MICHAEL STONEBRAKER AND DAVID J. DEWITT

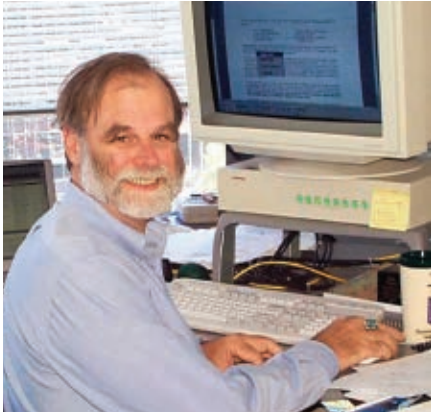
A Tribute to Jim Gray

JIM (JAMES NICHOLAS) GRAY was born January 12, 1944 and lost at sea off the coast of Northern California while sailing January 28, 2007. He was one of world's most distinguished computer scientists. His numerous contributions to the field of database systems were recognized through memberships in the National Academy of Sciences, the National Academy of Engineering, the American Academy of Arts and Sciences, and the European Academy of Science. He was also a fellow of both ACM and IEEE. Jim was awarded the 1998 ACM A.M. Turing Award for his seminal contributions to our understanding of the concept of transactions and their implementation.

At a tribute event at the University of California, Berkeley, last May 31, 700 of Jim's friends, family, and colleagues met to discuss both his professional accomplishments and the effect he had on their lives. Speaker after speaker discussed what he did in transaction processing and science applications, as well as the ways he had been a friend, mentor, and research collaborator to all.

Jim's pioneering research on transactions at IBM in the 1970s is the foundation for today's world of

e-commerce. Every time we use an ATM, reserve a seat on an airplane, or purchase an item on the Web, we are relying on the mechanisms Jim first developed 30-odd years ago. These techniques ensure that the “right” thing always happens—even in the presence of software and hardware failure. While they seem second-nature to us today, when Jim conceived them



they required very deep insight into the complexities of concurrently executing queries and updates against a shared database system.

Later in his career, Jim became interested in helping natural scientists with their work. He pioneered putting astronomy observation data into a database system. In this way scientists could query their data in SQL, rather than having to write custom programs in C++ or some other general-purpose language. Implementation of this idea for the Sloan Digital Sky Survey (www.sdss.org/) has resulted in more than 2,000 astronomy publications based on querying this data set through SQL.

Jim received his bachelor’s and Ph.D. degrees from the University of California, Berkeley, in 1966 and 1969, respec-



tively. Soon after receiving his Ph.D. he joined the IBM San Jose Research Laboratory (now known as the IBM Almaden Research Center) where he helped lead the design and development of System R, one of the first database systems to use the relational data model. In 1988, System R (along with INGRES, for the Interactive Graphics REtrieval System, project at Berkeley) was honored with the ACM Software Systems Award for pioneering development of relational database systems. It was as part of the System R project that Jim first developed the notion of what it means for transactions to be “serializable”; that is, they produce the same outcome as the serial ordering of the transactions. He also developed the connection between serializability and database consistency and how a simple protocol known as “two-phase locking” could be used to ensure that two or more transactions are serializable with respect to each other without the user having to understand the semantics of the transactions.

From the time he left IBM in 1980 to his joining Microsoft in 1995, Jim worked for Tandem Computers (1980–1990) on the parallel relational database system Non-Stop SQL and at Digital Equipment Corporation (1990–1995). Over the course of his career Jim also made numerous technical contributions beyond his work on transactions, including database system architectures and algorithms, fault tolerance, input/output architectures, parallel database systems, database system performance evaluation and benchmarking, multidimensional data analysis, and e-science, including the TerraServer (www.terraserver-usa.com) and the Sloan Digital Sky Survey project. When he disappeared at sea in 2007, he held the title of Technical Fellow at Microsoft.

That disappearance spurred the computer science community to action, and a massive amateur search effort was pulled together to augment the professional one launched by the U.S. Coast Guard. This effort entailed retargeting satellites to sweep the region of interest and posting the imagery on the Amazon Mechanical Turk site (www.mturk.com) so the distributed community could examine it in parallel to look for his sailboat, *Tenacious*. Possible sightings were then examined by experts in image rec-



ognition. It is the hope of the community that this imagery workflow will be automated and performed in real time during future searches. Parallel efforts searched for wreckage along the entire length of the California coastline and posted flyers at every marina in California. No trace of Jim’s boat was ever found. An extensive underwater search was equally unsuccessful. Hence, it is



likely that we will never know what happened to *Tenacious*, and the loss of Jim Gray will remain a mystery.

I (Michael) first met Jim while I was a struggling assistant professor at Berkeley in 1971. He was instrumental in helping me do the research that led to my first publication, which dealt with a simplification of Jay Forrester’s model of an urban area. I am forever grateful for his help motivating me in the publish-or-perish world of an assistant professor.

Jim was obviously brilliant, as anyone who talked to him quickly realized. However, he also read widely and knew a lot about a lot of things. In fact, he is one of the few people I have found to be intellectually intimidating. Moreover, he was always willing to read papers that other researchers sent him and

PHOTOGRAPHS (FROM UPPER LEFT) COURTESY OF ALEXANDER SZALAY, JOEL BARTLETT, DONNA DARNES



offer insightful comments. I routinely sent him my work in draft form and was always amazed by the breadth of knowledge reflected in his comments. They usually took the form: “Have you looked at System XYZ?; the people behind it looked at the problem you are considering.” XYZ would, of course, be an effort I had never heard of.

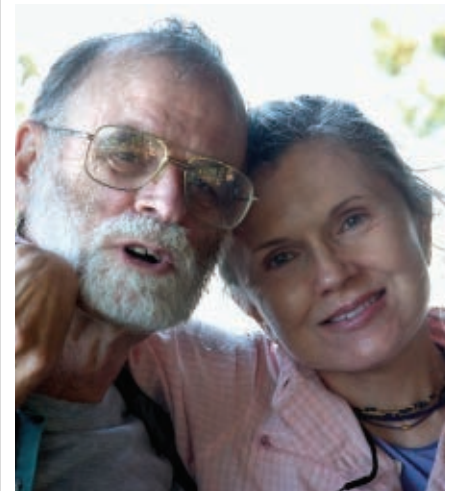
Jim was a mentor to many of the younger people in computer science and traveled widely to universities and research centers to interact with researchers. He was always willing to give service to the field. I remember vividly the creation in 2003 of the Conference on Innovative Data Systems Research (CIDR, www.cidrdb.org). We (Michael and David) became frustrated that SIGMOD routinely turned down our practical papers. Reaching the boiling point, we asked Jim to help start a new conference as a venue for such work, and Jim, as always, was willing to help. Moreover, the night before the opening session of the first CIDR conference in Asilomar, CA, we realized we did not have a data projector for showing PowerPoint slides. Rather than risk compromising the success of the conference, Jim made the five-hour round trip back to San Francisco to get a projector, returning to the Asilomar Conference Center at 3 A.M. That was Jim.



PHOTOGRAPH COURTESY OF JOEL BARTLETT

He was an unmanageable free spirit in the workplace who could write prodigious amounts of code and even more prodigious research reports.

Anecdotes reflecting his special character are legendary. He refused to conform to social norms; we never saw him wearing a coat and tie. He was an unmanageable free spirit in the workplace who could write prodigious amounts of code and even more prodigious research reports. It was reported at the Tribute that he had asked IBM to transfer him from its Thomas J. Watson Jr. Research Laboratory in Yorktown, NY, to its San Jose Research Laboratory in California to work on System R. When his boss refused, Jim quit on the spot and drove cross-country to be hired by the San Jose Lab. He loved to take people sailing on his boat, and it seems as if half the database community had this pleasure. Equally legendary are anecdotes of his backpacking and hiking trips in the Sierras.



Jim was a true scholar and friend. We will forever try to live up to the standard he set by his behavior. We can speak on behalf of the entire computer science community that we miss this mountain of a man every day. Our hearts and thoughts go out to his wife, Donna, his daughter, Heather, and his sister, Gail, who must deal with the ambiguous loss of Jim up close and personal. **□**

Michael Stonebraker (stonebraker@csail.mit.edu) is an adjunct professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology, Cambridge, MA, and the chief technology officer of Vertica Systems, Inc., and Blyedge Corp.

David J. DeWitt (dewitt@microsoft.com) is a technical fellow in the Microsoft Jim Gray Systems Lab, Madison, WI, and the John P. Morgridge Professor, Emeritus, in the Computer Sciences Department of the University of Wisconsin, Madison.

© 2008 ACM 0001-0782/08/1100 \$5.00



Figure 1: Jim Gray and the Sloan Digital Sky Survey telescope, Apache Point, NM.

DOI:10.1145/1400214.1400231

How he helped develop the SkyServer, delivering computation directly to terabytes of astronomical data.

BY ALEXANDER S. SZALAY

Jim Gray, Astronomer

JIM GRAY WORKED with astronomers for more than a decade, right up to the time he went missing in 2007. My collaboration with him created some of the world's largest astronomy databases and enabled us to test many unorthodox data-management ideas in practice. The astronomers collaborating with us have continued to be very receptive to them, embracing Jim as a card-carrying member of their community. Jim's contributions have left a permanent mark on astronomy worldwide, as well as on e-science in general.

Astronomy data has doubled in size every year for the past 20 years, due mostly to the emergence of electronic sensors. The largest sky survey of the past decade, the Sloan Digital Sky Survey, or SDSS (www.sdss.org), is often called the cosmic genome project. When it began in 1992, the size of the data set to be used for scientific analysis was measured in terabytes, shockingly large for the time. My group at Johns Hopkins University was selected by the SDSS Collaboration to build the science archive for the

SDSS, a task we quickly realized would require a powerful search engine with spatial search capabilities. Our experimental system, based on object-oriented technologies, was good enough to develop an understanding of how the eventual system should function, though we knew we would also need to do something different, most notably in terms of query performance.

One SDSS collaboration meeting in the mid-1990s took me to Seattle where I had dinner with Charles Simonyi, then at Microsoft, who recognized the similarities between our problem and the Microsoft TerraServer (www.terra-server.com), which provides free online access to U.S. Geological Survey digital aerial photographs, and immediately called Jim to arrange a meeting. A few weeks later I flew to San Francisco and visited him at the Bay Area Research Center. Thus began a lively discussion about the TerraServer, how it could be turned inside out for a new (astronomical) purpose, and how spatial searches over the Earth were both similar to and different from spatial searches over the sky. We spent a full day dissecting the problem.

Jim asked about our "20 queries," his incisive way of learning about an application, as a deceptively simple way to jump-start a dialogue between him (a database expert) and me (an astronomer or any scientist). Jim said, "Give me your 20 most important questions you would like to ask of your data system and I will design the system for you." It was amazing to watch how well this simple heuristic approach, combined with Jim's imagination, worked to produce quick results.

Jim then came to Baltimore to look over our computer room and within 30 seconds declared, with a grin, we had the wrong database layout. My colleagues and I were stunned. Jim explained later that he listened to the sounds the machines were making as they operated; the disks rattled too much, telling him there was too much random disk access. We began mapping SDSS database hardware re-


quirements, projecting that in order to achieve acceptable performance with a 1TB data set we would need a GB/sec sequential read speed from the disks, translating to about 20 servers at the time. Jim was a firm believer in using “bricks,” or the cheapest, simplest building blocks money could buy. We started experimenting with low-level disk IO on our inexpensive Dell servers, and our disks were soon much quieter and performing more efficiently.

Astronomy and the SkyServer


Toward the end of 2000 data started arriving from the SDSS telescope in Apache Point, NM (see Figure 1), and Jim said simply, “Let’s get to work.” So during Christmas and the New Year’s holiday we converted the whole object-oriented database schema to a Microsoft SQL Server-based schema. We modified many of our loading scripts by looking at Tom Barclay’s TerraServer code and soon, with Jim’s guidance, had a simple SQL Server version of the SDSS database.²⁴

The SDSS project was at first reluctant to even consider switching technologies, so for about a year the SQL Server database we had designed was a “cowboy” implementation, not part of the official SDSS data release. Coincidentally, Intel gave us a pool of servers to use to experiment with the database, giving us a show-and-tell meeting in San Francisco a few months after the first bits of data started to come in from the telescope. We decided to create a simple graphical interface on top of the database, similar to the one on the TerraServer, to enable astronomers and anyone else to visually browse the sky. My son, Tamas (13 at the time) came along for the Intel meeting and helped man the booth, telling us, “No self-respecting schoolkid would use such an interface,” that it had to be much more visually stimulating and interactive.

Jim gave one of his characteristically big laughs; we then looked at one another and realized we had our target audience. Even if astronomers were not ready, we would design a database and integrated Web site for schoolchildren. This was the moment we set out to build the SkyServer to connect the database to the pixels in the sky. The name was an obvious play on Ter-



We soon had the framework and the ability to load hundreds of GB of data in a reasonable amount of time, marking the transition of the SkyServer team from “cowboys” to “ranchers.”



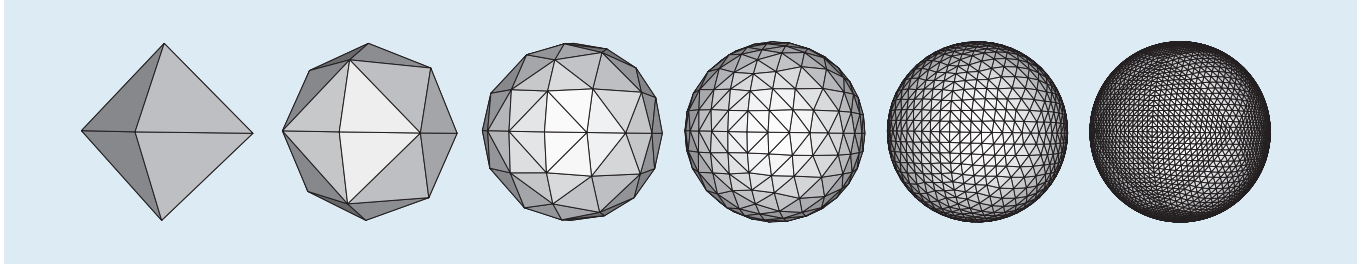
raServer, and we pitched it to the SDSS project as a tool for education and outreach, as well as for serious scientific investigation. When the first batch of SDSS data was officially declared public in 2001, the SkyServer, then running on computers donated by Compaq, appeared side by side with the official database for astronomers. We wrote simple scripts to create false-color images from the raw astronomy data and adopted the TerraServer scheme to build an image pyramid consisting of successive sets of tiles at different magnifications.

By the next year (2002), everyone realized that the SkyServer engine was much more robust and scalable than expected. Ani Thakar, a research scientist at Johns Hopkins, made a superhuman effort to convert the whole existing framework to SQL Server.²² Jim insisted on “two-phase loading,” that is, we would load each new batch of data into its own separate little database, then run data-cleaning code and accept the data only if it passed all the tests. This foresight turned out to be enormously useful; once the data started coming through the hose, we could recover from errors (there were lots of them) much more easily. We soon had the framework and the ability to load hundreds of GB of data in a reasonable amount of time, marking the transition of the SkyServer team from cowboys to “ranchers.”

Curtis Wong, manager of the Microsoft Next Media Research Group, then redesigned the SkyServer’s interface. His seemingly minor modifications of our style sheets had a huge effect on the entire site’s look and feel; it suddenly came alive. Many volunteers, including former Johns Hopkins student Steve Landy and physics teacher Rob Sparks, helped add content. Jordan Raddick, a science writer, created a new section of the Web site, with educational exercises and formal class materials for all students, from kindergarten to high school. Professional astronomers also appreciated the power of the visual tools, and the site quickly became popular, even in this community.

The next major step came with the emergence of Microsoft’s .NET Web services. Jim invited our development team (at Johns Hopkins) to San Francisco to the VSLive Conference (Janu-

Figure 2: The hierarchical subdivision of the sphere that forms the basis of the Hierarchical Triangular Mesh, starting with an octahedron.



ary 2002) where .NET was introduced and where our students entered a worldwide .NET programming contest, eventually coming in second. They created a set of services—called SkyQuery3—that performed queries across geographically separate databases. At the same time, Jim built a prototype for the ImageCutout, a Web service building dynamic image mosaics, that became the core of the next-generation user interfaces we developed for the SkyServer to integrate images and database content.¹⁹

Later, during a six-month sabbatical, Jim picked up a few astronomy textbooks, took them along on his sailboat, *Tenacious*, and while sailing quickly turned into a “native astronomer,” understanding the important concepts of astronomy. He thus enabled himself to participate in the reformulation of research ideas into elegant SQL, working with us side-by-side not only on database-related problems but on major-league astronomy research. We subsequently wrote many papers together where his ideas were quite relevant to astronomy.¹⁸ At the same time he taught us database design and computer science and invited several of our students to be interns at BARC.

As Jim spent more and more of his time in astronomy, he noted on one of his famous PowerPoint slides concerning relational database design: “I love working with astronomers, since their data is worthless.” He meant it in the most complimentary sense, that the data could be freely distributed and shared, since there were no financial implications or legal constraints. He went on to participate in many SDSS meetings, becoming a much-beloved and highly respected member of the astronomical community. His contributions are indeed very much appreciated, and in recognition of his work an asteroid is about to be named for

him by the International Astronomical Union.

Soon after the SkyServer was launched in 2001, it was obvious that astronomers would want to perform a variety of spatial searches for objects in the sky. The survey also had a rather complex geometry, and in order to describe it we would need an extensive framework for spatial operations. Over the next few years (2002–2006), with several of my students and postdocs (particularly Peter Kunszt) we wrote, again with Jim’s guidance, a fast package for spatial searches called Hierarchical Triangular Mesh (see Figure 2).¹⁷ We also built an interface to SQL Server and were soon performing blazingly fast searches over the sky. This emerged as one of the most notable features of the SkyServer. The tools eventually also made it into the shrink-wrap package of SQL Server 2005 as a demo on how to interface SQL to external software.^{4,8}

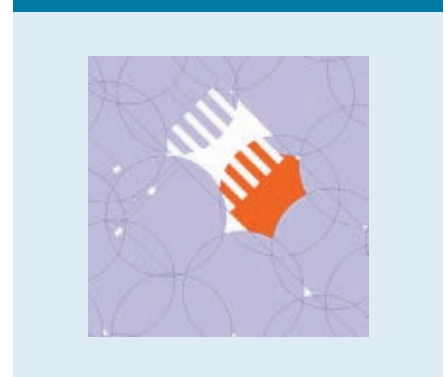
Jim was excited about these spatial computations, since they demonstrated one of his main convictions: that when you have lots of data, you take the computations to the data rather than the data to the computations. To Jim, there is nothing closer to the data than the database; thus the computations have to be done inside the database.⁹

As spatial searches grew in complexity, it became apparent that we would need even more extensive processing capabilities. Besides indexed searches, we needed better ways to represent complex polygons on the sphere. We ended up combining two complementary approaches. In one representation, polygons were represented as intersections of the unit sphere with a 3D polyhedral. The polyhedral was delimited by planes, so each convex polyhedron could be built from the intersection of a set of these half-spaces. This turned out to be handy for testing a point against a polygon in SQL. The inside

test focused on the dot-product of the Cartesian vector describing the point with the normal vector of the half-space against the distance of the plane from the origin.⁹ The dual representation (in terms of arcs) formed the outlines of the polygons (see Figure 3). We built tools to perform the set algebra of spherical polygons, including morphological operations over the sphere, a complex computational geometry library, all in SQL. I can think of no other person who would have thought of such an idea, much less been able to implement it. The library was subsequently converted to C# by Tamas Budavari and George Fekete of John Hopkins, though much of the code in SkyServer remains Jim’s original.

Jim realized there are two different types of spatial problems: one related to a localized, relatively small region, the other to a fuzzy (probabilistic) spatial join over much of the celestial sphere. He came up with the idea of using latitude zones and wrote the whole query, joining two tables with hundreds of millions of rows, as a single SQL statement, letting the optimizer do its magic in terms of parallelizing

Figure 3: Typical spherical polygon (orange) describing an area of uniform target selection for follow-up spectroscopic observation in the Sloan Digital Sky Survey arising from the intersection of geometries in the survey area.



the join¹⁰—one of the finest examples of SQL wizardry I have ever seen. He worked with Maria Nieto-Santisteban of Johns Hopkins to create parallel implementations of this cross-matching operation across many servers; performance is nothing short of stunning.^{6,12} These ideas are the basis of the next-generation SkyQuery engine we are building today.

It was around 2001 that astronomers began to explore the idea of a U.S. National Virtual Observatory (www.usvo.org/).^{10,21} Given the fact that most of the world’s astronomy data is public (“worthless”) and online, the time seemed right to develop a framework where all of it would appear as part of a single system. Jim was an enthusiastic supporter of the idea and an active participant in all the discussions about its design. His ideas are still at the heart of its service-based architecture. His advice helped us avoid many computational and design pitfalls we would undoubtedly have fallen into. He helped many different groups from around the world bring their data into databases; his astronomy collaborators are found everywhere, from Edinburgh to Beijing, Pasadena, Munich, and Budapest. He bought several sneakernet boxes, inexpensive servers that travel the world as an inexpensive way to transport data, and was highly

amused by the fact that in spite of the delays due to postal services and customs checks the bandwidth still exceeds that of the scientific world’s high-speed networks.¹¹

The SkyServer also turned out to be a groundbreaking exercise in publishing and curating digital scientific data. We learned that once a data set is released, it cannot be changed and must be treated like an edition of a printed book, in the sense that one would not destroy an old copy just because a new one appears on the shelves. To date, we carry forward all the old releases of SDSS data.

We also aimed to capture all relevant information in the database. We created a framework for automatically supporting physical units and descriptions by the database, using markup tags in the comments of our SQL scripts. We recently (2008) archived all email sent during the project in a free-text searchable database.

We were indeed anxious to see how scientists would interact with the database. Analyses, we knew, must be done as close to the data as possible, but it is also difficult to allow general users to create and run their own functions inside a shared, public database. Nolan Li, a graduate student at Johns Hopkins, and Wil O’Mullane, a senior programmer in the Johns Hopkins

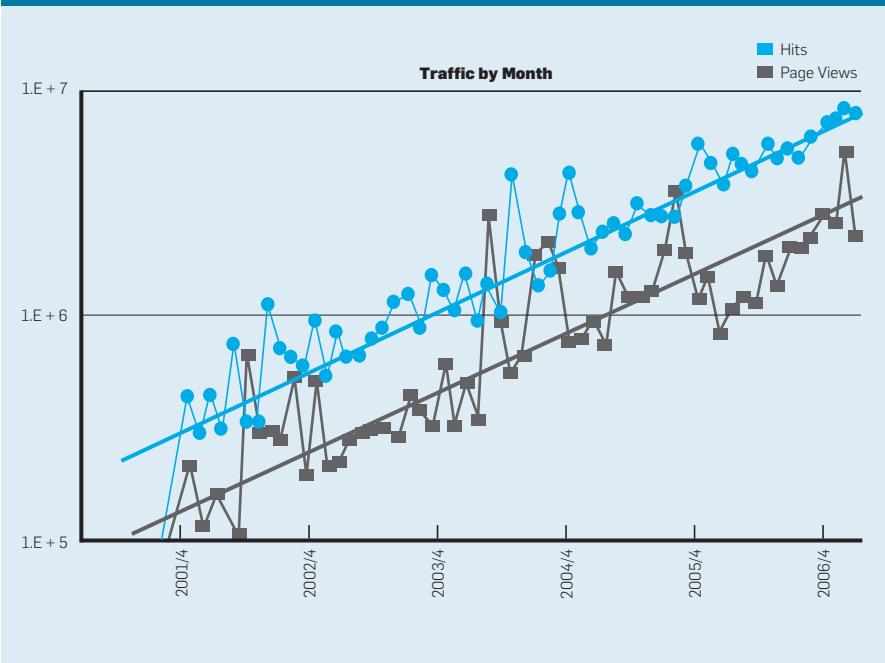
SDSS group, proposed giving users their own serverside databases (called MyDB/CasJobs) where they could do anything yet still link to the main database as well. Jim embraced the idea and was instrumental in turning it into generic dataspace.¹³

Over the years, we also noticed another interesting user pattern. Even though the MyDB interface gave users who wanted to run long jobs a way around our five-minute timeouts for anonymous queries, many astronomers and non-astronomers alike were writing Python and Perl crawlers where a simple query template was repeatedly submitted with a different set of parameters, occasionally leading to problems.

In one case someone was submitting a query every 10 seconds that was less than optimally written and so took more than 10 seconds to execute. As a result, the requests kept piling up, and the server became extremely overloaded. As we noted this odd behavior and identified and isolated the “guilty” query, Jim quickly modified the stored procedure that executed the user-written free-form SQL queries. He put in a statement conditional to the IP address of the user running the particular robot script, so, for that user alone, the query would not be executed but instead give the message: “Please contact Jim Gray at the following email address:...” The queries stopped immediately. We later learned they were coming from a CS graduate student in Tokyo who had the shock of his life from Jim’s email, which (for a student of CS) must have sounded like the voice of God. Jim followed up and sent the student an email that said: “It is OK to use the system and OK to send an email.”

We logged all traffic from day one and were amazed to see how it grew (see Figure 4) and how a *New York Times* article on a new SDSS result caused a huge spike in user traffic. It was gratifying to see that afterward the traffic continued to stay higher than before, indicating that many people, astronomers and non-astronomers alike, liked what they saw. Our analysis of SkyServer traffic found that most of the one million users were non-astronomers and that there is a power law with no obvious breaks in any of the

Figure 4: Aggregate SkyServer monthly traffic 2001–2006 when the number of Web hits doubled each year.



usage statistics.¹⁵ Jim liked to say: “You have nothing to fear but success or failure” and, of course, he never intended to fail at anything.

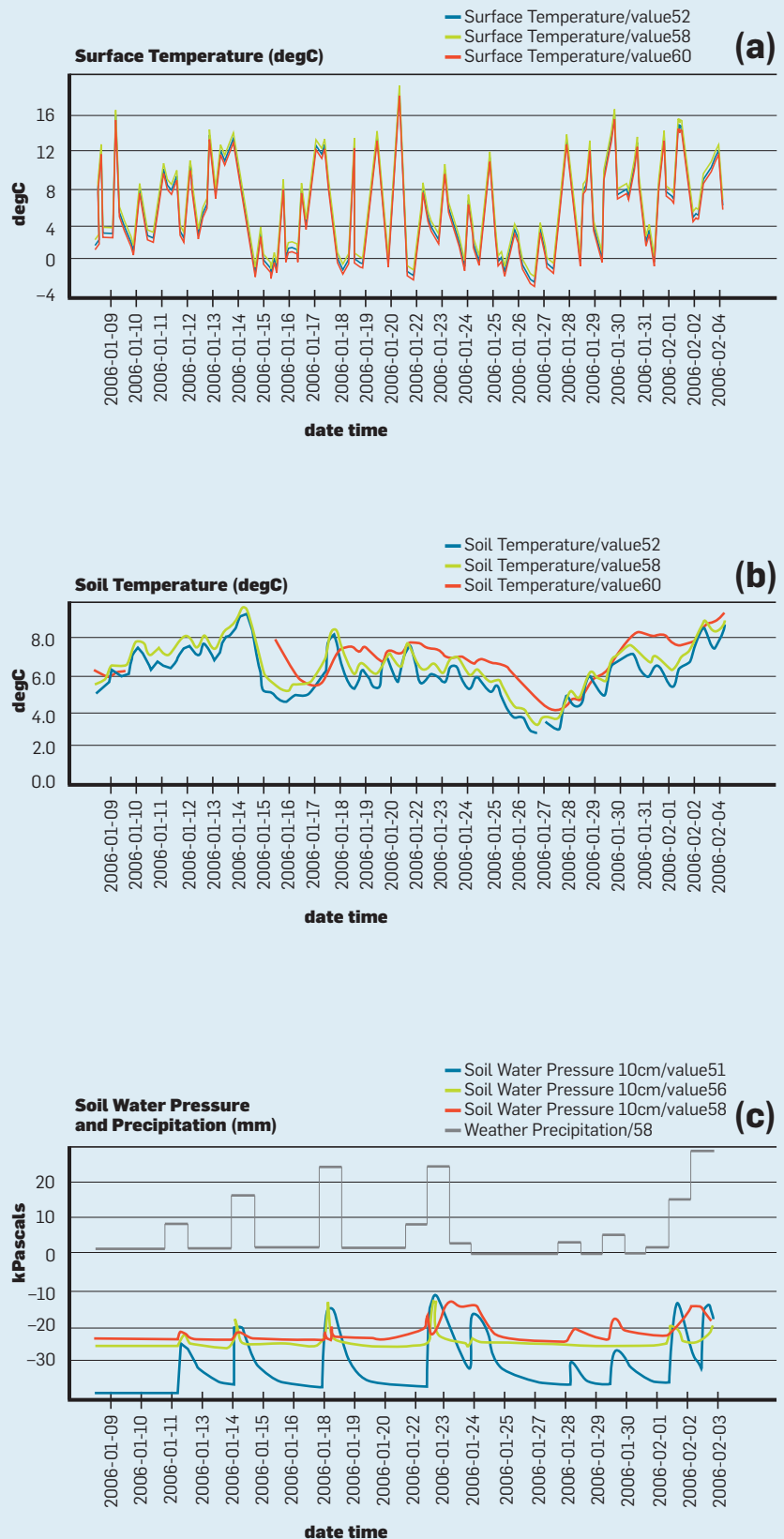
Beyond Astronomy

It became clear from our SkyServer experience that virtual observatories are sure to emerge on every scale of the physical world, from high-energy physics to nanotech, molecular biology, environmental observatories, planet Earth, even the entire universe. Many of the unknown issues related to managing huge amounts of data are common to all disciplines and revolve around our human, as well as our digital, inability to deal with increasing amounts of data.⁷

As a result we’ve been considering the broader implications of our Sky-Server work. The SDSS marked a transition to a new kind of science. Science itself has evolved over the centuries, from empirical to analytic, then to computational-X, where X represents many (if not all) scientific disciplines. With the emergence of large experiments like SDSS, where even data collection is via computer, a paradigm shift is under way. We are entering an era where there is so much data that the brute-force application of computational hardware is not enough to collect and analyze it all. We need to approach even the design of our experiments differently, taking an algorithmic perspective. Data management and enormous databases are inevitable in this new world, where business is e-business and science is e-science.¹⁶

SDSS data represents a wonderful opportunity to explore and experiment with how scientists adopt to new tools and new technologies. In the same spirit, Jim experimented with how tools and technologies carry over to other disciplines. For example, he consciously started (beginning in 2005) to develop relationships with molecular biologists and genomics researchers. I went along for some of his visits to the Whitehead Institute for Biomedical Research at MIT (www.wi.mit.edu) and the National Center for Biotechnology Information (www.ncbi.nlm.nih.gov/) and was amazed to find how similar many of the bioinformatics challenges were to those in astronomy. It was great to see Jim go native in biology with the same

Figure 5: Charts of sensor measurements generated from the On Line Analytical Processing data-cube for sensor deployment at Johns Hopkins University.




comfort he did in astronomy and how his “20 queries” cut through the communication gap in the various communities. The same thing happened when he started to work with oceanographers from the Monterey Bay Aquarium Research Institute (www.mbari.org/) and the North-East Pacific Time-Series Undersea Networked Experiments project (www.neptuneproject.org/).


He was among the first computer scientists to realize how the data explosion changes not only science but scientific computing as well. As the amount of data grows faster than our ability to transfer it through the network, the only solution that promises to keep up is to take the computation directly to the data.²⁰ This principle contrasts with recent trends in high-performance computing where the machines are increasingly CPU-intensive, while the ability to read and write data lags behind processing speed. Lively discussions with Jim and Gordon Bell of Microsoft Research about this problem resulted in a paper outlining what is wrong with today’s computing architectures²; I am immensely proud of having been a co-author. Our group at Johns Hopkins is now implementing the vision we outlined there, building a machine—called in Jim’s honor the GrayWulf (graywulf.org/)—specially tailored for data-intensive computations.

We realized that the data explosion in astronomy is due to the electronic charge-coupled device detectors that have replaced photographic plates. As semiconductor manufacturing matured, each year has brought a new generation of bigger and more sensitive detectors that could be replaced without affecting the telescopes themselves. Much as gene chips and gene sequencers have industrialized molecular biology, the revolution in Earth-observing satellite imagery has also been the result of better imaging devices. The common theme is that whenever an inexpensive sensing device is on an exponential growth path, a scientific revolution is imminent.

Such a revolution is taking place today with inexpensive wireless sensor networks, sometimes called “smart dust” after the University of California, Berkeley, project that first developed them almost a decade ago.



To Jim, there is nothing closer to the data than the database; thus the computations have to be done inside the database.⁹



It is expected that within the next five years there will be more sensors online than computers worldwide. Intel’s Berkeley Lab was among the first to develop such devices. My wife, Kathy Szlávecz, is a soil biologist interested in the soil ecosystem and has for years painstakingly sought and collected data involving environmental parameters. Jim connected her to the Berkeley lab, and after her seminar we came away with a shoebox full of Berkeley Motes (www.eecs.berkeley.edu/departement/EECSbrochure/c6-s1.html). At the same time Johns Hopkins hired Andreas Terzis, a computer scientist specializing in wireless sensors, and thus a new collaboration (lifeunderyourfeet.org/) was formed. Despite having only a shoestring budget, it still managed to build a small sensor network to study soil moisture and temperature.

Jim realized that in this field of enviro-sensor networks, almost everyone focuses on the first phase of the problem—collecting data. In astronomy we have learned the hard way that with exponential data growth one should worry about data processing and analysis even at the beginning; otherwise, it will be difficult to catch up once the data stream really opens up.¹

He was also very interested in the flexibility of the SkyServer framework. Another aspect of the environmental work is how interested scientists are in long-term trends and averages, even as they want to retain all the raw data and dive in whenever they find something unusual. We again went to work, converting in a matter of weeks the SkyServer framework into an end-to-end system to handle data from environmental science.²³ We wrote code to handle time-series and in-database calibrations. Soon, we had help from Stuart Ozer from Microsoft Research who built an OLAP data cube for the sensor data, the first ever (as far as we know) in a scientific application (see Figure 5).¹⁴

Collaborator and Friend

Over the years, as our collaboration intensified, our work days would start with Jim’s phone calls while he walked from home to BARC, followed by back-and-forth calls until early morning on the east coast (of the U.S.). Very often

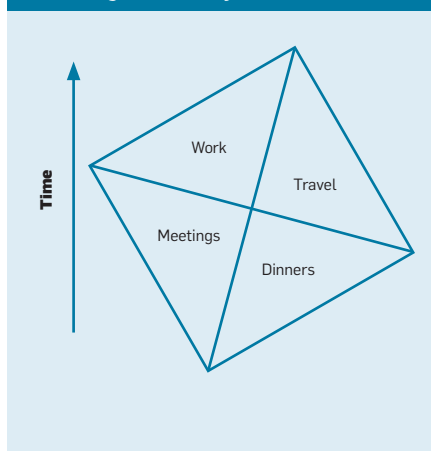
we were still talking at 3 A.M. my time or 7 A.M. his time. We spent a lot of time together, chasing bugs and arguing over code.

Jim had an uncanny ability to go for the jugular, recognizing the critical issue or bottleneck. I had the privilege of meeting some of the top physicists of the 20th century, including Richard Feynman and Yakov Zeldovich. Jim's mind worked the same way as theirs; like them, he could solve a problem on the back of an envelope.

He was also very good at getting results published. When he felt the time was right for us to write a paper, he would start with a quick draft, helping his collaborators (like me) with writer's block get up to speed. He was very generous, often doing much more than his collaborators, yet still insisted on others, particularly young researchers, take the role of lead author. He mentored many students, always patient and encouraging, trying to get them excited and lead by example.

He never gave up hands-on work. If he did not have time to write code or tinker with databases, it was not a good day for him. He had an inexhaustible source of energy; when everyone else was falling over, he kept going, pulling everyone along with him. Starting at 7 A.M. one day at BARC we kept going at a spurious SQL bug and never stood up (except for coffee) until 11 P.M. (when we finally found it). By then, most restaurants were closed, but Jim led the way through San Francisco's North Beach neighborhood until we found an inviting Italian restaurant and had a proper dinner.

Figure 6: The world of science according to Jim Gray.



He once took a piece of paper and drew a rectangle for me, with two diagonals splitting it into four pieces, then said: "This is our life" with the vertical axis representing the arrow of time (see Figure 6) and asked: "Alex, where are we on this diagram?" He did all he could to ensure the work part occupied as big a piece as possible.

Jim and I met rather late in our careers, a point in life when most people might perhaps be expected to establish good working relationships, but deep friendships generally come much earlier. For whatever reason, we connected, became close friends and had amazing conversations where the bandwidth regularly went way beyond the spoken word.

Jim's love of life and work inspired anyone fortunate enough to spend time with him. My friendship and collaboration took my career in new, entirely unexpected directions, away from astrophysics and into e-science. He affected the lives of many others in the same way. All of us privileged enough to call Jim a friend will forever be trying to turn at least some of the projects we dreamed of together into reality.

Acknowledgments

I thank Jim Gray for many years of advice, support, and friendship and Donna Carnes for being the strongest of all, holding us together when everything was falling apart. C

References

1. Balazinska, M., Deshpande, A., Franklin, M.J., Gibbons, P.B., Gray, J., Hansen, M., Liebholt, M., Nath, S., Szalay, A., and Tao, V. Data management in the worldwide sensor Web. *IEEE Pervasive Computing* (2007), 30.
2. Bell, G., Gray, J., and Szalay, A.S. Petascale computational systems. *IEEE Computer* (Jan. 2006), 39.
3. Budavári, T., Malik, T., Szalay, A.S., Thakar, A., and Gray, J. SkyQuery: A prototype distributed query Web service for the Virtual Observatory. In *Proceedings of the ADASS XII, ASP Conference Series*, H. Payne, R.I. Jedrzejewski, and R.N. Hook, Eds. (Baltimore, MD, Oct. 2002). Astronomical Society of the Pacific, San Francisco, 2003, 31.
4. Fekete, G., Szalay, A.S., and Gray, J. HTM2: Spatial toolkit for the Virtual Observatory. In *Proceedings of the ADASS, ASP Conference Series* (Strasbourg, France, Oct. 2003). Astronomical Society of the Pacific, San Francisco, 2003, 289.
5. Gray, J., Nieto-Santisteban, M.A., and Szalay, A.S. *The Zones Algorithm for Finding Points-Near-a-Point or Cross-Matching Spatial Datasets*, MSR-TR-2006-52. Microsoft Technical Report, Redmond, WA, 2006.
6. Gray, J., Szalay, A., Budavári, T., Thakar, A.R., Nieto-Santisteban, M.A., and Lupton, R. *Cross-Matching Multiple Spatial Observations and Dealing with Missing Data*, Microsoft Technical Report, MSR-TR-2006-175. Microsoft Technical Report, Redmond, WA, 2006.
7. Gray, J., Liu, D.T., Nieto-Santisteban, M.A., Szalay, A.S., Heber, G., and DeWitt, D. *Scientific Data Management in the Coming Decade*, MSR-TR-2005-10. Microsoft Technical Report, Redmond, WA, 2005.
8. Gray, J., Szalay, A.S., and Fekete, G. *Using Table Valued*

- Functions in SQL Server 2005 to Implement a Spatial Data Library*, MSR-TR-2005-122. Microsoft Technical Report, Redmond, WA, 2005.
9. Gray, J., Szalay, A.S., Fekete, G., O'Mullane, W., Thakar, A.R., Heber, G., and Rots, A.H. *There Goes the Neighborhood: Relational Algebra for Spatial Data Search*, MSR-TR-2004-32. Microsoft Technical Report, Redmond, WA, 2004.
10. Gray, J. and Szalay, A.S. Where the rubber meets the sky: Bridging the gap between databases and science. *IEEE Data Engineering Bulletin* (Dec. 2004), 4.
11. Gray, J., Chong, W., Barclay, T., Szalay, A.S., and Vandenberg, J. *TeraScale SneakerNet: Using Inexpensive Disks for Backup, Archiving, and Data Exchange*, MS-TR-2002-54. Microsoft Technical Report, Redmond, WA, 2002.
12. Nieto-Santisteban, M.A., Thakar, A.R., Szalay, A.S., and Gray, J. Large-scale query and xmatch, entering the parallel zone. In *Proceedings of the Astronomical Data Analysis Software and Systems XV ASP Conference Series*, C. Gabriel, C. Arviset, D. Ponz, and E. Solano, Eds. (El Escorial, Spain, Oct. 2005). Astronomical Society of the Pacific, San Francisco, 2006, 493.
13. O'Mullane, W., Gray, J., Li, N., Budavari, T., Nieto Santisteban, M., and Szalay, A.S. Batch query system with interactive local storage for SDSS and the VO. In *Proceedings of the ADASS XIII, F. Ochsenbein, M. Allen, and D. Egret, Eds.* (Strasbourg, France, Oct. 2003). Astronomical Society of the Pacific, San Francisco, 2004, 372.
14. Ozer, S., Szalay, A.S., Szlavecz, K., Terzis, T., Musáoiu-E., R., and Cogan, J. *Using Data-Cubes in Science: An Example from Environmental Monitoring of the Soil Ecosystem*, MSR-TR-2006-134. Microsoft Technical Report, Redmond, WA, 2006.
15. Singh, V., Gray, J., Thakar, A.R., Szalay, A.S., Raddick, J., Boroski, B., Lebedeva, S., and Yanny, B. *SkyServer Traffic Report: The First Five Years*, MSR-TR-2006-190. Microsoft Technical Report, Redmond, WA, 2006.
16. Szalay, A.S. and Gray, J. Science in an exponential world. *Nature* 413 (2006), 440-441.
17. Szalay, A.S., Gray, J., Fekete, G., Kunszt, P., Kukol, P., and Thakar, A. *Indexing the Sphere with the Hierarchical Triangular Mesh*, MSR-TR-2005-123. Microsoft Technical Report, Redmond, WA, 2005.
18. Szalay, A.S., Budavári, T., Connolly, A.J., Gray, J., Matsubara, T., Pope, A. and Szapudi, I. Spatial clustering of galaxies in large data sets. In *Proceedings of the SPIE Conference on Advanced Telescope Technologies* (Waikaloa, HI, July), The International Society for Optical Engineering, 2002, 1-12.
19. Szalay, A.S., Budavári, T., Malik, T., Gray, J., and Thakar, A. Web services for the Virtual Observatory. In *Proceedings of the SPIE Conference on Advanced Telescope Technologies* (Waikaloa, HI, July), The International Society for Optical Engineering, 2002, 124.
20. Szalay, A.S., Gray, J., and Vandenberg, J. Petabyte-scale data mining: Dream or reality? In *Proceedings of the SPIE Conference on Advanced Telescope Technologies* (Waikaloa, HI, July), The International Society of Optical Engineering, 2002, 333-338.
21. Szalay, A.S. and Gray, J. The World-Wide Telescope. *Science* 293 (2001), 2037-2040.
22. Szalay, A.S., Kunszt, P., Thakar, A., Gray, J., Slutz, D., and Brunner, R. Designing and mining multi-terabyte astronomy archives: The Sloan Digital Sky Survey. In *Proceedings of the SIGMOD 2000 Conference* (Madison, WI). ACM Press, New York, 2000, 451-462.
23. Szlavecz, K., Terzis, A., Musáoiu-E., R., Cogan, J., Small, S., Ozer, S., Burns, R., Gray, J., and Szalay, A.S. *Life Under Your Feet: An End-to-End Soil Ecology Sensor Network, Database, Web Server, and Analysis Service*, MSR-TR-2006-90. Microsoft Technical Report, Redmond, WA, 2006.
24. Thakar, A., Szalay, A., Kunszt, P., and Gray, J. Migrating a multiterabyte archive from object to relational database. *Computing in Science and Engineering* 5 (Sept./Oct. 2003), 16.

Alexander S. Szalay (szalay@jhu.edu) is Alumni Centennial Professor in the Department of Physics and Astronomy at The Johns Hopkins University, Baltimore, MD.

DOI:10.1145/1400214.1400232

Internet-based data on human interaction connects scientific inquiry like never before.

BY JON KLEINBERG

The Convergence of Social and Technological Networks

The past decade has witnessed a coming-together of the technological networks that connect computers on the Internet and the social networks that have linked humans for millennia. Beyond the artifacts that have sprung from this development—sites such as Facebook, LinkedIn, MySpace, Wikipedia, digg, del.icio.us, YouTube, and flickr—there is a broader process at work, a growing pattern of movement through online spaces to form connections with others, build virtual communities, and engage in self-expression.

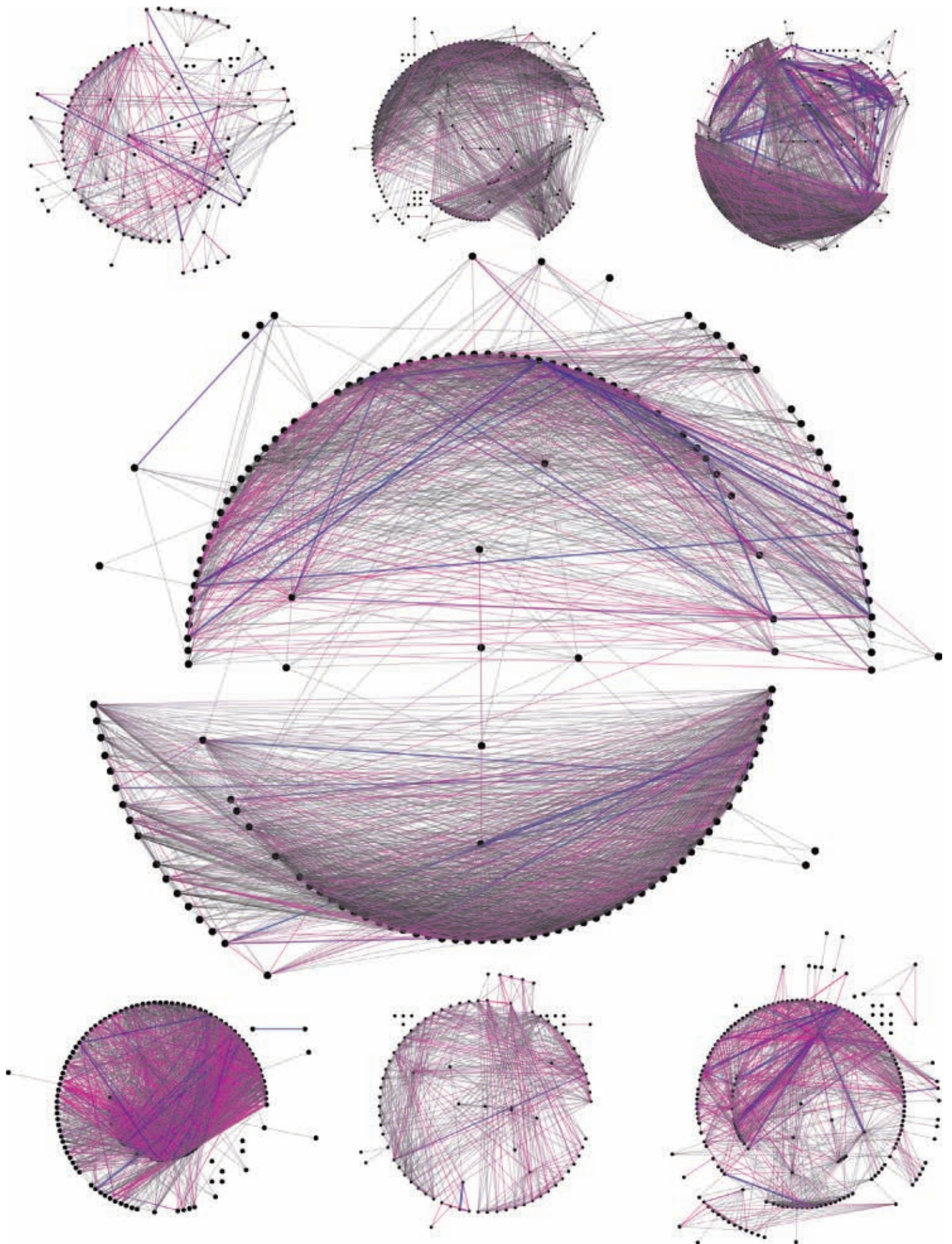
Even as these new media have led to changes in our styles of communication, they have also remained governed by longstanding principles of human social

interaction—principles that can now be observed and quantified at unprecedented levels of scale and resolution through the data being generated by these online worlds. Like time-lapse video or photographs through a microscope, these images of social networks offer glimpses of everyday life from an unconventional vantage point—images depicting phenomena such as the flow of information through an organization or the disintegration of a social group into rival factions. Science advances whenever we can take something that was once invisible and make it visible; and this is now taking place with regard to social networks and social processes.

Collecting social-network data has traditionally been hard work, requiring extensive contact with the group of people being studied; and, given the practical considerations, research efforts have generally been limited to groups of tens to hundreds of individuals. Social interaction in online settings, on the other hand, leaves extensive digital traces by its very nature. At the scales of tens of millions of individuals and minute-by-minute time granularity, we can replay and watch the ways in which people seek out connections and form friendships on a site like Facebook or how they coordinate with each other and engage in creative expression on sites like Wikipedia and flickr. We can observe a news story suddenly catching the attention of millions of readers or witness how looming clouds of controversy gather around a community of bloggers. These are part of the ephemeral dynamics of ordinary life, now made visible through their online manifestations. As such, we are witnessing a revolution in the measurement of collective human

The Nexus friend grapher application, created by Ivan Kozik, allows Facebook account holders to generate graphs illustrating their social network of friends. The resulting spheres not only demonstrate how friends are connected, but also indicate the interests shared by different groups of friends. For more information, or to create a graph, visit <http://nexus.ludios.net>.

FACEBOOK VISUALIZATION: NEXUS FRIEND GRAPHER (NEXUS.LUDIUS.NET) BY IVAN KOZIK



behavior and the beginnings of a new research area—one that analyzes and builds theories of large social systems by using their reflections in massive datasets.

This line of investigation represents a flow of ideas between computing and the social sciences that goes in both directions. Using datasets on collective human behavior, together with an algorithmic language for modeling social processes, we can begin to make progress on fundamental social-science questions, informed by a computational perspective. Meanwhile, social scientists' insights into these problems, which predate the Internet, are essential to understanding the current generation of computing systems. Indeed, most of the high-profile Internet applications to emerge over the past half-decade are governed not just by technological considerations but also by recurring and quantifiable principles of human social interaction; both technological and social forces, working together, shape the inherent operating constraints in such systems.

The resulting research questions arise from a coming-together of different styles of research, and it is important to recognize that analyses of truly massive social networks provide us with both more and less than we get from detailed studies at smaller scales. Massive datasets can allow us to see patterns that are genuine, yet literally invisible at smaller scales. But working at a large scale introduces its own difficulties. One doesn't necessarily know what any one particular individual or social connection signifies; and the friendships, opinions, and personal information that are revealed online come in varying degrees of reliability. One is observing social activity in aggregate, but at a fine-grained level the data is more difficult to interpret. The true challenge is to bridge this gap between the massive and the detailed, to find the points where these lines of research converge.

With that goal in mind, we discuss two settings where this research strategy is being pursued. We begin with the "small-world phenomenon" in social networks—the principle that we are all connected by short chains of acquaintances—and then look at the re-

lated problem of how ideas spread contagiously through groups of people.

The Small-World Phenomenon and Decentralized Search

When the playwright John Guare coined the term "six degrees of separation,"¹⁵ describing the notion that we are all just a few steps apart in the global social network, he was referring to a series of experiments performed by the social psychologist Stanley Milgram in the 1960s.³⁸ Milgram's work provided the first empirical evidence for this idea, and it is useful to consider the structure of his experiments and their significance.

Inspired in part by the work of the political scientist Ithiel de Sola Pool with the applied mathematician Manfred Kochen,⁷ Milgram asked each of a few hundred people in Boston and the Midwest to try directing a letter toward a designated "target" in the network—a stockbroker who lived in Sharon, Massachusetts. The participants in the experiment were given basic personal information about the target, including his address and occupation; but each participant could only mail the letter to someone he or she knew on a first-name basis, with the instructions to forward it on in this way toward the target as quickly as possible. The mail thus closed in on the town of Sharon, moving from friend to friend, with the successful letters reaching the target in a median of six steps.³⁸ This kind of experiment, constructing paths through social networks to distant target individuals, has been repeated by a number of other groups in subsequent decades.^{9, 12, 23}

Milgram's experiment and its follow-ups come with many caveats. In particular, they have tended to be most successful when the target is affluent and socially accessible; and even then, many chains fail to complete. Nevertheless, the striking fact at the heart of the results—that such short paths can be discovered in social networks—has been borne out by many subsequent analyses of large-scale network data. Quite recently, Leskovec and Horvitz built a social network from nearly a quarter-billion instant-messaging accounts on MSN Messenger, connecting two individuals if they engaged in a conversation over a one-month obser-

vation period.²⁶ The researchers found the average length of the shortest path between any two people on this system to be around 6.6—a number remarkably close to Milgram's, and obtained by utterly different means.

Modeling the Phenomenon. Mathematical models of this phenomenon start by asking *why* social networks should be so rich in short paths. In an influential 1998 paper, Watts and Strogatz sought to reconcile this abundance with the seemingly contrasting observation that the world is highly clustered, consisting of acquaintances who tend to be geographically and socially similar to one another.⁴⁰ They showed that adding even a small number of random social connections to a highly clustered network causes a rapid transition to a small world, with short paths appearing between most pairs of people. In other words, the world may look orderly and structured to each of us—with our friends and colleagues tending to know each other and have similar attributes—but a few unexpected links shortcutting through the network are sufficient to bring us all close together.


There is a further aspect to the Milgram experiment that is striking and inherently algorithmic: the experiment showed not just that the short paths were there but that people were able to find them.²⁰ When you ask someone in Omaha, Nebraska, as Milgram did, to use his or her social network to direct a letter halfway across the country to Sharon, Massachusetts, that person can't possibly know the precise course it will follow or whether it will even get there. The fact that so many of the letters zeroed in on the target suggests something powerful about the social network's ability to "funnel" information toward far-off destinations. The U.S. Postal Service does this when it delivers a letter, but it is centrally designed and maintained at considerable cost to do precisely this job; why should a social network, which has grown organically without any central control, be able to accomplish the same task with any reliability at all?

To begin modeling this phenomenon, suppose we all lived on a two-dimensional plane, spread out with a roughly uniform population density,


and that we each knew our next-door neighbors for some distance in each direction. Now, following Watts and Strogatz, we add a small number of random connections—say, each of us has a single additional friend chosen uniformly at random from the full population. Short paths appear, as expected, but one can prove that there is no procedure the people living in this world can perform—using only local information and without a global “bird’s-eye view” of the social network—to forward letters to faraway targets quickly.²⁰ In other words, in a structured world supplemented with purely random connections, the Milgram experiment would have failed: the short paths would have been there, but they would have been unfindable for people living in the network.

By extending things a little bit, however, we can get the model to capture the effect Milgram saw in real life. To do this, we keep everyone living on a two-dimensional plane but revisit the random connections, which are supposed to account for the unexpected far-flung friendships that make the world small. In reality, of course, these links are not completely random; they too are biased toward closer and more similar people. Suppose, then, that each person still has a random far-away friend, but that this friend is chosen with a probability that decays with the individual’s distance in the plane—say, by a “gravitational law” in which the probability of being friends with a person at a distance d decays as d^{-r} for some power r . Thus, as the exponent r increases, the world gets less purely random—the long-range friendships are still potentially far away, but overall they are more geographically clustered. What effect does this have on searching for targets in the network?

Analyzing this model, one finds that the effectiveness of Milgram-style search with local information initially gets better as r increases—because the world is becoming more orderly and easy to navigate—and then gets worse again as r continues increasing—because short paths actually start becoming too rare in the network. The best choice for the exponent r , when search is in fact very rapid, is to set it equal to 2. In other words, when the



A rumor, a political message, or a link to an online video—these are all examples of information that can spread from person to person, contagiously, in the style of an epidemic.



probability of friendship falls off like the square of the distance, we have a small world in which the paths are not only there but also can be found quickly by people operating without a global view.²⁰ The exponent of 2 is thus balanced at a point where short paths are abundant, but not so abundant as to be too disorganized to use.

Further analysis indicates that this best exponent in fact has a simple qualitative property that helps us understand its special role: when friendships fall off according to an inverse-square law in two dimensions, then on average people have about the same proportion of friends at each “scale of resolution”—at distances 1–10, 10–100, 100–1000, and so on. This property lets messages descend gradually through these distance scales, finding ways to get significantly closer to the target at each step and in this way completing short chains, just as Milgram observed.

Validating and Applying the Model. When such models were first proposed, it was unclear not only how accurate they were in real life but also how to go about collecting data to measure the accuracy. To do so, you would have to convince thousands of people to report where they lived and who their friends were—a daunting task.

But of course, the public profiles on social-networking sites readily do just that, and as these sites began to grow explosively in 2003 and 2004, Liben-Nowell et al. developed a framework for using this type of data to test the predictions of the small-world models.³⁰ In particular, they collected data from the friendship network of the public blogging site LiveJournal, focusing on half a million people who reported U.S. hometown locations and lists of friends on the site. They then had to extend the mathematical models to deal with the fact that real human population densities are highly nonuniform. To do so, they defined the distance between two people in an ordinal rather than absolute sense: they based the probability that a person v forms a link to a person w on the number of people who are closer to v than w is, rather than on the physical distance between v and w . Using this more flexible definition, the distribution of friendships in the data could

then in fact be closely approximated by the natural generalization of the inverse-square law.

It was difficult not to be a bit surprised by the alignment of theory and measurement. The abstract models were making very specific predictions about how friendships should depend on physical distance, and these predictions were being approximately borne out on data arising from real-world social networks. And there remains a mystery at the heart of these findings. While the fact that the distributions are so close does not necessarily imply the existence of an organizing mechanism (for example, see Bookstein⁵ for a discussion of this general issue in the context of social-science data), it is still natural to ask why real social networks have arranged themselves in a pattern of friendships across distance that is close to optimal for forwarding messages to faraway targets. Further, whatever the users of LiveJournal are doing, they are not explicitly trying to run versions of the Milgram experiment—if there are selective pressures driving the network toward this shape, they must be more implicit, and it remains a fascinating open question whether such forces exist and how they might operate.

Other research using online data has considered how friendship and communication depend on nongeographic notions of “distance.” For example, the probability that you know someone is affected by whether you and they have similar occupations, cultural backgrounds, or roles within a large organization. Adamic and Adar studied how communication depends on one such kind of distance: they measured how the rate of email messaging between employees of a corporate research lab fell off as they looked at people who were farther and farther apart in the organizational hierarchy.¹ Here too, this rate approximated an analogue of the inverse-square law—in a form adapted to hierarchies^{21, 39}—although the messages in the researchers’ data were skewed a bit more toward long-range contacts in the organization than short-range ones.

Finally, these models can rapidly turn into design principles for distributed computing systems as well. Modern peer-to-peer file-sharing systems

are built on the principle that there should not be a central index of the content being shared (in contrast, for example, to the way in which search engines like Google provide a central index for Web pages). As a result, looking up content in a peer-to-peer system follows a Milgram-style approach in which the hosts participating in the system must forward requests with only a local view.³¹ Mathematical models of small worlds—originally built with human networks in mind—can provide insights into the design of efficient solutions for this distributed search problem as well.

We’ve thus seen how viewing such models in the online domain can help us understand the global layout of social-networking sites, the flow of communications within organizations, and the design of peer-to-peer systems. We now look at how the insights we’ve gained here can provide perspective on an important related problem—the spread of information through large populations.

Social Contagion and the Spread of Ideas

Milgram’s experiment was about focusing a message on a particular target, but much of the information that flows through a social network radiates outward in many directions at once. A rumor, a political message, or a link to an online video—these are all examples of information that can spread from person to person, contagiously, in the style of an epidemic. This is an important process to understand because it is part of a broader pattern by which people influence one another over longer periods of time, whether in online or offline settings, to form new political and social beliefs, adopt new technologies, and change personal behavior—a process that sociologists refer to as the “diffusion of innovations.”³⁵ But while the outcomes of many of these processes are easily visible, their inner workings have remained elusive.

Some of the basic mathematical models for the diffusion of innovations posit that people’s adoption of new behaviors depends in a probabilistic way on the behaviors of their neighbors in the social network: as more and more of your friends buy a new product or join a new activity, you

are more likely to do so as well.¹³ Recent studies of online data have provided some of the first pictures of what this dependence looks like over large populations. In particular, Leskovec, Adamic, and Huberman studied how the probability of purchasing books, DVDs, and music from a large online retailer increased with the number of email recommendations a potential customer received.²⁵ Backstrom et al. determined the probability of joining groups in a large online community as a function of the number of friends who already belonged to the group.⁴ And Hill, Provost, and Volinsky¹⁶ analyzed how an individual’s adoption of a consumer telecommunications service plan depended on his or her connections to prior adopters of the service.


While the probability of adopting a behavior increases with the number of friends who have already adopted it, there is a “diminishing returns” pattern in which the marginal effect of each successive friend decreases.^{4,25} In many cases, however, an interesting deviation from this pattern is observed—a “0–1–2 effect,” in which the probability of joining an activity when two friends have done so is significantly more than twice the probability of joining when only one has done so.⁴

The structure of cascading behavior. Beyond these local mechanisms of social influence, it is instructive to trace out the overall patterns by which influence propagates through a large social network. In recent work, David Liben-Nowell and I investigated such global-scale processes by gathering data on chain-letter petitions that had spread widely over the Internet.²⁹ A particularly pervasive chain letter, which spread in 2002 and 2003, purported to organize opposition to the impending invasion of Iraq. Each copy of the petition contained the list of people who had received that particular copy, in the order in which they added their names and then passed it on to others in their email address books. In the process, several hundred of these copies had been sent to Internet mailing lists; by retrieving them from the mailing lists’ archives, we could reconstruct a large fragment of the branching tree-like trajectory by which the chain letter had spread.


The structure of the tree was surprising, as it challenged our small-world intuitions. Rather than fanning out widely, reaching many people with only a few degrees of separation, the chain letter spread in a deep and narrow pattern, with many paths consisting of several hundred steps. The short chains in the social network were still there, but the chain letter was getting to people by much more roundabout means. Moreover, we found a very similar structure for the one other large-scale chain letter on which we could find enough mailing-list data, this one claiming to be organizing support for National Public Radio.

Why this deep and narrow spreading pattern arises in multiple settings remains something of a mystery, but there are several hypotheses for reconciling it with the structure of a small world. In our work on chain letters, we analyzed a model based on the natural idea that people take widely varying amounts of time to act on messages as they arrive: when recipients forward the chain letter at different times to highly overlapping circles of friends, it can in effect “echo” through dense clusters in the social network, following a snaking path rather than a direct one. Simulations of this process on real social networks such as the one from LiveJournal produce tree structures very similar to the true one we observed.²⁹

It is also plausible that the nature of social influence—properties such as the 0–1–2 effect in particular—play an important role. Suppose that most people in the social network need to receive a copy of the letter at least twice before actually signing their name and sending it on. As Centola and Macy have recently argued, our long-range friendships may be much less useful for spreading information in situations such as these: you can learn of something the first time from a far-flung friend, but to get a second confirmatory hearing you may need to wait for the information also to arrive through your more local contacts.⁶ Such a pattern could slow down the progress of a chain letter, forcing it to slog through the dense structure of our local connections rather than exploit the long-range shortcuts that make the world small.



The availability of such rich and plentiful data on human interaction has closed an important feedback loop, allowing us to develop and evaluate models of social phenomena at large scales and to use these models in the design of new computing applications.



Contagion as a design principle. As with the decentralized search problem at the heart of the small-world phenomenon, the idea of contagion in networks has served as a design principle for a range of information systems. Early work in distributed computing proposed the notion of “epidemic algorithms,” in which information updates would be spread between hosts according to a probabilistic contagion rule.⁸ This has led to an active line of research, based on the fact that such algorithms can be highly robust and relatively simple to configure at each individual node.

More recently, contagion and cascading behavior have been employed in proposals for social computing applications such as word-of-mouth recommendation systems,²⁵ incentive mechanisms for routing queries to individuals possessing relevant information,²² and methods to track the spread of information among Weblogs.^{2, 14} Large-scale social contagion data also provides the opportunity to identify highly influential sets of people in a social network—the set of people who would trigger the largest cascade if they were to adopt an innovation.¹¹ The search for such influential sets is a computationally difficult problem, although recent work has shown that when social influence follows the kind of “diminishing returns” pattern discussed here, it is possible to find approximate methods with provable guarantees.^{19, 32}

Further Directions

Research on large-scale social-network data is proceeding in many further directions as well. While much of what we have been discussing involves the dynamic behavior of individuals in social networks, an important and complementary area of inquiry is how the structure of the network itself evolves over time.

Recent studies of large datasets have shed light on several important principles of network evolution. A central one, rooted in early work in the social sciences, is the principle of “preferential attachment”—the idea that nodes that already have many links will tend to acquire them at a greater rate.³³ An active line of research has shown how preferential attachment can lead to the highly

skewed distributions of links that one sees in real networks, with certain nodes acting as highly connected “hubs.”³

Another principle, also a key issue in sociology, is the notion of “triadic closure:” links are much more likely to form between two people when they have a friend in common.³⁴ Recent work using email logs has provided some of the first concrete measurements of the effect of triadic closure in a social-communication network.²⁴


Further principles have begun to emerge from recent studies of social and information networks over time, including “densification effects,” in which the number of links per node increases as the network grows, and “shrinking diameters,” in which the number of steps in the shortest paths between nodes can actually decrease even as the total number of nodes is increasing.²⁷

It is also intriguing to ask whether machine-learning techniques can be effective at predicting the outcomes of social processes from observations of their early stages. Problems here include the prediction of new links, the participation of people in new activities, the effectiveness of groups at collective problem-solving, and the growth of communities over time.^{4, 16, 17, 18, 28, 37} Recent work by Salganik, Dodds, and Watts raises the interesting possibility that the outcomes of certain types of social-feedback effects may in fact be *inherently unpredictable*.³⁶ Through an online experiment in which participants were assigned to multiple, independently evolving versions of a music-download site—essentially, a set of artificially constructed “parallel universes” in which copies of the site could develop independently—Salganik et al. found that when feedback was provided to users about the popularity of the items being downloaded, early fluctuations in the popularities of different items could get locked in to produce very different long-term trajectories of popularity. Developing an expressive computational model for this phenomenon is an interesting open question.

Ultimately, across all these domains, the availability of such rich and plentiful data on human interaction has closed an important feedback loop, allowing us to develop and evaluate models of social phenomena at

large scales and to use these models in the design of new computing applications. Such questions challenge us to bridge styles of scientific inquiry—ranging from subtle small-group studies to computation on massive datasets—that traditionally have had little contact with each other. And they are compelling questions in need of answers—because at their heart, they are about the human and technological connections that link us all, and the still-mysterious rhythms of the networks we inhabit.

Acknowledgments

I thank the National Science Foundation, the MacArthur Foundation, the Cornell Institute for the Social Sciences, Google, and Yahoo for their support and the anonymous reviewers of this manuscript for their comments and feedback. 

References

- Adamic, L. and Adar, E. How to search a social network. *Social Networks* 27, 3 (2005), 187–203.
- Adar, E., Zhang, L., Adamic, L.A., and Lukose, R.M. Implicit structure and the dynamics of blogspace. In *Proceedings of the Workshop on the Weblogging Ecosystem*, 2004.
- Albert, R. and Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74 (2002), 47–97.
- Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.
- Bookstein, A. Informetric distributions, Part II: Resilience to ambiguity. *J. American Society for Information Science* 41, 5 (1990), 376–386.
- Centola, D. and Macy, M. Complex contagions and the weakness of long ties. *American J. of Sociology* 113 (2007), 702–734.
- de Sola Pool, I. and Kochen, M. Contacts and influence. *Social Networks* 1, 1 (1978) 5–51.
- Demers, A.J., Greene, D.H., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturges, H.E., Swinehart, D.C., and Terry, D.B. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing* (1987), 1–12.
- Dodds, P., Muhamad, R., and Watts, D. An experimental study of search in global social networks. *Science* 301 (2003), 827–829.
- Dodds, P. and Watts, D. Universal behavior in a generalized model of contagion. *Physical Review Letters* 92 (2004).
- Domingos, P. and Richardson, M. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2001), 57–66.
- Garfield, E. It's a small world after all. *Current Contents* 43 (1979), 5–10.
- Granovetter, M. Threshold models of collective behavior. *American J. of Sociology* 83 (1978), 1420–1443.
- Gruhl, D., Liben-Nowell, D., Guha, R.V., and Tomkins, A. Information diffusion through blogspace. In *Proceedings of the 13th International World Wide Web Conference*, 2004.
- Guare, J. *Six Degrees of Separation: A Play*. Vintage Books, 1990.
- Hill, S., Provost, F., and Volinsky, C. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science* 21, 2 (2006), 256–278.
- Hoff, P.D., Raftery, A.E., and Handcock, M.S. Latent space approaches to social network analysis. *J. American Statistical Association* 97 (2002).
- Kearns, M., Suri, S., and Montfort, N. An experimental study of the coloring problem on human subject networks. *Science* 313 (2006), 824–827.
- Kempe, D., Kleinberg, J., and Tardos, E. Maximizing the spread of influence in a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2003), 137–146.
- Kleinberg, J. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing* (2000), 163–170.
- Kleinberg, J. Small-world phenomena and the dynamics of information. In *Proceedings of the 14th Advances in Neural Information Processing Systems* (2001), 431–438.
- Kleinberg, J. and Raghavan, P. Query incentive networks. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science* (2005), 132–141.
- Korte, C. and Milgram, S. Acquaintance networks between racial groups: Application of the small world method. *J. Personality and Social Psychology* 15 (1978).
- Kossinets, G., and Watts, D. Empirical analysis of an evolving social network. *Science* 311 (2006), 88–90.
- Leskovec, J., Adamic, L., and Huberman, B. The dynamics of viral marketing. *ACM Transactions on the Web* 1, 1 (May 2007).
- Leskovec, J. and Horvitz, E. Worldwide buzz: Planetary-scale views on an instant-messaging network. In *Proceedings of the 17th International World Wide Web Conference*, 2008.
- Leskovec, J., Kleinberg, J.M., and Faloutsos, C. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2005), 177–187.
- Liben-Nowell, D. and Kleinberg, J. The link-prediction problem for social networks. *J. American Society for Information Science and Technology* 58, 7 (2007), 1019–1031.
- Liben-Nowell, D. and Kleinberg, J. Tracing information flow on a global scale using Internet chain-letter data. In *Proceedings of Natl. Acad. Sci.* 105, 12 (Mar. 2008), 4633–4638.
- Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., and Tomkins, A. Geographic routing in social networks. In *Proceedings of Natl. Acad. Sci.* 102, 33 (Aug. 2005), 11623–11628.
- Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Commun. Surveys and Tutorials* 7, 2 (2005), 72–93.
- Mossel, E. and Roch, S. On the submodularity of influence in social networks. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, 2007.
- Newman, M.E.J. The structure and function of complex networks. *SIAM Review*, 45 (2003), 167–256.
- Rapoport, A. Spread of information through a population with socio-structural bias I: Assumption of transitivity. *Bulletin of Mathematical Biophysics* 15, 4 (Dec. 1953), 523–533.
- Rogers, E. *Diffusion of Innovations, 4th Edition*. Free Press, 1995.
- Salganik, M., Dodds, P., and Watts, D. Experimental study of inequality and unpredictability in an artificial cultural market. *Science* 311 (2006), 854–856.
- Sarkar, P. and Moore, A.W. Dynamic social network analysis using latent space models. In *Advances in Neural Information Processing Systems*, 2005.
- Travers, J. and Milgram, S. An experimental study of the small world problem. *Sociometry* 32, 4 (1969), 425–443.
- Watts, D.J., Dodds, P.S., and Newman, M.E.J. Identity and search in social networks. *Science* 296 (May 2002), 1302–1305.
- Watts, D.J. and Strogatz, S.H. Collective dynamics of ‘small-world’ networks. *Nature* 393 (1998), 440–442.

Jon Kleinberg (kleinber@cs.cornell.edu) is a professor of computer science at Cornell University, Ithaca, NY. His work focuses on issues at the interface of networks and information, with an emphasis on the social and information networks that underpin the Web and other online media. He is a recipient of MacArthur, Packard, and Sloan Foundation Fellowships, and the Nevanlinna Prize from the International Mathematical Union.

research highlights

P. 74

**Technical
Perspective
The Polaris
Tableau System**

By Jim Gray

P. 75

**Polaris: A System for Query,
Analysis, and Visualization of
Multidimensional Databases**

By Chris Stolte, Diane Tang, and Pat Hanrahan

P. 85

**Technical
Perspective
Safeguarding
Online Information
against Failures
and Attacks**

By Barbara Liskov

P. 86

**Zyzyva: Speculative Byzantine
Fault Tolerance**

By Ramakrishna Kotla, Allen Clement, Edmund Wong, Lorenzo Alvisi,
and Mike Dahlin

Technical Perspective

The Polaris Tableau System

By Jim Gray

**Jim Gray nominated the Polaris paper for the Research Highlights section and wrote the first draft of this Technical Perspective in November 2006. David Patterson revised the essay in August 2008.*

DATA-INTENSIVE APPLICATIONS often have dozens of independent dimensions with a set of measurements for each. Such high-dimensional datasets involving many variables and measurements are increasingly common, but good tools to analyze them are not. If you have ever been frustrated when trying to plot a useful graph from a simple spreadsheet, you would appreciate the value of a system that allows users to create stunning graphs interactively and easily from large multidimensional datasets.

Stolte, Tang, and Hanrahan have done that with Polaris, a declarative visual query language that unifies the strengths of visualization and database communities. It allows users to visualize relationships between data using shape, size, orientation, color, and texture in all kinds of graphs, and leverages the advances in database systems to optimize performance of accesses to large datasets. This combination lets you interactively explore the raw data or perform data analysis. It is a major improvement over how analysis is currently done.

Their work makes three advances in parallel: First, they show how to automatically construct graphs, charts, maps, and timelines as table visualizations. While these ideas are implicit in many graphing packages, the authors unified several approaches into a simple algebra for graphical presentation of quantitative and categorical information. The unification makes it easy to switch from one representation to another and to change or add dimensions to a graphical presentation. Second, they unify this graphical language with the SQL query languages, producing a declarative visual query language in which a single “program” specifies both data retrieval and data presentation. The third advance is a GUI that “writes” the visual queries as you drag and drop

dimensions or measurements in a data viewer. This combination makes it simple for users to ask “what if” questions for large multidimensional datasets.

Here is the “Visual SQL” to create a map by U.S. ZIP code of fundraising by the U.S. presidential candidates through May 2008. It places the results on a map using the latitude and longitude and lets the system pick the size of the circles representing the relative amounts of fundraising. It totals the amount of fundraising per candidate per ZIP code.

In fact the research for this work was conducted several years ago and incor-

porated as a commercial product—the Tableau system—that can analyze high-dimensional data from flat files, spreadsheets, and SQL data sources. The data algebra and graph algebra developed here is key to the success of the visual query language and Tableau.

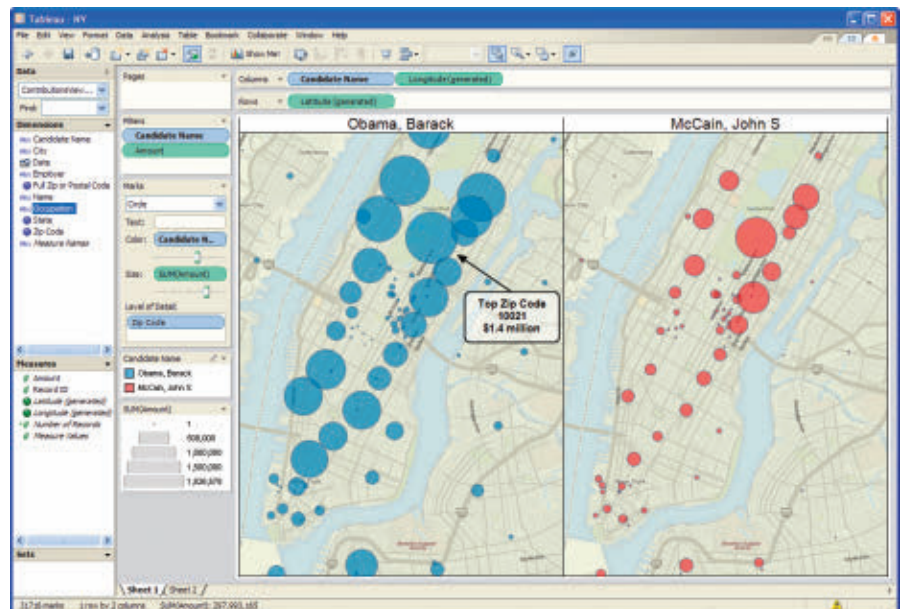
Hence, this is a rare paper that explains both the basic premise and its real-world evaluation. The notion that a formal algebra of relationships between tables and visual encodings would help the exploratory nature of the system was indeed validated. However, they found that default values of the visual encodings were important since few users opted to choose the details of shape and color selection, since they were not trained graphic designers nor psychologists and would rather spend their time exploring the data. □

```
SELECT Latitude ON ROWS,
( [Candidate Name] * Longitude ) ON COLUMNS,
[Candidate Name] ON COLOR,
SUM(Amount) ON SIZE,
[Zip Code] ON LEVEL_OF_DETAIL
FROM [Contributions View]
WHERE [Candidate Name] = { "John McCain", "Barack Obama" }
```

The VizSQL code above results in the following SQL code to collect the data:

```
SELECT ([Contributions View].[Candidate Name]),
([Contributions View].[Zip Code]),
(SUM([Contributions View].[Amount]))
FROM [dbo].[Contributions View]
WHERE ((([Contributions View].[Candidate Name]) IN ('McCain, John S', 'Obama, Barack'))
GROUP BY ([Contributions View].[Candidate Name]), ([Contributions View].[Zip Code])
```

The graph here is the output of this query zoomed into Manhattan Island in New York City.



Polaris: A System for Query, Analysis, and Visualization of Multidimensional Databases

By Chris Stolte, Diane Tang, and Pat Hanrahan

Abstract

During the last decade, multidimensional databases have become common in the business and scientific worlds. Analysis places significant demands on the interfaces to these databases. It must be possible for analysts to easily and incrementally change both the data and their views of it as they cycle between hypothesis and experimentation.

In this paper, we address these demands by presenting the Polaris formalism, a visual query language for precisely describing a wide range of table-based graphical presentations of data. This language compiles into both the queries and drawing commands necessary to generate the visualization, enabling us to design systems that closely integrate analysis and visualization. Using the Polaris formalism, we have built an interactive interface for exploring multidimensional databases that analysts can use to rapidly and incrementally build an expressive range of views of their data as they engage in a cycle of visual analysis.

1. INTRODUCTION

Nowadays, structured databases are widely used. Corporations store every sales transaction in large data warehouses. International research projects such as the Human Genome Project and Digital Sky Survey are generating massive scientific databases. Organizations such as the United Nations are making a wide range of global indicators on issues ranging from carbon emission to the adoption of technology publicly available via the Internet.

Unfortunately, our ability to collect and store data has rapidly exceeded our ability to analyze it. A major challenge in computer science is how to extract meaning from data: to discover structure, find patterns, and derive causal relationships. An analytical session cycles between hypothesis, experiment, and discovery. Often the path of exploration is unpredictable, and thus analysts need to be able to rapidly change both what data they are viewing and how they are viewing that data. This exploratory analysis process places significant demands on the human-computer interfaces to these databases. Few good tools exist.

In this paper, we present a formal approach to building visualization systems that addresses these demands.

The authors dedicate this article to the memory of Jim Gray, whose pioneering work inspired this research.

The first contribution is the Polaris formalism, a declarative visual query language that specifies a wide range of 2D graphic displays. The three key components of the formalism are (1) a table algebra that captures the structure of tables and spatial encodings, (2) a graphic taxonomy that results in an intuitive specification of graphic types, and (3) a system for effective visual encoding. This language allows for easily changing between different graphic displays as well as adding or removing data.

The second main contribution is the combination of this visual query language with the underlying database queries needed. This allows us to combine both visualization as well as the underlying data transformations to support the exploratory process.

The final contribution is the Polaris interface that allows users to incrementally construct a visual specification by dragging fields onto “shelves” (see Figure 1). Each intermediate specification is valid and corresponds to a graphical data display, giving the user quick visual feedback to support this analysis. This interface is built on top of the visual query language that specifies both the data and graphical transformations needed, thus combining statistical analysis and visualization. Polaris enables visual analysis by allowing an analyst to answer a question by composing a picture of what they want to see.

It has been 6 years since this work was originally published. In that time, the technology has been commercialized by Tableau Software as Tableau Desktop and is currently in use by thousands of companies and tens of thousands of users. As a result, we have gained considerable experience that has validated the effectiveness of the visual query language and interface and resulted in extensions and revisions to both.

2. OVERVIEW

Polaris has been designed to support the interactive exploration of large multidimensional relational databases or data cubes. Relational databases organize data into tables where each row in a table corresponds to a basic entity or fact and each column represents a property of that entity.¹⁸ We refer to a row in a relational table as a *tuple* or *record*, and a column as a *field*. A single database will contain many heterogeneous but interrelated tables.

A previous version of this paper was published in IEEE’s *Transactions on Visualization and Computer Graphics*, vol 8, issue 1 (Jan. 2002), pp. 52–65.

We can classify fields in a database as nominal, ordinal, quantitative, or interval.^{4,16} This classification is the field's **scale**. Polaris reduces this categorization to ordinal and quantitative by treating intervals as quantitative and assigning an ordering to the nominal fields to treat them as ordinal. A field's scale affects its visual representation. Quantitative fields are continuous, and are shown as axes or smoothly varying values. Ordinal scales are represented discretely, as headers or different classes.

The fields within a relational table can also be partitioned into two types: dimensions and measures. This classification is the field's **role**. Dimensions and measures are similar to independent and dependent variables in traditional analysis. For example, a product name or type would be a dimension while the product price or size would be a measure. The field's role determines how the query is generated. Measures are computed using an aggregation function and dimensions form the groups to be aggregated.

Polaris originally treated ordinal fields as dimensions and quantitative fields as measures. With experience, however, we have found that a field's scale and role are orthogonal, and may change depending on the question. For example, when asking the question "What is the average age of people purchasing a product?" the field Age is acting as a measure. However, when asking the question "What is the average amount spent classified by customer age?" then Age is acting as a dimension. The current implementation of Tableau uses simple heuristics based on a field's data type and domain cardinality to determine a field's default role and scale, but allows both to be easily changed.

To effectively support the analysis process in large multi-dimensional databases, an analysis tool must meet several demands:

- **Exploratory interface:** Analysts must be able to rapidly and incrementally change what data they are viewing and how they are viewing that data as they explore hypotheses.
- **Multiple display types:** Analysis consists of different tasks such as discovering correlations between variables, finding patterns, and locating outliers. An analysis tool must be able to generate displays suited to these disparate tasks.
- **Data-dense displays:** The databases typically contain a large number of records and dimensions. Analysts need to be able to create visualizations that will simultaneously display many dimensions of large subsets of the data.

Polaris addresses these demands by providing an interface for rapidly and incrementally generating table-based displays. In Polaris, a table consists of a number of rows, columns, and layers. Each table axis may contain multiple nested dimensions. Each table entry, or *pane*, contains a set of records that are visually encoded as a set of marks to create a graphic.

Several characteristics of tables make them particularly effective for displaying multidimensional data:

- **Multivariate:** The table structure can encode multiple dimensions, enabling the display of high-dimensional data.
- **Comparative:** Tables generate *small-multiple* displays of information, which are easily compared to expose patterns and trends across dimensions.²⁰
- **Familiar:** Table-based displays have an extensive history. Statisticians are accustomed to using tabular displays of graphs, such as scatterplot matrices and Trellis displays, for analysis.^{2,7,20}

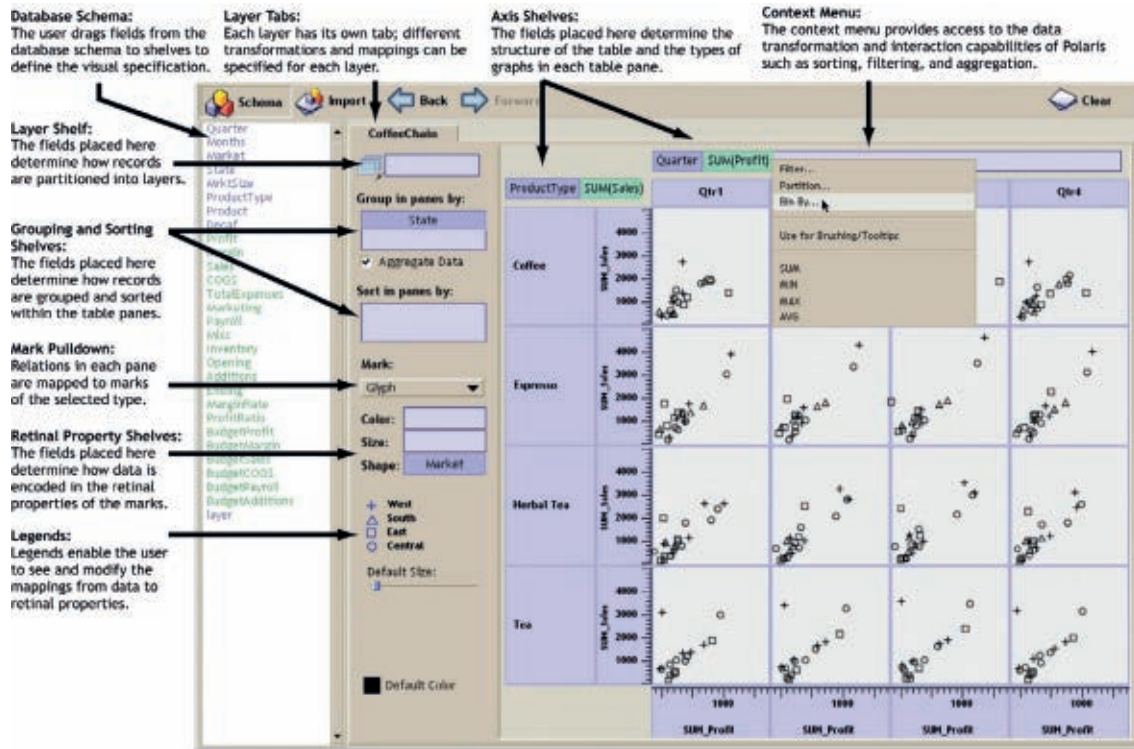
Figure 1 shows the Polaris user interface. In this example, the analyst has constructed a matrix of scatterplots showing sales versus profit for different product types in different quarters. The primary interaction technique is to drag-and-drop fields from the database schema onto shelves throughout the display. We call a given configuration of fields on shelves a *visual specification*. The visual specification is converted into the language using simple operations. The specification determines the analysis and visualization operations to be performed by the system, defining:

- The mapping of data sources to layers. Multiple data sources may be combined in a single Polaris visualization. Each data source maps to a separate layer or set of layers.
- The number of rows, columns, and layers in the table and their relative orders (left to right as well as back to front). The database dimensions assigned to rows are specified by the fields on the *y* shelf, columns by fields on the *x* shelf, and layers by fields on the *layer* (*z*) shelf. Multiple fields may be dragged onto each shelf to show categorical relationships.
- The selection of records from the database and the partitioning of records into different layers and panes.
- The grouping of data within a pane and the computation of statistical properties, aggregates, and other derived fields. Records may also be sorted into a given drawing order.
- The type of graphic displayed in each table pane. Each graphic consists of a set of marks, one mark per record in that pane.
- The mapping of data fields to retinal properties of the marks in the graphics. The mappings used for any given visualization are shown in a set of automatically generated legends.

Analysts can interact with the resulting visualizations in several ways. Each mark represents a tuple, so selecting a single mark in a graphic by clicking on it pops up a detail window that displays user-specified field values for the tuples corresponding to that mark. The tuples represented by a set of marks can be cut and pasted into a spreadsheet by selecting the marks representing the tuples. Analysts can draw rubber bands around a set of marks to brush or highlight related records, either within a single table or between multiple Polaris displays.

In Section 3, we describe how the visual specification is used to generate graphics. In Section 4, we describe the supported data transformations and how the visual specifications are

Figure 1: The Polaris user interface. Analysts construct table-based displays of data by dragging fields from the database schema onto shelves throughout the display. A given configuration of fields on shelves is called a visual specification. The specification unambiguously defines the analysis and visualization operations to be performed by the system to generate the display.



used to generate the database queries for statistical analysis.

3. GENERATING GRAPHICS

The visual specification consists of three components: (a) the specification of the table configuration, (b) the type of graphic inside each pane, and (c) the details of the visual encodings (for more details, see¹⁷). We discuss each of these in turn.

3.1. Table algebra

We define an algebra as a formal mechanism to specify table configurations. When analysts place fields on shelves, as shown in Figure 1, they are implicitly creating expressions in this algebra.

A complete table configuration consists of three separate expressions in this table algebra. Two of the expressions define the configuration of the x and y axes of the table, partitioning the table into rows and columns. The third expression defines the z -axis of the table, which partitions the display into *layers*. The x , y , and z expressions form clauses in the language.

The operands in this table algebra are the names of the ordinal and quantitative fields of the database. We will use A , B , and C to represent ordinal fields and P , Q , and R to represent quantitative fields. We assign sequences of values to each field symbol in the following manner: to ordinal fields we assign the members of the ordered domain of the field, and to quantitative fields we assign the single element set containing the field name.

Ordinal and quantitative fields generate tables with

$$A = \text{domain}(A) = \{a_1, \dots, a_n\}$$

$$P = \{P\}$$

different structures. Ordinal fields partition the table into rows and columns using headers, whereas quantitative fields generate axes.

A valid expression in our algebra is one or more field symbols with operators between each pair of adjacent symbols, and with parentheses used to alter the precedence of the operators. The operators in the algebra are cross (\times), nest ($/$), and concatenation ($+$), listed in order of precedence. The precise semantics of each operator is defined in terms of its effects on sequences.

Concatenation: The plus operator concatenates two sequences:

$$A+B = \{a_1, \dots, a_n\} + \{b_1, \dots, b_m\}$$

$$= \{a_1, \dots, a_n, b_1, \dots, b_m\}$$

$$A+P = \{a_1, \dots, a_n\} + \{P\}$$

$$= \{a_1, \dots, a_n, P\}$$

$$P+Q = \{P\} + \{Q\}$$

$$= \{P, Q\}$$

Cross: The cross operator performs a Cartesian product of elements in the two sequences:

Nest: The nest operator is similar to the cross operator, but it only creates sequence entries for which there exist records

$$\begin{aligned}
 A \times B &= \{a_1, \dots, a_n\} \times \{b_1, \dots, b_m\} \\
 &= \{a_1 b_1, \dots, a_1 b_m, \\
 &\quad a_2 b_1, \dots, a_2 b_m, \dots, \\
 &\quad a_n b_1, \dots, a_n b_m\} \\
 A \times P &= \{a_1, \dots, a_n\} \times P \\
 &= \{a_1 P, \dots, a_n P\}
 \end{aligned}$$

with those domain values. If we define R to be the dataset being analyzed, r to be a record, and $A(r)$ to be the value of the field A for the record r , then we can define the nest operator as follows:

$$A/B = \{a_i b_j \mid \exists r \in R \text{ st } A(r) = a_i \ \& \ B(r) = b_j\}$$

The intuitive interpretation of the *nest* operator is “B within A”. For example, given the fields *quarter* and *month*, the expression *quarter/month* would be interpreted as those months within each quarter, resulting in three entries for each quarter. In contrast, *quarter × month* would result in 12 entries for each quarter. Data cubes represent hierarchies explicitly and there is no need to compute the nest relationship.

Using the above semantics for each operator, every expression in the algebra can be reduced to a single sequence, with each entry in the sequence being an ordered term consisting of zero or more ordinal values with zero or more quantitative field names. We call this set evaluation of an expression the

normalized form. The normalized form of an expression determines one axis of the table: the table axis is partitioned into columns (or rows or layers) so that there is a one-to-one correspondence between set entries in the normalized set and columns. Figure 2 illustrates the configurations resulting from a number of expressions.

Analysts can also combine multiple data sources in a single Polaris visualization. When multiple data sources are imported, each data source is mapped to a distinct layer (or set of layers). While all data sources and all layers share the same configuration for the x and y axes of the table, each data source can have a different expression for partitioning its data into layers.

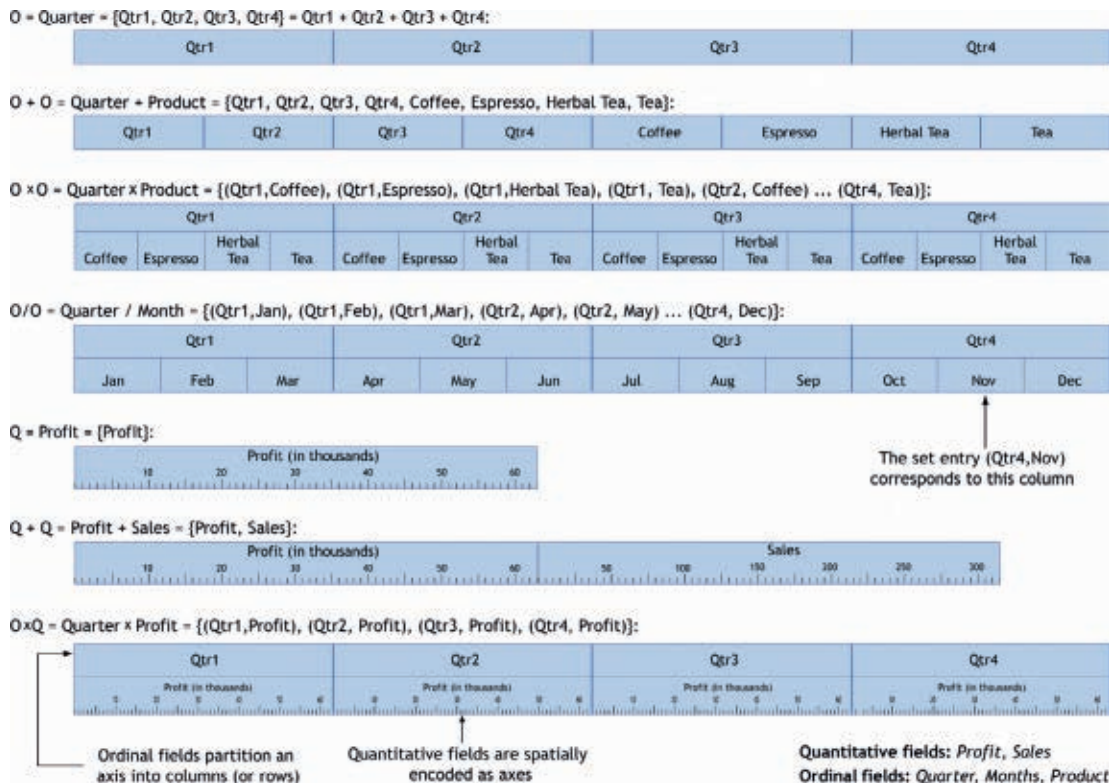
In retrospect, the Polaris table algebra is very similar to the operations in the MDX query language for data cubes.

3.2. Types of graphics

Given a table configuration, the next step is to specify the type of graphic in each pane. We have developed a taxonomy of graphics that results in an intuitive and concise specification of graphic types. This taxonomy is based on both the axes within each pane (implicitly specified from the table configuration via the role and scale of the innermost field in the sequence) as well as the mark type representing a tuple (e.g., text, shape, bar, etc.). We group this taxonomy into three families (illustrated in Figure 3) based on the axes: ordinal-ordinal, ordinal-quantitative, and quantitative-quantitative.

Each family contains a number of variants depending

Figure 2: The graphical interpretation of several expressions in the table algebra. Each expression in the table algebra can be reduced to a single sequence of terms, and that sequence can then be directly mapped into a configuration for an axis of the table.



on how records are mapped to marks. For example, selecting a bar in an ordinal–quantitative pane will result in a bar chart, whereas selecting a line mark results in a line chart. The mark set currently supported in Polaris includes the rectangle, circle, glyph, text, Gantt bar, line, polygon, and image. There are two types of marks; single tuple marks and multituple marks. Multituple marks form a single graphical entity from a set of marks; an example is a polygon mark where each vertex of the polygon is a single tuple.

Following Cleveland,⁸ we further structure the space of graphics by the number of independent and dependent variables. For example, a graphic where both axes encode independent variables is different than a graphic where one axis encodes an independent variable and the other encodes a dependent variable ($y = f(x)$). By default, dimensions of the database are interpreted as independent variables and measures as dependent variables. We briefly discuss the defining characteristics of the three families within our categorization. It should be noted that these rules allow us to automatically choose a default mark given the types of the fields on the axes.

Ordinal–Ordinal Graphics: The characteristic member of this family is the table, either of numbers or of marks encoding attributes of the source records.

In ordinal–ordinal graphics, the axis variables are typically independent of each other, and the task is focused on understanding patterns and trends in some function $f(O_x, O_y) \rightarrow R$, where R represents the fields encoded in the retinal properties of the marks. This can be seen in the heatmap in Figure 3, where the analyst is studying gene expression as a function of experiment and gene. Figure 6(a) shows another example where lines of source code are color-encoded with the number of cache misses attributable to that line.

Ordinal–Quantitative Graphics: Well-known representatives of this family of graphics are the bar chart, the dot plot, and the Gantt chart.

In an ordinal–quantitative graphic, the quantitative variable is often dependent on the ordinal variable, and the

analyst is trying to understand or compare the properties of some set of functions $f(O) \rightarrow Q$. The cardinality of the record set affects the structure of the graphics in this family: When the cardinality of the record set is one, the graphics are simple bar charts or dot plots. When the cardinality is greater than one, additional structure may be introduced to accommodate the additional records (e.g., a stacked or clustered bar chart).

The ordinal and quantitative values may be independent variables, such as in a Gantt chart. Here, each pane represents all events in a category; each event has a type as well as a beginning and end time. Figure 6(c) shows a table of Gantt charts, with each Gantt chart displaying the thread scheduling and locking activity on a CPU within a multiprocessor computer.

Quantitative–Quantitative Graphics: Graphics of this type are used to understand the distribution of data as a function of one or both quantitative variables and to discover causal relationships between the two quantitative variables, such as in a scatterplot matrix. Figure 3 illustrates another example of a quantitative–quantitative graphic: the map. In this figure, the analyst is studying election results by county.

3.3. Visual mappings

Each record in a pane is mapped to a mark. There are two components to the visual mapping. The first component, described in Section 3.2, determines the type of graphic and mark. The second component encodes fields of the records into visual or retinal properties of the selected mark. The visual properties in Polaris are based on Bertin’s retinal variables:⁴ shape, size, orientation, color (value and hue), and texture.

Allowing analysts to explicitly encode different fields of the data to retinal properties of the display greatly enhances the data density and the variety of displays that can be generated. However, in order to keep the specification succinct, analysts should not be required to construct the mappings. Instead, they should be able to simply specify that a field be

Figure 3: The families of graphics within our taxonomy with examples of well-known charts from each family. The taxonomy structures the space of graphics into three families by the types of fields assigned to their axes and then further structures each family by the number of independent and dependent variables. Using this taxonomy we can derive the type of graphic within each pane from the table axes expressions and the mark type.

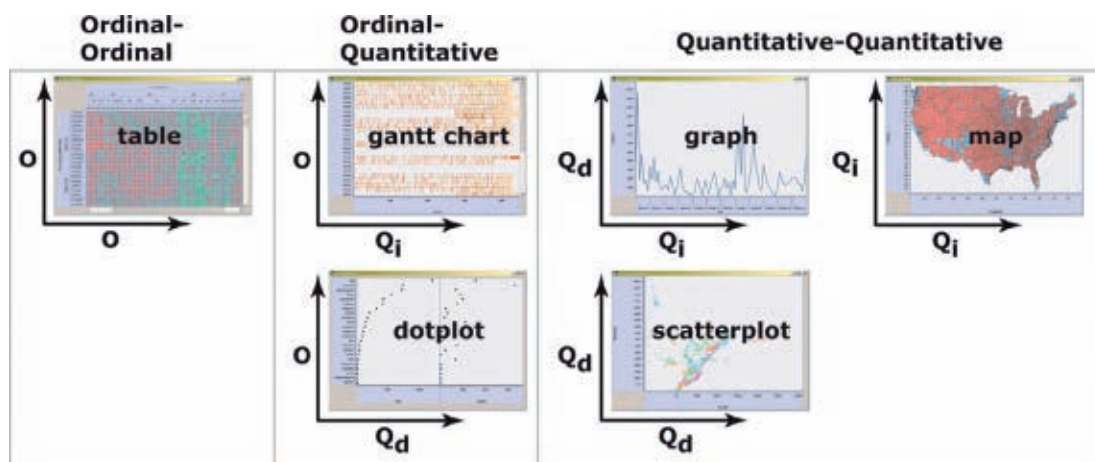
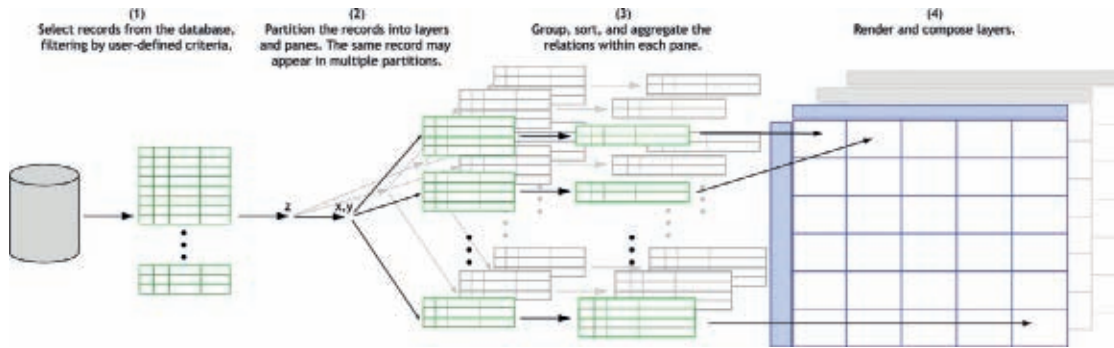


Figure 5: The transformations and data flow within Polaris. The visual specification generates queries to the database to select subsets of the data for analysis, then to filter, sort, and group the results into panes, and then finally to group, sort and aggregate the data within panes.



phase of the data flow partitions the retrieved records into groups corresponding to each pane in the table. As we discussed in Section 3.1, the *normalized set form* of the table axis expressions determines the table configuration. The table is partitioned into rows, columns, and layers corresponding to the entries in these sets.

The ordinal values in each set entry define the criteria by which records will be sorted into each row, column, and layer. Let $Row(i)$ be the predicate that represents the selection criteria for the i th row, $Column(j)$ be the predicate for the j th column, and $Layer(k)$ the predicate for the k th layer. For example, if the y -axis of the table is defined by the normalized set:

$$\{a_1b_1P, a_1b_2P, a_2b_1P, a_2b_2P\}$$

then there are four rows in the table, each defined by an entry in this set, and Row would be defined as:

$$\begin{aligned} Row(1) &= (A = a_1 \text{ and } B = b_1) \\ Row(2) &= (A = a_1 \text{ and } B = b_2) \\ Row(3) &= (A = a_2 \text{ and } B = b_1) \\ Row(4) &= (A = a_2 \text{ and } B = b_2) \end{aligned}$$

Given these definitions, the records to be partitioned into the pane at the intersection of the i th row, the j th column, and the k th layer can be retrieved with the following query:

```
SELECT *
WHERE {Row(i) and Column(j) and
      Layer(k)}
```

To generate the groups of records corresponding to each of the panes, we must iterate over the table executing this SELECT statement for each pane, which is clearly nonoptimal. Various optimizations are discussed in.¹⁷

Step 3: Transforming Records within the Panes: The last phase of the data flow is the transformation of the records in each pane. If the visual specification includes aggregation, then each measure in the database schema must be assigned an aggregation operator. If the user has not

selected an aggregation operator for a measure, that measure is assigned the default aggregation operator (SUM). We define the term *aggregates* as the list of the aggregations that need to be computed. For example, if the database contains the quantitative fields Profit, Sales, and Payroll, and the user has explicitly specified that the average of Sales should be computed, then *aggregates* is defined as:

```
aggregates =
SUM(Profit), AVG(Sales), SUM(Payroll)
```

Aggregate field filters (for example, $SUM(Profit) > 500$) could not be evaluated in Step 1 with all of the other filters because the aggregates had not yet been computed. Thus, those filters must be applied in this phase. We define the relational predicate *filters* as in Step 1 for unaggregated fields.

Additionally, we define the following lists:

- G : the field names in the grouping shelf,
- S : the field names in the sorting shelf, and
- dim : the dimensions in the database.

The necessary transformation can then be expressed by the SQL statement:

```
SELECT {dim}, {aggregates}
GROUP BY {G}
HAVING {filters}
ORDER BY {S}
```

If no aggregate fields are included in the visual specification, then the remaining transformation simply sorts the records into drawing order:

```
SELECT *
ORDER BY {S}
```

5. RESULTS

Polaris is useful for performing the type of exploratory data analysis advocated by statisticians such as Bertin³ and Cleveland.⁸ We demonstrate the capabilities of Polaris as an exploratory interface to multidimensional databases by considering the following scenario.

At Stanford, researchers developing Argus,⁹ a parallel graphics library, found that its performance had linear speedup when using up to 31 processors, after which its performance diminished rapidly. Using Polaris, we recreate the analysis they performed using a custom-built visualization tool.⁵

Initially, the developers hypothesized that the diminishing performance was a result of too many remote memory accesses, a common performance problem in parallel programs. They collected and visualized detailed memory statistics to test this hypothesis. Figure 6(a) shows a visualization constructed to display this data. The visualization is composed of two linked Polaris instances. One displays a bird's eye view of multiple source code files with each line of code represented by a single pixel height bar and the other displays the detailed source-code. In both views, the hue of each line of code encodes the number of cache misses suffered by that line. Upon seeing these displays, they could tell that memory was in fact not the problem.

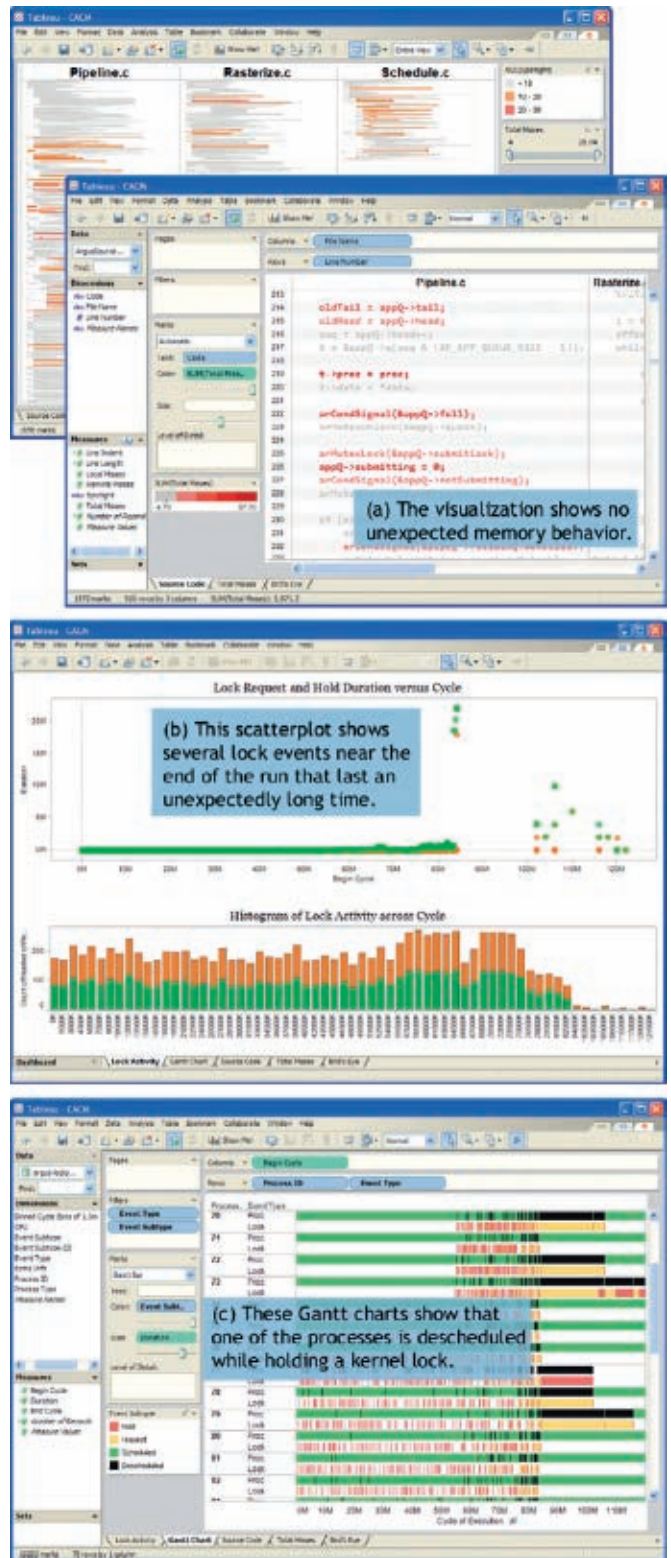
The developers next hypothesized that lock contention might be a problem, so they reran Argus and collected detailed lock and scheduling information. The data is shown in Figure 6(b) using a dashboard within Polaris to create a composite visualization with two linked projections of the same data. One projection shows a scatterplot of the start cycle versus cycle duration for the lock events (requests and holds). The second shows a histogram over time of initiated lock events. The scatterplot shows that toward the end of the run, the duration of lock events (both holds and requests) was taking an unexpectedly long time. That observation correlated with the histogram showing that the number of lock requests peaked and then tailed off towards the end of the run indicated that this might be a fruitful area for further investigation.

A third visualization, shown in Figure 6(c), shows the same data using Gantt charts to display both lock events and process-scheduling events. This display shows that the long lock requests correspond to descheduled periods for most processes. One process, however, has a descheduled period corresponding to a period during which the lock was held. This behavior, which was due to a bug in the operating system, was the source of the performance issues.

This example illustrates several important points about the exploratory process. Throughout the analysis, both the data that users want to see and how they want to see it change continually. Analysts first form hypotheses about the data and then create new views to test those hypotheses. Certain displays enable an understanding of overall trends, whereas others show causal relationships. As the analysts better understand the data, they may want to drill-down in the visible dimensions or display entirely different dimensions.

Polaris supports this exploratory process through its visual interface. By formally categorizing the types of

Figure 6: A scenario demonstrating the use of Polaris to analyze the performance of a parallel graphics library.



graphics, Polaris is able to provide a simple interface for rapidly generating a wide range of displays, allowing analysts to focus on the analysis task rather than interface.

6. EXPERIENCE

In the 6 years since this work was originally published, we have gained considerable experience with the formalism and the interface. In that time, the technology has been commercialized and extended by Tableau Software as Tableau Desktop and is used by thousands of companies and tens of thousands of users. The system has also been adapted to the web, so it is possible to perform analysis within a browser. The uses are diverse, ranging from disease research in the jungles of Central America to marketing analysis in Fortune 50 companies to usability analysis by video game designers, and the data sizes range from small spreadsheets to billions of rows of data. The many types of users indicate the ubiquity of data and the demand for new tools. This experience has emphasized three points to us: (1) the importance of a formal approach, (2) the importance of an architecture that leverages database technology rather than replaces it, and (3) the importance of building effective defaults into the graphical interface.

One question early on was: “do we need a formalism?” Most visualization systems have predefined types of charts and use wizards to help the user construct graphs. Having a language allows us to generate an unlimited number of different types of graphics. Restricting the set of views to a small set limits the power of visualization; this would be like building into a query language a small set of predefined queries. Experience has shown that this flexibility makes it possible to *incrementally* build new views, which is key to smoothly supporting the analysis process. Both of these aspects of Polaris are enabled by the formal nature of the algebra, where every addition or deletion leads to a new algebraic statement.

The formalism also enables us to unify the specification of the visualization with the database query: users can change the query used to fetch the data and their view of it simultaneously. In subsequent work, we have proved that the language is complete; that is, it is possible to generate any statement in the relational algebra. A major problem with many visual interfaces is that they restrict the types of queries that can be formed.

This unification of visualization and database queries is also a key architectural decision that makes it possible to use our system as a front-end to large parallel database servers. This makes it easy to access important data in existing data sources, to leverage high performance database technology (e.g., database appliances, massively parallel computation, column stores), and to avoid data replication and application-specific data silos. Why move a terabyte of data if you don't have to?

One potential issue with a compositional language is that it creates a large space of possible visualizations. While many are effective and aesthetically pleasing, many are not. Thus, choosing default graphics is an important part of any production system and allows for additional succinctness in the language. However, the issue is not just with choosing default graphics. Generating effective visual mappings (e.g., color, shape) by default is not a fundamental aspect of the language, but is equally important. Effective defaults enable users to focus on their task and questions rather than the

details of color or shape selection, especially since many users are not trained as graphic designers or psychologists.

7. RELATED WORK

The related work to Polaris can be divided into two categories: formal graphical specifications and database exploration tools.

7.1. Formal graphical specifications

We have built on the work of several researchers' insights into the formal properties of graphic communication, such as Bertin's *Semiology of Graphics*,⁴ Cleveland's experimental results on the perception of data,^{7,8} Wilkinson's formalism for statistical graphics,²² and Mackinlay's APT system.¹² However, the Polaris formalism is innovative in several ways. One key aspect of our approach is that all specifications can be compiled directly into queries. Existing formalisms do not consider the generation of queries to be related to the presentation of information. Another innovation is the use of an algebra to describe table-based displays. Tables are particularly effective for displaying multidimensional data, as multiple dimensions of the data can be explicitly encoded in the structure of the table. Finally, our formalism is the basis for several interactive tools for analyzing and exploring large data warehouses and this usage has affected the development of the formalism.

7.2. Database exploration tools

The second area of related work is visual query and database exploration tools. Academic projects such as Visage,¹⁴ DEViser,¹¹ and Tioga-2¹ have focused on developing visualization environments that support interactive database exploration through *visual queries*. Users construct queries and visualizations through their interactions with the visualization system interface. These systems have flexible mechanisms for mapping query results to graphs, and support mapping database records to retinal properties. However, none of these systems is based on an expressive formal language for graphics nor do they leverage table-based organizations of their visualizations.


Finally, existing systems, such as XmdvTool,²¹ Spotfire,¹⁵ and XGobi⁶ have taken the approach of providing a set of predefined visualizations, such as scatterplots and parallel coordinates. These views are augmented with interaction techniques, such as brushing and zooming, which can be used to refine the queries. We feel that this approach is much more limiting than providing the user with a set of building blocks that can be used to interactively construct and refine a wide range of displays to suit any analysis task.

8. CONCLUSIONS

We have presented Polaris, a visual query language for databases and a graphical interface for authoring queries in the language. The Polaris formalism uses succinct visual specifications to describe a wide range of table-based visualizations of multidimensional information. Visual specifications can be compiled into both the queries and the drawing commands necessary to generate the displays, thus unifying analysis and visualization into a single visual query language.

Using the Polaris formalism, we have built the Polaris interface. The Polaris interface directly supports the cycle of analysis. Analysts can incrementally create sophisticated visualizations using simple drag-and-drop operations to construct a visual specification. This interface has been commercialized and extended by Tableau Software and is now in use by tens of thousands of users in thousands of companies.

Acknowledgments

The authors especially thank Robert Bosch for his contribution to the design and implementation of Polaris, his review of manuscript drafts, and for many useful discussions. The authors also thank Maneesh Agrawala for his insightful reviews and discussions on early drafts of this paper. This work was supported by the Department of Energy through the ASCI Level 1 Alliance with Stanford University. 

References

- Aiken, A., Chen, J., Stonebraker, M., and Woodruff, A. Tioga-2: A direct manipulation database visualization environment. *Proceeding of the 12th International Conference on Data Engineering*, February 1996, pp. 208–217.
- Becker, R., Cleveland, W. S., and Douglas Martin, R. Trellis graphics displays: A multi-dimensional data visualization tool for data mining. *3rd Annual Conference on Knowledge Discovery in Databases*, August 1997.
- Bertin, J. *Graphics and Graphic Information Processing*. Walter de Gruyter, Berlin, 1980.
- Bertin, J. *Semiology of Graphics*. The University of Wisconsin Press, Madison, WI, 1983. Translated by W. J. Berg.
- Bosch, R., Stolte, C., Stoll, G., Rosenblum, M., and Hanrahan, P. Performance analysis and visualization of parallel systems using SimOS and Rivet: A case study. *Proceedings of the Sixth Annual Symposium on High-Performance Computer Architecture*, January 2000, pp. 360–371.
- Buja, A., Cook, D., and Swayne, D. F. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.
- Cleveland, W. S. *The Elements of Graphing Data*. Wadsworth Advanced Books and Software, Pacific Grove, CA, 1985.
- Cleveland, W. S. *Visualizing Data*. Hobart Press, New Jersey, 1993.
- Igehy, H., Stoll, G., and Hanrahan, P. The design of a parallel graphics interface. *Proceedings of SIGGRAPH*, 1998, pp. 141–150.
- Kosslyn, S. M. *Elements of Graph Design*. W.H. Freeman and Co., New York, NY, 1994.
- Livny, M., Ramakrishnan, R., Beyer, K., Chen, G., Donjerkovic, D., Lawande, S., Myllymaki, J., and Wenger, K. DEVis: Integrated querying and visual exploration of large datasets. *Proceeding of ACM SIGMOD*, May, 1997.
- Mackinlay, J. D. Automating the design of graphical presentations of relational information. *ACM Transaction of Graphics*, April 1986, pp. 110–141.
- Rogowitz, B. and Treinish, L. How NOT to lie with visualization. *Computers in Physics*, May/June 1996, pp. 268–274.
- Roth, S. F., Lucas, P., Senn, J. A., Gombert, C. C., Burks, M. B., Strofollino, P. J., Kolojchick, J., and Dunmire, C. Visage: A user interface environment for exploring information. *Proceeding of Information Visualization*, October 1996, pp. 3–12.
- Spotfire Inc. [online] Available: <http://www.spotfire.com>, cited September 2001.
- Stevens, S. S. On the theory of scales of measurement. *Science*, 103:677–680.
- Stolte, C. Query, analysis, and visualization of multidimensional databases. PhD Dissertation, Stanford University, June 2003.
- Thomsen, E. *OLAP Solutions: Building Multidimensional Information Systems*. Wiley Computer Publishing, New York, 1997.
- Travis, D. *Effective Color Displays: Theory and Practice*. Academic Press, London, 1991.
- Tufte, E. R. *The Visual Display of Quantitative Information*. Graphics Press, Box 430, Cheshire, CT, 1983.
- Ward, M. XmdvTool: Integrating multiple methods for visualizing multivariate data. *Proceedings of Visualization*, October 1994, pp. 326–331.
- Wilkinson, L. *The Grammar of Graphics*. Springer, New York, NY, 1999.

Chris Stolte is founder and vice president of Tableau Software, Seattle, WA. Diane Tang is a software researcher at Google, Inc., Mountain View, CA. Pat Hanrahan is Canon USA Professor in the School of Engineering at Stanford University, Stanford, CA.

© ACM 0001-0782/08/1100 \$5.00

Take Advantage of ACM's Lifetime Membership Plan!

- ◆ **ACM Professional Members** can enjoy the convenience of making a single payment for their entire tenure as an ACM Member, and also be protected from future price increases by taking advantage of **ACM's Lifetime Membership** option.
- ◆ **ACM Lifetime Membership** dues may be tax deductible under certain circumstances, so becoming a Lifetime Member can have additional advantages if you act before the end of 2008. (Please consult with your tax advisor.)
- ◆ Lifetime Members receive a certificate of recognition suitable for framing, and enjoy all of the benefits of **ACM Professional Membership**.

Learn more and apply at:

<http://www.acm.org/life>



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Technical Perspective

Safeguarding Online Information against Failures and Attacks

By Barbara Liskov

THE INTERNET IS increasingly the place where both users and organizations store their information. Storage is becoming a commodity; for example, consider the storage offerings by companies such as Google and Amazon.

A key benefit for individual users of commodity Internet storage is that they can access their information from anywhere at anytime from any device. Thus they no longer have to use their PC to access their files or their email. Furthermore, online information can easily be shared with others, giving rise to new applications based on support for collaboration.

However, the full benefit of online storage will be realized only if users can access their data whenever they want. Users need storage that is highly reliable (it is not lost) and highly available (accessible when needed). They will not be satisfied with less reliability or availability than they can get by storing their information locally. Providing these guarantees requires replication: by storing copies of information on multiple computers, it is possible to prevent loss and provide accessibility.

Replication has been the subject of research for over 20 years. The details of the replication protocols depend on the failure model, and two are in common use. The first is the crash model, in which either a computer is up and running as required by the protocol, or it has crashed and is doing nothing. The second is the Byzantine model, in which computers are allowed to fail in arbitrary ways. The Byzantine model is more general: in addition to crashes, it handles failures in which a faulty computer continues to run the protocol while misbehaving. For example, a machine might indicate it has executed a command to update information, but discard the new information.

During the 1980s there was a great deal of research on replication protocols that handle crash failures for two reasons: crashes were the most com-

mon failures at the time, and it is much easier to think about crashes than Byzantine failures. This work led to protocols that survive f failures using $2f + 1$ replicas, which is the minimum needed in an asynchronous setting like the Internet. Also, the protocols provide good performance, close to what an unreplicated system can provide.

However, these protocols are unable to handle arbitrary (Byzantine) failures, which are becoming more common. One source of Byzantine failures is software errors; typically these are non-deterministic errors, because deterministic errors are much more likely than non-deterministic ones to be removed during testing. The second source, and one of increasing concern today, is malicious attacks in which an adversary manages to get control of a computer and cause it to misbehave. To handle these problems, researchers have developed replication protocols that provide Byzantine fault tolerance.

Prior to the late 1990s, work on Byzantine-fault-tolerant replication was only of theoretical interest because the protocols were so costly or worked only in a synchronous network. This changed with the invention of PBFT, the first Byzantine-fault-tolerant replication protocol that could be used in practice in an asynchronous network. PBFT provides state machine replication; that is, it handles arbitrary operations on the service state. It requires the minimum of $3f + 1$ replicas to tolerate f failures.

The development of PBFT led to renewed interest in Byzantine-fault-tolerant replication protocols. Researchers have investigated a number of research questions, including:

Level of Consistency. At one extreme is a replication protocol like PBFT that appears to behave as if there were just one copy of the data. But performance can be improved by providing weaker guarantees.

Other Approaches. PBFT makes use


of a primary replica to direct the protocol; researchers have invented protocols that avoid the use of a primary either completely or partially.

Failure Analysis. Replication protocols like PBFT work correctly provided no more than f replicas are faulty. But if that bound is exceeded, can any guarantees be made?

Performance, performance, performance. Improved protocols that have better performance (lower latency, higher throughput) are always of interest.

The work on Zyzzyva presented here is concerned with the last topic. Zyzzyva achieves excellent performance when all replicas are non-faulty. It pays for this gain in performance in the non-failure case by offering reduced performance when there are failures. Importantly, its techniques should allow it to achieve performance that is close to that of an unreplicated system most of the time.

Today there are Byzantine-fault-tolerant replication protocols efficient enough to be deployed in a real setting. But when might this happen? Here we can learn from the work on crash replication. Although developed in the 1980s, these protocols weren't used in real systems until around 2000. The reason for this delay was a perception that the reliability provided by these approaches wasn't really needed in practice. This perception changed as more critical state was stored online. The concern about cost also changed, since computers are much cheaper, and the network is much faster.

I expect that someday there will be a practical deployment that tolerates Byzantine failures. The decision to take this step will depend on the criticality of the data. At some point incurring the cost of replication will be preferable to being liable should the data be lost or unavailable. 

Barbara Liskov (liskov@csail.mit.edu) is the Ford Professor of Engineering, MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA.

Zyzyva: Speculative Byzantine Fault Tolerance

By Ramakrishna Kotla,* Allen Clement, Edmund Wong, Lorenzo Alvisi, and Mike Dahlin

Abstract

A longstanding vision in distributed systems is to build reliable systems from unreliable components. An enticing formulation of this vision is Byzantine fault-tolerant (BFT) state machine replication, in which a group of servers collectively act as a correct server even if some of the servers misbehave or malfunction in arbitrary (“Byzantine”) ways. Despite this promise, practitioners hesitate to deploy BFT systems at least partly because of the perception that BFT must impose high overheads.

In this article, we present Zyzyva, a protocol that uses speculation to reduce the cost of BFT replication. In Zyzyva, replicas reply to a client’s request without first running an expensive three-phase commit protocol to agree on the order to process requests. Instead, they optimistically adopt the order proposed by a primary server, process the request, and reply immediately to the client. If the primary is faulty, replicas can become temporarily inconsistent with one another, but clients detect inconsistencies, help correct replicas converge on a single total ordering of requests, and only rely on responses that are consistent with this total order. This approach allows Zyzyva to reduce replication overheads to near their theoretical minima and to achieve throughputs of tens of thousands of requests per second, making BFT replication practical for a broad range of demanding services.

1. INTRODUCTION

Mounting evidence suggests that real systems must contend not only with simple crashes but also with more complex failures ranging from hardware data corruption²² to nondeterministic software errors²⁵ to security breaches. Such failures can cause even highly engineered services to become unavailable or to lose data. For example, a single corrupted bit in a handful of messages recently brought down the Amazon S3 storage service for several hours,³ and several well-known e-mail service providers have occasionally lost customer data.¹⁴

Byzantine fault-tolerant (BFT) state machine replication is a promising approach to masking many such failures and constructing highly reliable and available services. In BFT replication, $n \geq 3f + 1$ servers collectively act as a *correct* server even if up to f servers misbehave or malfunction in arbitrary (“Byzantine”) ways.^{15,16}

Today, three trends make real-world deployment of BFT increasingly attractive. First, as noted above, there is mounting evidence of non-fail-stop behaviors in real systems, motivating the use of new techniques to improve robustness. Second, the growing value of data and the falling costs of hardware make it advantageous for service providers to trade increasingly inexpensive hardware for the peace of

mind potentially provided by BFT replication. Third, improvements to the state of the art in BFT algorithms^{1,4,6,13,23,26} have narrowed the gap between BFT replication costs and the costs already being paid for non-BFT replication by many commercial services. For example, by default, the Google file system uses three-way replication of storage,⁹ which is roughly the cost of tolerating one Byzantine failure by using three full replicas plus one additional lightweight node to help the replicas coordinate their actions.²⁶

Unfortunately, practitioners hesitate to deploy BFT systems at least partly because of the perception that BFT must impose high overheads. This concern motivates our work, which seeks to answer a simple question: *Can we build a system that tolerates a broad range of faults while meeting the demands of high-performance services?*

To answer this question, this article presents Zyzyva.[†] Zyzyva seeks to make BFT replication deployable for the widest range of practical services by implementing the extremely general abstraction of a replicated state machine at an extremely low cost.

The basic idea of BFT state machine replication is simple: a client sends a request to a replicated service and the service’s distributed agreement protocol ensures that correct servers execute the same requests in the same order.²⁴ If the service is deterministic, each correct replica thus traverses the same series of states and produces the same reply to each request. The servers send their replies back to the client, and the client accepts a reply that matches across a sufficient number of servers.

Zyzyva builds on this basic approach, but reduces its cost through *speculation*. As is common in existing BFT replication protocols, an elected *primary* server proposes an order on client requests to the other server *replicas*.⁴ However, unlike in traditional protocols, Zyzyva replicas then immediately execute requests speculatively, without running an expensive agreement protocol to definitively establish the order. As a result, if the primary is faulty, correct replicas’ states may diverge, and they may send different responses to a client. Nonetheless, Zyzyva preserves correctness because a correct client detects such divergence and avoids acting on a reply until the reply and the sequence of preceding requests are *stable* and guaranteed to eventually be adopted

[†]Zyzyva (ZIZ-uh-vuh) is the last word in most dictionaries. According to dictionary.com, a zyzyva is “any of various South American weevils of the genus *Zyzyva*, often destructive to plants.”

A previous version of this paper was published in *Proceedings of 21st ACM Symposium on Operating Systems Principles (SOSP)*, October 2007, p. 45–58.

*This work was mostly done when the author was at the University of Texas at Austin.

by all correct servers. Thus, applications at correct clients observe the traditional abstraction of a replicated state machine that executes requests in a linearizable¹⁰ order.

Essentially, Zyzzyva “rethinks the sync”¹⁹ for BFT. Whereas past BFT systems have pessimistically enforced the condition that *a correct server only emits replies that are stable*, Zyzzyva recognizes that this condition is stronger than required. Instead, Zyzzyva enforces the weaker condition: *a correct client only acts on replies that are stable*. This change allows us to move the output commit from the servers to the client, which in the optimized case allows servers to avoid expensive all-to-all communication that they would otherwise require to ensure the stronger condition.

Leveraging the client in this way allows us to minimize server overheads and maximize throughputs in the optimized, failure-free case. As a result, Zyzzyva’s peak measured throughput of over 86K requests/second on 3.0GHz Pentium-IV machines makes it feasible to utilize BFT replication in a broad range of demanding services. Despite this aggressive optimization to the fault-free case, Zyzzyva retains good performance of over 82K requests/second even when up to f backup replicas crash. In fact, Zyzzyva’s replication costs, processing overheads, and communication latencies approach their theoretical lower bounds.

2. SYSTEM MODEL

To maximize fault tolerance, BFT replication assumes what is essentially an adversarial failure model. Under this model, faulty nodes (servers or clients) may deviate from their intended behavior in arbitrary ways, representing problems such as hardware faults, software faults, node misconfigurations, or even malicious attacks. This model further assumes a strong adversary that can coordinate faulty nodes to compromise the replicated service. Note, however, that our model assumes the adversary cannot break cryptographic techniques like collision-resistant hashes, encryption, and signatures; we denote a message m signed by principal q ’s public key as $\langle m \rangle_{\sigma_q}$. Zyzzyva ensures its safety and liveness properties if at most f replicas are faulty, and it assumes a finite client population, any number of which may be faulty.

It makes little sense to build a system that can tolerate Byzantine replicas/servers[‡] and clients but that can be corrupted by an unexpectedly slow node or network link, hence we design Zyzzyva so that its safety properties hold in any asynchronous distributed system where nodes operate at arbitrary speeds and are connected by a network that may fail to deliver messages, corrupt them, delay them, or deliver them out of order.

Unfortunately, ensuring both safety and liveness for consensus in an asynchronous distributed system is impossible if any server can crash,⁸ let alone if servers can be Byzantine. Zyzzyva’s liveness, therefore, is ensured only during intervals in which messages sent to correct nodes are processed within some arbitrarily large fixed (but potentially unknown) worst-case delay from when they are sent. This assumption appears easy to meet in practice if broken links are eventually repaired.

Zyzzyva implements a BFT service using state machine replication.^{16,24} Traditional state machine replication techniques

[‡]We use the terms *replica* and *server* interchangeably.

can be applied only to deterministic services. Zyzzyva copes with the nondeterminism present in many real-world applications such as file systems and databases using standard techniques to abstract the observable application state at the replicas and to resolve nondeterministic choices via the agreement stage.²³

If a client of a service issues an erroneous or malicious request, Zyzzyva’s job is to ensure that the request is processed consistently at all correct replicas; the replicated service, itself, is responsible for protecting its application state from such erroneous requests. Services typically limit the damage by authenticating clients and enforcing access control. For example, in a replicated file system, if a client tries to write a file without appropriate credentials, correct replicas could all process the request by returning an error code.

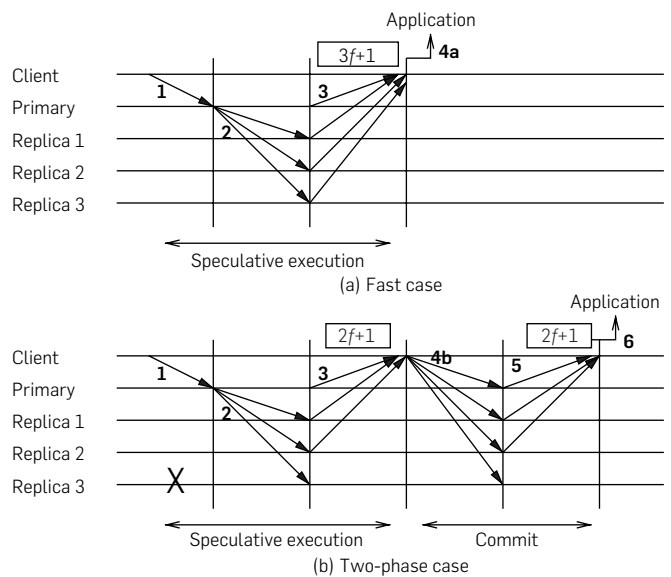
3. AGREEMENT PROTOCOL

Zyzzyva is a state machine replication protocol executed by $3f + 1$ replicas and based on three subprotocols: (1) agreement, (2) view change, and (3) checkpoint. The *agreement* subprotocol orders requests for execution by the replicas. Agreement operates within a sequence of *views*, and in each view a single replica, designated the *primary*, is responsible for leading the agreement subprotocol. The *view change* subprotocol coordinates the election of a new primary when the current primary is faulty or the system is running slowly. The *checkpoint* subprotocol limits the state that must be stored by replicas and reduces the cost of performing view changes.

For simplicity, this article focuses on the agreement subprotocol. The view change and execution subprotocols are similar to those used previously.^{4,26} Interested readers may refer to Kotla et al.¹¹ for the full protocol.

Figure 1 shows the communication pattern for a single instance of Zyzzyva’s agreement subprotocol. In the fast, no-fault case (Figure 1a), a client simply sends a request to the primary, the primary forwards the request to the replicas, and the replicas

Figure 1: Protocol communication pattern for agreement within a view for (a) the fast case and (b) the two-phase faulty replica case. The numbers refer to the main steps of the protocol in the text.



execute the request and send their responses to the client.

A request *completes* at a client when the client has a sufficient number of matching responses to ensure that all correct replicas will eventually execute the request and all preceding requests in the same order, thus guaranteeing that all correct replicas process the request in the same way, issue the same reply, and transition to the same subsequent system state. To allow a client to determine when a request *completes*, a client receives from replicas *responses* that include both an application-level *reply* and the *history* on which the reply depends. The *history* is the sequence of all requests executed by a replica prior to and including this request.

As Figure 1 illustrates, a request *completes* at a client in one of two ways. First, if the client receives $3f + 1$ matching responses (Figure 1a), then the client considers the request *complete* and acts on it. Second, if the client receives between $2f + 1$ and $3f$ matching responses (Figure 1b), then the client gathers $2f + 1$ matching responses and distributes this *commit certificate* to the replicas. A commit certificate includes cryptographic proof that $2f + 1$ servers agree on a linearizable order for the request and all preceding requests, and successfully storing a commit certificate to $2f + 1$ servers (and thus at least $f + 1$ correct servers) ensures that no other ordering can muster a quorum of $2f + 1$ servers to contradict this order. Therefore, once $2f + 1$ replicas acknowledge receiving a commit certificate, the client considers the request *complete* and acts on the corresponding reply.

Zyzyva then ensures the following safety condition:

SAF If a request with sequence number n and history h_n completes, then any request that completes with a higher sequence number $n' \geq n$ has a history $h_{n'}$ that includes h_n as a prefix.

If fewer than $2f + 1$ responses match, then to ensure liveness the client retransmits the request to all replicas at increasing intervals, and replicas demand that the primary orders retransmitted requests. If the primary orders requests too slowly or orders requests inconsistently, a replica will suspect that the primary is faulty. If a sufficient number of replicas suspect that the primary is faulty, then a view change occurs and a new primary is elected.

Zyzyva thereby ensures the following liveness condition assuming eventual synchrony⁸:

LIV Any request issued by a correct client eventually completes.

In the rest of this section, we detail Zyzyva's agreement subprotocol by considering three cases: (1) the *fast case* when all nodes act correctly and no timeouts occur, (2) the *two-phase case* that can occur when a nonprimary replica is faulty or some timeouts occur, and (3) the *view change* case that can occur when the primary is faulty or more serious timeouts occur. Table 1 summarizes the labels we give to fields in messages. Most readers will be happier if on their first reading they skip the text marked "Additional Pedantic Details."

⁸In practice, eventual synchrony⁷ can be achieved by using exponentially increasing timeouts.⁴

Table 1: Labels given to fields in messages.

Label	Meaning
c	Client ID
CC	Commit certificate
d	Digest (cryptographic one-way hash) of client request message: $d = H(m)$
i, j	Server IDs
h_n	History through sequence number h_n encoded as cryptographic one-way hash: $h_n = H(h_{n-1}, d)$
m	Message containing client request
max_n	Maximum sequence number accepted by replica
n	Sequence number
ND	Selection of nondeterministic values needed to execute a request
o	Operation requested by client
OR	Order request message
POM	Proof of misbehavior
r	Application reply to a client operation
t	Time stamp assigned to an operation by a client
u	View number

3.1. Fast case

Figure 1a illustrates the basic flow of messages in the fast case. We trace these messages through the system to explain the protocol, with the numbers in the figure corresponding to the numbers of major steps in the text. As the figure illustrates, the fast case proceeds in four major steps:

1. Client sends request to the primary.
2. Primary receives request, assigns sequence number, and forward ordered request to replicas.
3. Replica receives ordered request, speculatively executes it, and responds to the client.
- 4a. Client receives $3f + 1$ matching responses and completes the request.

To ensure correctness, the messages are carefully constructed to carry sufficient information to link these steps with one another and with past system actions. We now detail the contents of each message and describe the steps each node takes to process each message.

3.1.1. Message processing details

1. Client sends request to the primary.

A client c requests an operation o be performed by the replicated service by sending a message $\langle \text{REQUEST}, o, t, c \rangle_{\sigma_c}$ to the replica it believes to be the primary (i.e., the primary for the last response the client received).

Additional Pedantic Details: If the client guesses the wrong primary, the retransmission mechanisms discussed in step 4c below forward the request to the current primary. The client's time stamp t is included to ensure exactly once semantics of execution of requests.⁴

2. Primary receives request, assigns sequence number, and forwards ordered request to replicas.

A view's primary has the authority to propose the order in which the system should execute requests. It does so by producing ORDER-REQ messages in response to client REQUEST messages.

In particular, when the primary p receives message $m = \langle \text{REQUEST}, o, t, c \rangle_{\sigma_c}$ from client c , the primary assigns to the request a sequence number n in the current view u and relays a message $\langle \langle \text{ORDER-REQ}, u, n, h_n, d, ND \rangle_{\sigma_p}, m \rangle$ to the backup replicas where n and u indicate the proposed sequence number and view number for m , digest $d = H(m)$ is the cryptographic one-way hash of m , $h_n = H(h_{n-1}, d)$ is a cryptographic hash summarizing the history, and ND is a set of values for nondeterministic application variables (time in file systems, locks in databases, etc.) required for executing the request.

Additional Pedantic Details: The primary only takes the above actions if $t > t_c$ where t_c is the highest time stamp previously received from c .

3. Replica receives ordered request, speculatively executes it, and responds to the client.

When a replica receives an ORDER-REQ message, it optimistically assumes that the primary is correct and that other correct replicas will receive the same request with the same proposed order. It therefore speculatively executes requests in the order proposed by the primary and produces a SPEC-RESPONSE message that it sends to the client.

In particular, upon receipt of a message $\langle \langle \text{ORDER-REQ}, u, n, h_n, d, ND \rangle_{\sigma_p}, m \rangle$ from the primary p , replica i accepts the ordered request if m is a well-formed REQUEST message, d is a correct cryptographic hash of m , u is the current view, $n = \max_n + 1$ where \max_n is the largest sequence number in i 's history, and $h_n = H(h_{n-1}, d)$. Upon accepting the message, i appends the ordered request to its history, executes the request using the current application state to produce a reply r , and sends to c a message $\langle \langle \text{SPEC-RESPONSE}, u, n, h_n, H(r), c, t \rangle_{\sigma_i}, i, r, OR \rangle$ where $OR = \langle \text{ORDER-REQ}, u, n, h_n, d, ND \rangle_{\sigma_p}$.

Additional Pedantic Details: A replica may only accept and speculatively execute requests in sequence-number order, but message loss or a faulty primary can introduce holes in the sequence number space. Replica i discards the ORDER-REQ message if $n \leq \max_n$. If $n > \max_n + 1$, then i discards the message, sends a message $\langle \text{FILL-HOLE}, u, \max_n + 1, n, i \rangle_{\sigma_i}$ to the primary, and starts a timer. Upon receipt of a message $\langle \text{FILL-HOLE}, u, k, n, i \rangle_{\sigma_i}$ from replica i , the primary p sends a $\langle \langle \text{ORDER-REQ}, u, n', h_n, d, ND \rangle_{\sigma_p}, m' \rangle$ to i for each request m' that p ordered in $k \leq n' \leq n$ during the current view; the primary ignores fill-hole requests from other views. If i receives the valid ORDER-REQ messages

needed to fill the holes, it cancels the timer. Otherwise, the replica broadcasts the FILL-HOLE message to all other replicas and initiates a view change when the timer fires. Any replica j that receives a FILL-HOLE message from i sends the corresponding ORDER-REQ message, if it has received one. If, in the process of filling in holes in the replica sequence, replica i receives conflicting ORDER-REQ messages, then the conflicting messages form a proof of misbehavior (POM) as described in protocol step 4d.

4a. Client receives $3f + 1$ matching responses and completes the request.

In the absence of faults and timeouts, the client receives matching SPEC-RESPONSE messages from all $3f + 1$ replicas. The client can then consider the request and its history to be *complete* and delivers the reply r to the application.

$3f + 1$ identical replies with identical histories suffice to ensure that a client can rely on a response. In particular, $3f + 1$ matching responses means all correct servers have executed the request and all preceding requests in the same order, so correct servers can always form a majority to vote to keep this response, even across view changes.¹¹ In particular, the view change subprotocol executes across $2f + 1$ responsive servers, but any group of $2f + 1$ servers must include at least $f + 1$ correct servers and at most f faulty servers. Thus, the correct servers are always able to vote to keep this response, including both the application reply and the history of previous actions.

Therefore, upon receiving $3f + 1$ distinct messages $\langle \langle \text{SPEC-RESPONSE}, u, n, h_n, H(r), c, t \rangle_{\sigma_i}, i, r, OR \rangle$, where i identifies the replica issuing the response, a client determines if they match. SPEC-RESPONSE messages from distinct replicas *match* if they have identical $u, n, h_n, H(r), c, t, OR$, and r fields.

3.2. Two-phase case

If the network, primary, or some replicas are slow or faulty, the client c may not receive matching responses from all $3f + 1$ replicas. The two-phase case applies when the client receives between $2f + 1$ and $3f$ matching responses. As Figure 1b illustrates, steps 1–3 occur as described above, but step 4 is different:

4b. Client receives between $2f + 1$ and $3f$ matching responses, assembles a commit certificate, and transmits the commit certificate to the replicas.

The commit certificate is cryptographic proof that a majority of correct servers agree on the ordering of requests up to and including the client's request. Protocol steps 5 and 6 complete the second phase of agreement by ensuring that enough servers have this proof.

5. Replica receives a COMMIT message from a client containing a commit certificate and acknowledges with a LOCAL-COMMIT message.

6. Client receives a LOCAL-COMMIT messages from $2f + 1$ replicas and completes the request.

Again, the details of message construction and processing

are designed to allow clients and replicas to link the system's actions together into a single linearizable history.

3.2.1. Message processing details

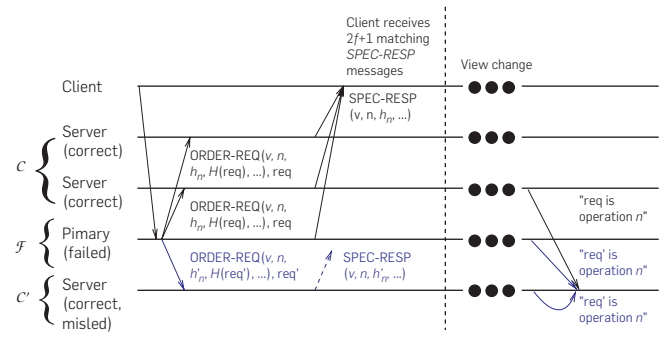
4b. Client receives between $2f + 1$ and $3f$ matching responses, assembles a commit certificate, and transmits the commit certificate to the replicas.

A client c sets a timer when it first issues a request. When this timer expires, if c has received matching speculative responses from between $2f + 1$ and $3f$ replicas, then c has a proof that a majority of correct replicas agree on the order in which the request should be processed. Unfortunately, the replicas, themselves, are unaware of this quorum of matching responses—they only know of their local decision, which may not be enough to guarantee that the request completes in this order.

Figure 2 illustrates the problem. A client receives $2f + 1$ matching speculative responses indicating that a request req was executed as the n th operation in view u . Let these responses come from $f + 1$ correct servers C ; and f faulty servers F ; and assume the remaining f correct servers C' received an ORDER-REQ message from a faulty primary proposing to execute a different request req' at sequence number n in view u . Suppose a view change occurs at this time. The view change subprotocol must determine what requests were executed with what sequence numbers in view u so that the state in view $u + 1$ is consistent with the state in view u . Furthermore, since up to f replicas may be faulty, the view change subprotocol must be able to complete using information from only $2f + 1$ replicas. Suppose now that the $2f + 1$ replicas contributing state to a view change operation are one correct server from C , f faulty servers from F , and f correct but misled servers from C' . In this case, only one of the replicas initializing the new view is guaranteed to vote to execute req as operation n in the new view, while as many as $2f$ replicas may vote to execute req' in that position. Thus, the system cannot ensure that view $u + 1$'s state reflects the execution of req as the operation with sequence number n .

Before client c can rely on this response, it must take additional steps to ensure the stability of this response. The client

Figure 2: Example of a problem that could occur if a client were to rely on just $2f + 1$ matching responses without depositing a commit certificate with the servers.



therefore sends a message $\langle \text{COMMIT}, c, CC \rangle_{\sigma_c}$ where CC is a commit certificate consisting of a list of $2f + 1$ replicas, the replica-signed portions of the $2f + 1$ matching SPEC-RESPONSE messages from those replicas, and the corresponding $2f + 1$ replica signatures.

Additional Pedantic Details: CC contains $2f + 1$ signatures on the SPEC-RESPONSE message and a list of $2f + 1$ nodes, but, since all the responses received by c from replicas are identical, c only needs to include *one* replica-signed portion of the SPEC-RESPONSE message. Also note that, for efficiency, CC does not include the body r of the reply but only the hash $H(r)$.

5. Replica receives a COMMIT message from a client containing a commit certificate and acknowledges with a LOCAL-COMMIT message.

When a replica i receives a message $\langle \text{COMMIT}, c, CC \rangle_{\sigma_c}$ containing a valid commit certificate CC proving that a request should be executed with a specified sequence number and history in the current view, the replica first ensures that its local history is consistent with the one certified by CC . If so, replica i (1) stores CC if CC 's sequence number exceeds the stored certificate's sequence number and (2) sends a message $\langle \text{LOCAL-COMMIT}, u, d, h, i, c \rangle_{\sigma_i}$ to c .

Additional Pedantic Details: If the local history simply has holes encompassed by CC 's history, then i fills them as described in 3. If, however, the two histories contain different requests for the same sequence number, then i initiates the view change subprotocol. Note that as the view change protocol executes, correct replicas converge on a single common history, and those replicas whose local state reflect the "wrong" history (e.g., because they speculatively executed the "wrong" requests) restore their state from a cryptographically signed distributed checkpoint.¹¹

6. Client receives a LOCAL-COMMIT messages from $2f + 1$ replicas and completes the request.

The client resends the COMMIT message until it receives corresponding LOCAL-COMMIT messages from $2f + 1$ distinct replicas. The client then considers the request to be complete and delivers the reply r to the application.

$2f + 1$ local commit messages suffice to ensure that a client can rely on a response. In particular, at least $f + 1$ correct servers store a commit certificate for the response, and since any commit or view change requires participation by at least $2f + 1$ of the $3f + 1$ servers, any subsequent committed request or view change includes information from at least one correct server that holds this commit certificate. Since the commit certificate includes $2f + 1$ signatures vouching for the response, even a single correct server can use the commit certificate to convince all correct servers to accept this response (including the application reply and the history.)

Additional Pedantic Details: When the client first sends the COMMIT message to the replicas it starts a timer. If this timer

expires before the client receives $2f + 1$ LOCAL-COMMIT messages then the client moves on to protocol steps described in the next subsection.

3.2.2. Client trust

At first glance, it may appear imprudent to rely on clients to transmit commit certificates to replicas (4b): what if a faulty client sends an altered commit certificate (threatening safety) or fails to send a commit certificate (imperiling liveness)?

Safety is ensured even if clients are faulty because commit certificates are authenticated by $2f + 1$ replicas. If a client alters a commit certificate, correct replicas will ignore it.

Liveness is ensured for correct clients because commit certificates are cumulative: successfully storing a commit certificate for request n at $2f + 1$ replicas commits those replicas to a linearizable total order for all requests up to request n . So, if a faulty client fails to deposit a commit certificate, that client may not learn when its request *completes*, and a replica whose state has diverged from its peers may not immediately discover this fact. However, if at any future time a correct client issues a request, that request (and a linearizable history of earlier requests on which it depends) will either (1) complete via $3f + 1$ matching responses (4a), (2) complete via successfully storing a commit certificate at $2f + 1$ replicas (4b–6), or (3) trigger a view change (4c or 4d below).

3.3. Timeout and view change cases

Cases 4a and 4b allow a client c 's request to complete with $2f + 1$ to $3f + 1$ matching responses. However, if the primary or network is faulty, c may not receive matching SPEC-RESPONSE or LOCAL-COMMIT messages from even $2f + 1$ replicas. Cases 4c and 4d therefore ensure that a client's request either completes in the current view or that a new view with a new primary is initiated. In particular, case 4c is triggered when a client receives fewer than $2f + 1$ matching responses and case 4c occurs when a client receives responses indicating inconsistent ordering by the primary.*

4c. Client receives fewer than $2f + 1$ matching SPEC-RESPONSE messages and resends its request to all replicas, which forward the request to the primary in order to ensure that the request is assigned a sequence number and eventually executed.

A client sets a second timer when it first issues a request. If the second timer expires before the request *completes*, the client suspects that the primary may not be ordering requests as intended, so it resends its REQUEST message through the remaining replicas so that they can track the request's progress and, if progress is not satisfactory, initiate a view change. This case can be understood by examining the behavior of a nonprimary replica and of the primary.

Replica. When nonprimary replica i receives a message $\langle \text{REQUEST}, o, t, c \rangle_{\sigma_c}$ from client c , then if the request has a higher time stamp than the currently cached response for

*Note that cases 4b and 4c are not exclusive of 4d; a client may receive messages that are both sufficient to complete a request and also a proof of misbehavior against the primary.

that client, i sends a message $\langle \text{CONFIRM-REQ}, u, m, i \rangle_{\sigma_i}$ where $m = \langle \text{REQUEST}, o, t, c \rangle_{\sigma_c}$ to the primary p and starts a timer. If the replica accepts an ORDER-REQ message for this request before the timeout, it processes the ORDER-REQ message as described above. If the timer fires before the primary orders the request, the replica initiates a view change.

Primary. Upon receiving the message $\langle \text{CONFIRM-REQ}, u, m, i \rangle_{\sigma_i}$ from replica i , the primary p checks the client's time stamp for the request. If the request is new, p sends a new ORDER-REQ message using a new sequence number as described in step 2.

Additional Pedantic Details: If replica i does not receive the ORDER-REQ message from the primary, the replica sends the CONFIRM-REQ message to all other replicas. Upon receipt of a CONFIRM-REQ message from another replica j , replica i sends the corresponding ORDER-REQ message it received from the primary to j ; if i did not receive the request from the client, i acts as if the request came from the client itself. To ensure eventual progress, a replica doubles its current timeout in each new view and resets it to a default value if a view succeeds in executing a request.

Additionally, to retain exactly once semantics, replicas maintain a cache that stores the reply to each client's most recent request. If a replica i receives a request from a client and the request matches or has a lower client-supplied time stamp than the currently cached request for client c , then i simply resends the cached response to c . Similarly, if the primary p receives an old client request from replica i , p sends to i the cached ORDER-REQ message for the most recent request from c . Furthermore, if replica i has received a commit certificate or stable checkpoint for a subsequent request, then the replica sends a LOCAL-COMMIT to the client even if the client has not transmitted a commit certificate for the retransmitted request.

4d. Client receives responses indicating inconsistent ordering by the primary and sends a proof of misbehavior to the replicas, which initiate a view change to oust the faulty primary.

If client c receives a pair of SPEC-RESPONSE messages containing valid messages $OR = \langle \text{ORDER-REQ}, u, n, h_n, d, ND \rangle_{\sigma_j}$ for the same request ($d = H(m)$) in the same view u with differing sequence number n or history h_n or ND , then the pair of ORDER-REQ messages constitutes a proof of misbehavior² (POM) against the primary. Upon receipt of a POM, c sends a message $\langle \text{POM}, u, POM \rangle_{\sigma_c}$ to all replicas. Upon receipt of a valid POM message, a replica initiates a view change and forward the POM message to all other replicas.

4. EVALUATION

This section examines the performance of Zyzzyva and compares it with existing approaches. We run our experiments on 3.0GHz Pentium-4 machines with the Linux 2.6 kernel. We use MD5 for hashing and UMAC⁴ for message authentication codes (MACs). MD5 is known to be vulnerable, but we use it to make our results comparable with those in the literature. Since Zyzzyva uses fewer MACs per request than any of the competing algorithms, our advantages over other

algorithms would be increased if we were to use the more secure, but more expensive, SHA-256.

For comparison, we run Castro and Liskov’s implementation of Practical Byzantine Fault Tolerance (PBFT)⁴ and Cowling et al.’s implementation of hybrid quorum (HQ)⁶; we scale-up HQ’s measured throughput for the small request/response benchmark by 9% to account for their use of SHA-1 rather than MD5. We include published throughput measurements for Q/U¹; we scale Q/U’s reported performance up by 7.5% to account for our use of 3.0 GHz rather than 2.8 GHz machines. We also compare against the measured performance of an unreplicated server.

To stress-test Zyzzyva we use the microbenchmarks devised by Castro and Liskov⁴ In the 0/0 benchmark, clients send null requests and receive null replies. In the 4/0 benchmark, clients send 4KB requests and receive a null replies. In the 0/4 benchmark, clients send null requests and receive 4KB replies. In all experiments, we configure all BFT systems to tolerate $f = 1$ faults; we examine performance for other configurations elsewhere.¹¹

In the preceding sections, we describe a simplified version of the protocol. In our extended paper,¹² we detail a number of optimizations, all implemented in the prototype measured here, that (1) reduce encryption costs by replacing public key signatures with MACs,⁴ (2) improve throughput by agreeing on the order of batches of requests,⁴ (3) reduce the impact of lost messages by caching out-of-order messages, (4) improve read performance by optimizing read-only requests,⁴ reduce bandwidth by allowing most replicas to send hashes rather than full replies to clients,⁴ (5) improve the performance of Zyzzyva’s two-phase case by using a commit optimization in which replicas use a client hint to initiate and complete the second phase to commit the request before they execute the request and send the response (with the committed history) back to the client, and (6) reduce overheads by including MACs only for a preferred quorum.⁶ In the extended paper we also describe Zyzzyva5, a variation of the protocol that requires $5f + 1$ agreement replicas but that improves performance in the presence of faulty replicas by completing in three one-way message exchanges as in Figure 1a even when up to f nonprimary replicas are faulty.

In the following experiments, unless noted otherwise, we use all of the optimizations other than preferred quorums for Zyzzyva. PBFT⁴ does not implement the preferred quorum optimization, but HQ does.⁶ We do not use the read-only optimization for Zyzzyva and PBFT unless we state so explicitly.

4.1. Cost model

Our evaluation focuses on three metrics that BFT replication must optimize to be practical for a broad range of services: replication cost, throughput, and latency. Before we dive into experimental evaluation in the following sections, Table 2 puts our results in perspective by providing a high-level analytic model of Zyzzyva and of several other recent BFT protocols. The table also shows lower bounds on BFT state machine replication overheads for each of these dimensions.

In the first row of the table body, replication *cost* refers to the number of replicas required to construct a system that tolerates f Byzantine faults. The importance of minimizing this metric for practical services is readily apparent. We show two values: *replicas with application state* indicates the number of replicas that must both participate in the coordination protocol and also maintain application state for executing application requests. Conversely, *total replicas* indicates the total number of machines that must participate in the protocol including, for some protocols, “witness nodes” that do not maintain application state or execute application requests. This distinction is important because witness nodes may be simpler or less expensive than nodes that must also execute requests to run the replicated service.

Zyzzyva and PBFT (with Yin et al.’s optimization for separating agreement and execution²⁶) meet the replication cost lower bounds of $2f + 1$ application replicas (so a majority of nodes are correct)²⁴ and $3f + 1$ total replicas (so agreement on request order can be reached).²¹

In the next row of the table body, *throughput* is determined by the processing overhead per request. Our simple model focuses on CPU intensive cryptographic operations. All of the systems we examine use Castro’s MAC authenticator construct⁴ to avoid using expensive asymmetric cryptography operations.

Table 2: Properties of state-of-the-art and optimal Byzantine fault-tolerant (BFT) service replication systems tolerating f faults, using MACS for authentication,⁴ assuming preferred quorum optimization, and using a batch size of b .⁴

		PBFT ⁴	Q/U ¹	HQ ⁶	Zyzzyva	State Machine Replication Lower Bound
Cost	Total replicas	$3f + 1$	$5f + 1$	$3f + 1$	$3f + 1$	$3f + 1$ ²¹
	Replicas with application state	$2f + 1$ ²⁶	$5f + 1$	$3f + 1$	$2f + 1$	$2f + 1$ ²⁴
Throughput	MAC ops at bottleneck server	$2 + (8f + 1)/b$	$2 + 8f$	$4 + 4f$	$2 + 3f/b$	2 ^a
Latency	NW one-way latencies on critical path	4	2	4	3	2 or 3 ^b

Bold entries denote protocols that match known lower bounds or those with the lowest known cost.

^a It is not clear that this trivial lower bound is achievable.

^b The distributed systems literature typically considers three one-way latencies to be the lower bound for agreement on client requests¹⁷; two one-way latencies is achievable if no request contention is assumed.

The (trivial) lower bound on processing overhead is for each server to process two MAC operations per client request: one to verify the client's request and one to authenticate its reply. *Zyzyva* and PBFT approach this bound by using a *batching* optimization in which the primary accumulates a batch of b client requests and leads agreement on that batch rather than on each individual request. *Zyzyva*'s speculative execution allows it to avoid several rounds of all-to-all communication among servers, so it requires fewer MAC operations per batch than PBFT.

In the last row of the table body, *latency* counts the number of one-way message delays from when a client issues a request until it receives a reply. In the general case, agreement requires three message delays,¹⁷ and *Zyzyva* matches this bound by having requests go from the client to the primary to the replicas to the client. Q/U circumvents this bound by optimizing for the case of no request contention so that requests go directly from the client to the replicas to the client. We chose to retain the extra hop through the primary in *Zyzyva* because it facilitates batching, which we consider to be an important throughput optimization.

The models described in this subsection focus on what we regard as important factors for understanding the performance trade-offs of different algorithms, but they necessarily omit details present in implementations. Also, as is customary,^{1,4,6,23,26} Table 2 compares the protocols' performance during the optimized case of fault-free, timeout-free execution. In the rest of this section we experimentally examine these protocols' throughput, latency, and performance during failures.

4.2. Throughput

Figure 3 shows the throughput measured for the 0/0 benchmark for *Zyzyva*, *Zyzyva5*,¹¹ PBFT, and HQ (scaled as noted above). For reference, we also show the peak throughput reported for Q/U¹ in the $f = 1$ configuration, scaled to our environment as described above.

Zyzyva executes over 50K requests/second without batching, and this number rises to 86K requests/second when batching is activated with 10 requests per batch. As the

Figure 3: Realized throughput for the 0/0 benchmark as the number of client varies. Q/U throughput is scaled from¹.

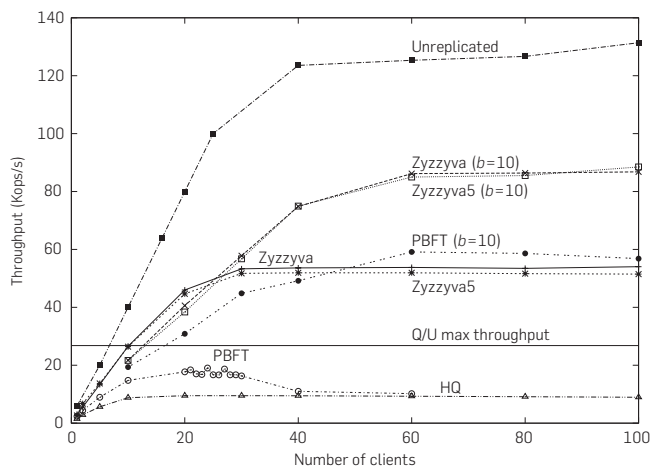


figure illustrates, *Zyzyva* enjoys a significant throughput advantage over the other protocols.

It is also worth noting that when batching is enabled, *Zyzyva*'s throughput is within 35% of the throughput of an unreplicated server that simply replies to client requests over an authenticated channel. Furthermore, this gap would fall if the service being replicated were more demanding than the null service examined here. Overall, we speculate that *Zyzyva*'s throughput is sufficient to support BFT replication for a broad range of demanding services.

4.3. Latency

Figure 4 shows the latencies of *Zyzyva*, *Zyzyva5*, HQ, and PBFT for the 0/0, 0/4, and 4/0 workloads with a single client issuing one request at a time. We examine both the default read/write requests that use the full protocol and read-only requests that can exploit *Zyzyva* and PBFT's read-only optimization.⁴

We did not succeed in getting Abd-El-Malek et al.'s implementation of Q/U running in our environment. However, because Table 2 suggests that Q/U may have a latency advantage over other protocols, for comparison we implement an idealized model of Q/U designed to provide an optimistic estimate of Q/U's latency in our environment. In our idealized implementation, a client simply generates and sends $4f + 1$ MACs with a request, each replica verifies $4f + 1$ MACs (1 to authenticate the client and $4f + 1$ to validate the reported state), each replica in a preferred quorum⁶ generates and sends $4f + 1$ MACs (1 to authenticate the reply to the client and $4f$ to authenticate the new state) with a reply to the client, and the client verifies $4f + 1$ MACs.

For the read/write 0/0 and 4/0 benchmarks, Q/U does have a modest latency advantage over *Zyzyva* as predicted by Table 2. For the read-only benchmarks, the situation is reversed with *Zyzyva* exhibiting modestly lower latency than Q/U because *Zyzyva*'s read-only optimization completes read-only requests in two message delays (like Q/U) but uses fewer cryptographic operations.

Figure 5 shows latency and throughput as we vary offered load for the 0/0 benchmark. As the figure illustrates, batching in *Zyzyva*, *Zyzyva5*, and PBFT increases latency but also increases

Figure 4: Latency for 0/0, 0/4, and 4/0 benchmarks.

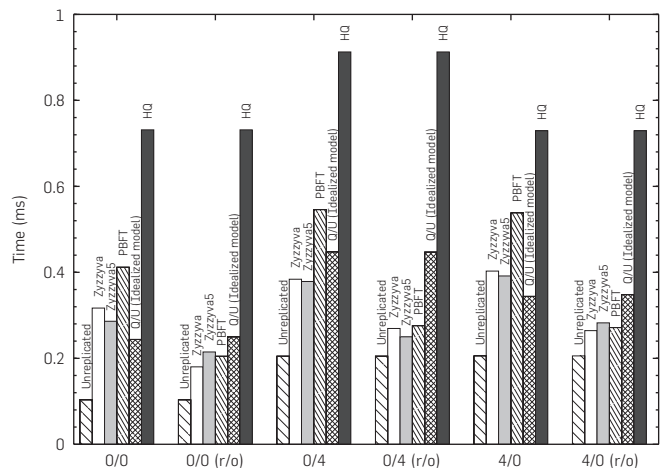
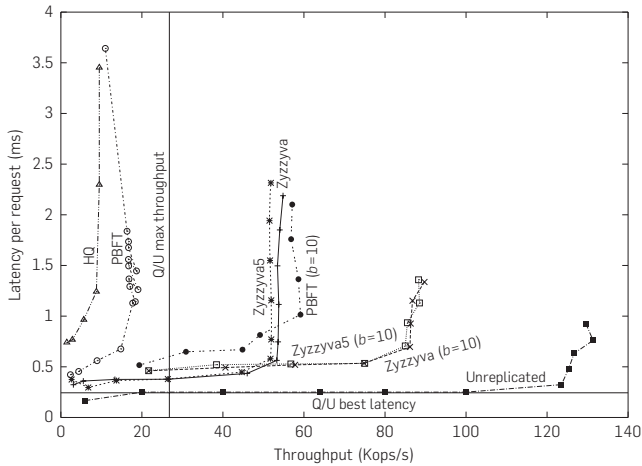


Figure 5: Latency versus throughput for the O/O benchmark. Q/U throughput is scaled from 1. Q/U best latency is the measured latency for our idealized model implementation of Q/U under low offered load.



peak throughput. Adaptively setting the batch size in response to workload characteristics is an avenue for future work.

Overall, all of the BFT protocols do add service latency compared to an unreplicated server, but Zyzyva is generally competitive with the best protocols by this metric. We speculate that the additional 120 to 250 microseconds that Zyzyva requires compared to an unreplicated server will be a significant barrier for only the most demanding services, and we note that the relative gap would shrink for services that do more than execute the null request.

4.4. Performance during failures

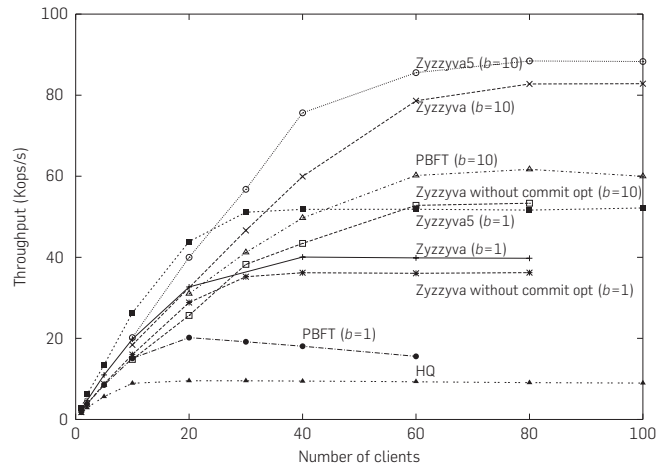
Zyzyva guarantees correct execution with any number of faulty clients and up to f faulty replicas. However, its performance is optimized for the case of failure-free operation, and a single faulty replica can force Zyzyva to execute the slower two-phase protocol.

One solution is to buttress Zyzyva’s fast one-phase path by employing additional servers. Zyzyva5¹¹ uses a total of $5f + 1$ servers ($2f + 1$ full replicas and $3f$ additional witnesses) to allow the system to complete requests via the fast communication pattern shown in Figure 1a when the client receives $4f + 1$ (out of $5f + 1$) matching replies.

Surprisingly, however, even when running with $3f + 1$ replicas, Zyzyva remains competitive with existing protocols even when it falls back on two-phase operation. In particular, Zyzyva’s cryptographic overhead at the bottleneck replica increases from $2 + ((3f + 1)/b)$ to $3 + ((5f + 1)/b)$ operations per request if we simply execute the two-phase algorithm described above.** Furthermore, our implementation also includes a *commit optimization*¹² that reduces cryptographic overheads to $2 + ((5f + 1)/b)$ cryptographic operations per request by having replicas that suspect a faulty primary initiate and complete the second phase to commit the request before they execute the request and send the response (with

** As noted at the start of this section we omit the preferred quorums optimization in our experimental evaluations, so the $2 + (3f+1)b$ MAC operations per request in our measurements is higher than the $2 + 3f/b$ listed in Table 2.

Figure 6: Realized throughput for the O/O benchmark as the number of clients varies when $f = 1$ nonprimary replicas fail to respond to requests.



the committed history) back to the client.

Figure 6 compares throughputs of Zyzyva, Zyzyva5, PBFT, and HQ in the presence of f nonprimary-server fail-stop failures. We do not include a discussion of Q/U in this section as the throughput numbers of Q/U with failures are not reported,¹ but we would not expect a fail-stop failure by a replica to significantly reduce the performance shown for Q/U in Figure 3. Also, we do not include a line for the unreplicated server case as the throughput falls to zero when the only server suffers a fail-stop failure.

As Figure 6 shows, without the commit optimization, falling back on two-phase operation reduces Zyzyva’s maximum throughput from 86K requests/second (Figure 3) to 52K requests/second. Despite this extra overhead, Zyzyva’s “slow case” performance remains within 13% of PBFT’s performance, which is less highly tuned for the failure-free case and which suffers no slowdown in this scenario. Zyzyva’s commit optimization repairs most of the damage caused by a fail-stop replica, maintaining a throughput of 82K requests/second which is within 5% of the peak throughput achieved for the failure-free case. For systems that can afford extra witness replicas, Zyzyva5’s throughput is not significantly affected by the fail-stop failure of a replica, as expected.

5. RELATED WORK

Zyzyva stands on the shoulders of recent efforts that have dramatically cut the costs and improved the practicality of BFT replication. Castro and Liskov’s PBFT protocol⁴ devised techniques to eliminate expensive signatures and potentially fragile timing assumptions, and it demonstrated high throughputs of over 10K requests/second. This surprising result jump started an arms race in which researchers reduced replication costs,²⁶ and improved performance^{1,6,13} of BFT service replication. Zyzyva incorporates many of the ideas developed in these protocols and folds in the new idea of speculative execution to construct an optimized fast path that significantly outperforms existing protocols and that has replication cost, processing overhead, and latency that approach the theoretical minima for these metrics.

Numerous BFT agreement protocols^{4,6,13,18,23,26} have used *tentative execution* to reduce the latency experienced by clients. This optimization allows replicas to execute a request tentatively as soon as they have collected the equivalent of a Zyzzyva commit certificate for that request. This optimization may appear similar to Zyzzyva's support for *speculative execution*, but there are two fundamental differences. First, Zyzzyva's speculative execution allows requests to complete at a client after a single phase, without the need to compute a commit certificate: this reduction in latency is not possible with traditional tentative executions. Second, and more importantly, in traditional BFT systems a replica can execute a request tentatively only after the replica knows that all previous requests have been committed. In Zyzzyva, replicas continue to execute requests speculatively, without waiting to know that requests with lower sequence numbers have completed. This difference is what lets Zyzzyva leverage speculation to achieve not just lower latency but also higher throughput.

Speculator²⁰ allows clients to speculatively complete operations at the application level and perform client-level roll-back. A similar approach could be used in conjunction with Zyzzyva to support clients that want to act on a reply optimistically rather than waiting on the specified set of responses.

Zyzzyva's focus is on maximizing the peak performance of BFT replication. Conversely, Clement et al.⁵ argue that BFT systems should seek not only to ensure safety but also good performance in the presence of Byzantine faults, and their Aardvark system eliminates *fragile optimizations* that maximize best-case performance but that can allow a faulty client or server to drive the system down expensive execution paths.

6. CONCLUSION

By systematically exploiting speculation, Zyzzyva exhibits significant performance improvements over existing BFT protocols. The throughput overheads and latency of Zyzzyva approach the theoretical lower bounds for any BFT state machine replication protocol.

Looking forward, although we expect continued progress in improving the performance (for example, by making additional assumptions about the application characteristics) and robustness (in the presence of broader range of failure scenarios) of BFT replication, we believe that Zyzzyva demonstrates that BFT overheads should no longer be regarded as a barrier to using BFT replication for even many highly demanding services. acknowledgments

Acknowledgments

This work was supported in part by NSF grants CNS-0720649, CNS-0509338, and CNS-0411026. We thank Rodrigo Rodrigues, James Cowling, and Michael Abd-El-Malek for sharing source code for PBFT, HQ, and Q/U, respectively. We are grateful for Maurice Herlihy's feedback on earlier drafts of this article. □

References

- Abd-El-Malek, M., Ganger, G., Goodson, G., Reiter, M., and Wylie, J. Fault-scalable Byzantine fault-tolerant services. *Proceedings of the Symposium on Operating Systems Principles*, October 2005.
- Aiyer, A. S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.-P., and Porth, C. BAR fault tolerance for cooperative services. *Proceedings of the Symposium on Operating Systems Principles*, pp. 45–58, October 2005.

- Amazon s3 availability event: July 20, 2008. <http://status.aws.amazon.com/s3-20080720.html>.
- Castro, M., and Liskov, B. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, November 2002.
- Clement, A., Marchetti, M., Wong, E., Alvisi, L., and Dahlin, M. Making Byzantine fault tolerant services tolerate Byzantine faults. Technical Report 08-27, UT Austin Department of Computer Sciences, May 2008.
- Cowling, J., Myers, D., Liskov, B., Rodrigues, R., and Shrira, L. HQ replication: A hybrid quorum protocol for Byzantine fault tolerance. *Proceedings of the Symposium on Operating Systems Design and Implementation*, November 2006.
- Dwork, C., Lynch, N., and Stockmeyer, L. Consensus in the presence of partial synchrony. *Journal of the ACM*, 1988.
- Fischer, M., Lynch, N., and Paterson, M. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, April 1985.
- Ghemawat, S., Gobioff, H., and Leung, S. The Google File System. *Proceedings of 19th ACM Symposium on Operating Systems Principles*, October 2003.
- Herlihy, M., and Wing, J. Linearizability: A correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3), 1990.
- Kotla, R., Alvisi, L., Dahlin, M., Clement, A., and Wong, E. Zyzzyva: Speculative Byzantine fault tolerance. *ACM Symposium on Operating Systems Principles*, 2007.
- Kotla, R., Alvisi, L., Dahlin, M., Clement, A., and Wong, E. Zyzzyva: Speculative Byzantine fault tolerance. Technical Report UTCS-TR-07-40, University of Texas at Austin, 2007.
- Kotla, R. and Dahlin, M. High-throughput Byzantine fault tolerance. *Proceedings of the International Conference on Dependable Systems and Networks*, June 2004.
- Kotla, R., Dahlin, M., and Alvisi, L. SafeStore: A durable and practical storage system. *Proceedings of the USENIX Annual Technical Conference*, June 2007.
- Lamport, L., Shostak, R., and Pease, M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3): 382–401, July 1982.
- Lamport, L. Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems*, 6(2):254–280, April 1984.
- Lamport, L. Lower bounds for asynchronous consensus. *Proceedings of the FUDICO*, pp. 22–23, June 2003.
- Martin, J.-P. and Alvisi, L. Fast Byzantine consensus. *IEEE TODS*, 3(3):202–215, July 2006.
- Nightingale, E., Veeraraghavan, K., Chen, P., and Flinn, J. Rethink the sync. *Proceedings of the Symposium on Operating Systems Design and Implementation*, 2006.
- Nightingale, E. B., Chen, P., and Flinn, J. Speculative execution in a distributed file system. *Proceedings of the Symposium on Operating Systems Principles*, October 2005.
- Pease, M., Shostak, R., and Lamport, L. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, April 1980.
- Prabhakaran, V., Bairavasundaram, L., Agrawal, N., Arpaci-Dusseau, H. G. A., and Arpaci-Dusseau, R. IRON file systems. *Proceedings of the Symposium on Operating Systems Principles*, 2005.
- Rodrigues, R., Castro, M., and Liskov, B. BASE: Using abstraction to improve fault tolerance. *Proceedings of the Symposium on Operating Systems Principles*, October 2001.
- Schneider, F. B. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, 22(4):299–319, 1990.
- Yang, J., Sar, C., and Engler, D. Explode: A lightweight, general system for finding serious storage system errors. *Proceedings of the Symposium on Operating Systems Design and Implementation*, 2006.
- Yin, J., Martin, J.-P., Venkataramani, A., Alvisi, L., and Dahlin, M. Separating agreement from execution for Byzantine fault tolerant services. *Proceedings of the Symposium on Operating Systems Principles*, October 2003.

Ramakrishna Kotla (kotla@microsoft.com)
Microsoft Research Silicon Valley, Mountain View, CA.

Allen Clement (aclement@cs.utexas.edu)
Department of Computer Sciences, University of Texas, Austin.

Edmund Wong (elwong@cs.utexas.edu)
Department of Computer Sciences, University of Texas, Austin.

Lorenzo Alvisi (lorenzo@cs.utexas.edu)
Department of Computer Sciences, University of Texas, Austin.

Mike Dahlin (dahlin@cs.utexas.edu)
Department of Computer Sciences, University of Texas, Austin.

CAREERS

California State University, Fullerton

The Department of Computer Science invites applications for tenure-track positions at the level of Assistant Professor, starting fall 2009. The department's instructional programs lead to bachelors and masters degrees. CSUF is an urban university located in Orange County, surrounded by high technology industry and a diverse community. For a complete description of the position and qualifications, go to <http://diversity.fullerton.edu/>.

Colorado State University Tenure-Track Faculty in Systems/Software/ High-Performance Computing Department of Computer Science

The Department of Computer Science at Colorado State University solicits applications for a tenure-track faculty position preferably at the level of assistant professor, beginning Fall 2009. Applicants must have a Ph.D. in computer science or a related field, and demonstrate potential for excellence in research and teaching in the area of Systems Software/High Performance Computing, which includes distributed systems, virtualization, concurrent systems, storage systems, and the hardware/software interface with an emphasis on software.

The department has over 325 undergraduate majors and 165 graduate students in Master's and Ph.D. programs. The department has 19 tenure-track faculty with strong research programs in artificial intelligence, computer vision, distributed computation, embedded systems, networks, security, and software engineering.

Colorado State University is located in Fort Collins, at the base of the Rocky Mountains. Fort Collins was ranked first in "Best Places to Live" in the western U.S. among small cities by Money Magazine. More information can be obtained at <http://www.cs.colostate.edu>.

Applications must be received by January 7, 2009 at <http://www.natsci.colostate.edu/searches/compsci/> to ensure full consideration.

The anticipated start date is August 16, 2009. Complete applications of semi-finalists will be reviewed by all faculty in the Department.

CSU is an EO/AA employer.

Duke University Department of Computer Science

The Department of Computer Science at Duke University invites applications and nominations for faculty positions at all levels, to begin August 2009. We are interested in strong candidates in all active research areas of computer science, both core and interdisciplinary areas, including algorithms, artificial intelligence, computational biology, computational economics, computer architecture, computer vision, database systems, distributed systems, machine learning, networking, and security.

The department is committed to increasing the diversity of its faculty, and we strongly encourage applications from women and minority candidates.

A successful candidate must have a solid disciplinary foundation and demonstrate promise of outstanding scholarship in every respect, including research and teaching. Please refer to www.cs.duke.edu for information about the department.

Applications should be submitted online at www.cs.duke.edu/facsearch. A Ph.D. in computer science or related area is required. To guarantee full consideration, applications and letters of reference should be received by January 4, 2009.

Durham, Chapel Hill, and the Research Triangle of North Carolina are vibrant, diverse, and thriving communities, frequently ranked among the best places in the country to live and work. Duke and the many other universities in the area offer a wealth of education and employment opportunities for spouses and families.

Drake University Department of Mathematics and Computer Science

Seeks an outstanding teacher and promising scholar for tenure-track Assistant Professor, starting August 2009. Ph.D. or near completion in computer science or a related area. In addition to teaching, also expected to help with curricular design and development of first year seminars. Interdisciplinary experience a plus. For further information see Faculty Search at: <http://www.drake.edu/artsci/mathcs/>

Drake University is an equal-opportunity employer.

Fairfield University Department of Mathematics and Computer Science Faculty Position

The Department of Mathematics and Computer Science at Fairfield University invites applications for one tenure track position in computer science, at the rank of assistant professor, to begin in September 2009. We seek a highly qualified candidate with demonstrated excellence in and enthusiasm for teaching, a desire to contribute to the culture and development of a small program, and evidence of research potential. A doctorate in computer science is required. The teaching load is 3 courses/9 credit hours per semester.

The successful candidate will have a strong background in software design/languages and will be expected to teach a wide variety of courses including: Introduction to Computer Science, Data Structures, Software Design, Theory of Programming Languages, and Compiler Design.

Fairfield University, founded by the Jesuits, is a comprehensive university with about 3,200 undergraduates and a strong emphasis on liberal arts education. The department has an active faculty of 14 full-time tenured or tenure track mem-

bers. We offer a BS in computer science as well as a BS and an MS in mathematics.

Fairfield offers competitive salaries and compensation benefits. The picturesque campus is located on Long Island Sound in southwestern Connecticut, about 50 miles from New York City. Fairfield is an Affirmative Action/Equal Opportunity Employer. For more information see <http://fairfield.edu/mac/index.html>.

Applicants should send a cover letter, a curriculum vitae, teaching and research statements, and three letters of recommendation commenting on the applicant's experience and promise as a teacher and scholar, to Dr. Matt Coleman, Chair of the Department of Mathematics and Computer Science, Fairfield University, 1073 N. Benson Rd., Fairfield CT 06824-5195. Full consideration will be given to complete applications received by January 9, 2009.

Florida State University, Department of Computer Science Tenure-Track Assistant Professor

The Department of Computer Science invites applications for a tenure-track position at the Assistant Professor rank. Research areas related to mobile computing are preferred; strong applicants with expertise in other areas of Computer Science will also be considered. Applicants should hold a PhD in Computer Science or a closely related field, and have excellent research and teaching accomplishments / potential.

The department offers degrees at the BS, MS, and PhD levels. FSU is classified as a Carnegie Research I university. Its primary role is to serve as a center for advanced graduate and professional studies while emphasizing research and providing excellence in undergraduate education. Further information can be found at <http://www.cs.fsu.edu>. FSU is located in the beautiful and picturesque Florida capital - a city of approximately 250,000, about an hour's drive from the Gulf Coast.

Screening will begin December 1, 2008 and will continue until the position is filled. Please use the on-line application form at <http://www.cs.fsu.edu/positions/apply.html>. Questions can be e-mailed to recruitment@cs.fsu.edu. The Florida State University is an Equal Opportunity/Affirmative Action employer, committed to diversity in hiring.

Franklin & Marshall College Professor of Computer Science

Franklin & Marshall College invites applications for a tenure-track Assistant Professor position in COMPUTER SCIENCE, beginning Fall 2009.

The successful candidate will have a Ph.D. in Computer Science (or a related field) with a specialty in informatics. While the initial appointment will be in the Department of Mathematics, the College is committed to forming a new department of Computer Science. In addition, in 2008,

the Howard Hughes Medical Institute awarded the College a substantial grant to launch a bioinformatics program. The successful candidate will— together with colleagues in Computer Science and Biology—design, implement, and sustain new majors in Computer Science and Bioinformatics.

Our teaching load is 3/2 and includes participation in the College's general education requirement, "Foundations," and/or our First-Year Seminar program. A course release is available in each of the first two years for work on program development. Salary will be competitive with computer science salaries at other liberal arts institutions, and significant support is provided for students to collaborate on research.

Teaching experience, evidence of scholarly achievement, and demonstrated interest in collaborating with colleagues from other disciplines are all required. Candidates should submit the following to Barbara Nimershiem, Chair, Computer Science Search Committee, Franklin & Marshall College, P.O. Box 3003, Lancaster PA 17604-3003 USA:

- ▶ a letter of application;
- ▶ a curriculum vitae;
- ▶ teaching and research statements;
- ▶ teaching evaluations (by students and/or supervisors);
- ▶ a graduate transcript; and
- ▶ three letters of recommendation, including at least one that addresses the applicant's teaching ability.

We will not accept application materials electronically. Completed applications received by December 1, 2008, are guaranteed full consider-

ation, although review of applications will continue until the position is filled. Direct any questions to cssearch@fandm.edu or call Barbara Nimershiem at 717-291-3932.

Franklin & Marshall College is a highly selective liberal arts college located in Lancaster, Pennsylvania, about one and one half hours from both Philadelphia and Baltimore. For more information about the College, see our web site at www.fandm.edu. Franklin & Marshall College has a demonstrated commitment to cultural pluralism. EOE

Hong Kong University of Science & Technology Assistant/Associate Professor

The Department of Computer Science and Engineering is expected to have two faculty positions open at Assistant/Associate Professor level for the 2009-2010 academic year. The Department currently has 40 faculty members recruited from major universities and research institutions around the world, about 600 undergraduate students, and about 160 postgraduate students. The medium of instruction is English. More information on the Department can be found at <http://www.cse.ust.hk>.

The Department is looking for faculty candidates with interests in multidisciplinary research in the areas of computational science, such as financial engineering, bioinformatics, and computational modeling and simulation. Additional research areas including embedded systems, programming languages and compiler, and multi-core computing will be considered. Applicants

should have a PhD degree and demonstrated potential in teaching and research.

Initial appointment will be made on contract terms for 3 years which is renewable subject to mutual agreement. Salary is highly competitive and will be commensurate with qualifications and experience. A gratuity will be payable upon satisfactory completion of contract. Fringe benefits including medical/dental benefits, annual leave and housing will be provided where applicable.

Applications should be sent through e-mail including a cover letter, curriculum vitae (including the names and contact information of at least three referees), a research statement and a teaching statement (all in PDF format) to csrecruit@cse.ust.hk. Priority will be given to applications received by 31 January 2009. Applicants will be promptly acknowledged through e-mail upon receiving the electronic application material.

Helsinki Institute for Information Technology (HIIT) Director

Applications for the position of DIRECTOR of the Helsinki Institute for Information Technology HIIT are now welcomed.

HIIT conducts world-class research on future information technology. It is a joint research institution of Helsinki University of Technology TKK and the University of Helsinki (UH).

Please visit www.hiit.fi/jobs to see a full job description and to know how to apply.

The application deadline is 13 October 2008.

Faculty Positions, All Levels

Several faculty positions are available at Cornell's department of Computer Science. Candidates are invited to apply from any area of computer science and at all levels including tenured, tenure track, or lecturer. We are especially interested in programming languages, scientific computing, computational biology, networking, and machine learning. However, we are open to hiring in all other areas as well, including artificial intelligence, databases, game design, graphics, robotics, security, systems, and theory of computation.

To ensure full consideration, applications should be received by January 15, 2009, but will be accepted until all positions are filled.

Applicants should submit a curriculum vita, brief statements of research and teaching interests through the web at <http://www.cis.cornell.edu/apply>, and arrange to have at least three references either uploaded on the Web or sent to: **Faculty Recruiting Committee Chair, Department of Computer Science, 4130 Upson Hall, Cornell University, Ithaca, NY 14853-7501 or freeruit@cs.cornell.edu**

Cornell University, located in Ithaca, New York, is an inclusive, dynamic, and innovative Ivy League university and New York's land-grant institution. Its staff, faculty, and students impart an uncommon sense of larger purpose and contribute creative ideas and best practices to further the university's mission of teaching, research, and outreach.



Cornell University

Cornell University is an Equal Opportunity Employer and encourages applications from women and ethnic minorities

<http://chronicle.com/jobs/profiles/2377.htm>



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Faculty Positions in Computer Science Ecole polytechnique fédérale de Lausanne

The School of Computer and Communication Sciences at EPFL invites applications for faculty positions in computer science. We are primarily seeking candidates **for tenure-track assistant professor positions, but suitably qualified candidates for senior positions will also be considered.**

Successful candidates will develop an independent and creative research program, participate in both undergraduate and graduate teaching, and supervise PhD students.

Candidates from all areas of computer science will be considered, but preference will be given to candidates with interests in **algorithms, bioinformatics, graphics, machine learning, and design methodologies for integrated systems.**

Significant start-up resources and research infrastructure will be available. Internationally competitive salaries and benefits are offered.

To apply, please follow the application procedure at <http://icrecruiting.epfl.ch>. The following documents are requested in PDF format: curriculum vitae, including publication list, brief statements of research and teaching interests, names and addresses (including e-mail) of 3 references for junior positions, and 6 for senior positions. Screening will start on **January 1, 2009**. Further questions can be addressed to :

**Professor Willy Zwaenepoel
Dean**

**School of Computer and
Communication Sciences EPFL
CH-1015 Lausanne, Switzerland
recruiting.ic@epfl.ch**

For additional information on EPFL, please consult:
<http://www.epfl.ch> or <http://ic.epfl.ch>

EPFL is an equal opportunity employer.

Kansas State University
 Department of Computing and
 Information Sciences

The Department of Computing and Information Sciences seeks a dynamic individual for its department head. The successful candidate will be a senior scholar/researcher who possesses a Ph.D. degree in Computer Science or in a closely related field, has excellent management abilities, and possesses effective interpersonal skills. The new head is anticipated to be tenured at full professor; candidates from all areas of computing are invited to apply.

The new head will administer teaching, research, scholarship, outreach, and service activities in a diverse and dynamic department consisting of 17 tenure-track faculty and 3 instructors, who lead research programs in bioinformatics and data mining, embedded and distributed systems, programming languages, security, and software engineering. Information about the department can be found at www.cis.ksu.edu.

To apply, email a cover letter, vita, and contact information for four references to recruiting@cis.ksu.edu or mail hard copy to:

David Schmidt, Chair, Department Head
 Search Committee
 Computing and Information Sciences Dept.
 234 Nichols Hall, Kansas State University
 Manhattan, KS 66506
 (telephone: 785-532-7912)

Electronic submission (pdf, ps, doc, or txt) is preferred. Review of applications commences January 1, 2009 and continues until the position is filled.

KANSAS STATE UNIVERSITY IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER. QUALIFIED WOMEN AND MINORITIES ARE ENCOURAGED TO APPLY. Paid for by Kansas State University.

**Max Planck Institute for Software
 Systems (MPI-SWS)**

Tenure-track faculty openings

Applications are invited for tenure-track and tenured faculty positions in all areas related to the design, analysis and engineering of software systems, including programming languages, formal methods, security, distributed, networked and embedded systems, databases and information systems, and human-computer interaction. A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups. Senior candidates must have demonstrated leadership abilities and recognized international stature.

MPI-SWS, founded in 2005, is part of a network of eighty Max Planck Institutes, Germany's premier basic research facilities. MPIs have an established record of world-class, foundational research in the fields of medicine, biology, chemistry, physics, technology and humanities. Since 1948, MPI researchers have won 17 Nobel prizes. The new MPI-SWS aspires to meet the highest standards of excellence and international recognition with its research in software systems.

To this end, the institute offers a unique environment that combines the best aspects of a uni-



UNIVERSITÄT
 DES
 SAARLANDES



Saarland University is seeking to establish several Junior Research Groups (W1/W2)

within the recently established Cluster of Excellence "Multimodal Computing and Interaction" which was established by the German Research Foundation (DFG) within the framework of the German Excellence Initiative.

The term "multimodal" describes the different types of digital information such as text, speech, images, video, graphics, and high-dimensional data, and the way it is perceived and communicated, particularly through vision, hearing, and human expression. The challenge is now to organize, understand, and search this multimodal information in a robust, efficient and intelligent way, and to create dependable systems that allow natural and intuitive multimodal interaction. We are looking for highly motivated young researchers with a background in the research areas of the cluster, including algorithmic foundations, secure and autonomous networked systems, open science web, information processing in the life sciences, visual computing, large-scale virtual environments, synthetic virtual characters, text and speech processing and multimodal dialog systems. Additional information on the Cluster of Excellence is available on <http://www.mmci.uni-saarland.de>. Group leaders will receive junior faculty status at Saarland University, including the right to supervise Bachelor, Master and PhD students. Positions are limited to five years.

Applicants for W1 positions (phase I of the program) must have completed an outstanding PhD. Upon successful evaluation after two years, W1 group leaders are eligible for promotion to W2. Direct applicants for W2 positions (phase II of the program) must have completed a postdoc stay and must have demonstrated outstanding research potential and the ability to successfully lead their own research group. Junior research groups are equipped with a budget of 80k to 100k Euros per year to cover research personnel and other costs.

Saarland University has leading departments in computer science and computational linguistics, with more than 200 PhD students working on topics related to the cluster (see <http://www.informatik-saarland.de> for additional information). The German Excellence Initiative recently awarded multi-million grants to the Cluster of Excellence "Multimodal Computing and Interaction" as well as to the "Saarbrücken Graduate School of Computer Science". An important factor to this success were the close ties to the Max Planck Institute for Computer Science, the German Research Center for Artificial Intelligence (DFKI), and the Max Planck Institute for Software Systems which are co-located on the same campus.

Candidates should submit their application (curriculum vitae, photograph, list of publications, short research plan, copies of degree certificates, copies of the five most important publications, list of five references) to the coordinator of the cluster, Prof. Hans-Peter Seidel, MPI for Computer Science, Campus E1 4, 66123 Saarbrücken, Germany. Please, also send your application as a single PDF file to applications@mmci.uni-saarland.de.

The review of applications will begin on January 15, 2009, and applicants are strongly encouraged to submit applications by that date; however, applications will continue to be accepted until January 31, 2009. Final decisions will be made following a candidate symposium that will be held during March 9 – 13, 2009.

Saarland University is an equal opportunity employer. In accordance with its policy of increasing the proportion of women in this type of employment, the University actively encourages applications from women. For candidates with equal qualification, preference will be given to people with physical disabilities.

versity department and a research laboratory:

- a) Faculty receive generous base funding to build and lead a team of graduate students and post-docs. They have full academic freedom and publish their research results freely.
- b) Faculty have the opportunity to supervise doctoral theses, teach graduate and undergraduate courses, and have the flexibility to incorporate teaching into their research agenda.
- c) Faculty are provided with outstanding technical and administrative support facilities as well as internationally competitive compensation packages.

Funds have been committed to grow the institute to a strength of 17 tenured and tenure-track faculty, and about 100 doctoral and post-doctoral positions. Additional growth through outside funding is expected. We maintain an open, international and diverse work environment and seek applications from outstanding researchers regardless of national origin or citizenship. The working language is English; knowledge of the German language is not required for a successful career at the institute.

The institute is located in Kaiserslautern and Saarbruecken, in the tri-border area of Germany, France and Luxembourg. The area offers a high standard of living, beautiful surroundings and easy access to major metropolitan areas in the center of Europe, as well as a stimulating, competitive and collaborative work environment. In immediate proximity are the MPI for Informatics, Saarland University, the Technical University of Kaiserslautern, the German Center for Artificial

Intelligence (DFKI), and the Fraunhofer Institutes for Experimental Software Engineering and for Industrial Mathematics.

Qualified candidates should apply online at <http://www.mpi-sws.org/application>.

The review of applications will begin on January 12, 2009, and applicants are strongly encouraged to apply by that date; however, applications will continue to be accepted until February 27, 2009.

The Max Planck Society is committed to increasing the representation of minorities, women and individuals with physical disabilities in Computer Science. We particularly encourage such individuals to apply.

Mississippi State University Faculty Positions in Computer Science or Software Engineering

The Department of Computer Science and Engineering (<http://www.cse.msstate.edu>) is seeking to fill an open position for a tenure-track faculty member at the Assistant/Associate Professor levels. Evidence of strong potential for excellence in research (including the ability to attract external funding) and teaching at the graduate and undergraduate levels is required. The primary area of interest for this position is Software Engineering, but applicants in the areas of artificial intelligence, scientific computing, networking and high performance computing, computer security, and graphics and visualization will also be considered. Mississippi State University is the

largest university in the State of Mississippi with approximately 1000 faculty and 18,000 students. The Department of Computer Science and Engineering has 18 tenure-track faculty positions and offers academic programs leading to the bachelor's, master's and doctoral degrees in computer science and bachelor's degrees in software engineering and computer engineering. The Department also cooperates with the Department of Electrical and Computer Engineering in offering master's and doctoral degrees in computer engineering, and participates in the interdisciplinary program in computational engineering leading to master's and doctoral degrees. Faculty members and graduate students work with a number of on-campus research centers including the Center for Computer Security Research, the High Performance Computing Collaboratory, the Institute for Neurocognitive Science and Technology, the Institute for Digital Biology, the Center for Advanced Vehicular Systems and the GeoResources Institute. Seven faculty members in the department have been recognized by NSF CAREER awards, and faculty members within the department currently hold grants from the NSF, NASA, Department of Defense, Department of Justice, industry, as well as others. Department research expenditures total around three million dollars per year. Mississippi State is ranked in the top 100 among public universities in the nation by the National Science Foundation for total institutional research expenditures.

Applicants should submit a letter of application, curriculum vita, teaching statement, research statement, and names and contact information of at least three references online at <http://www.jobs.msstate.edu/>. Review of applications will begin not earlier than November 2008 and continue until the position is filled. MSU is an Affirmative Action/Equal Opportunity Employer.

New Jersey Institute of Technology Assistant Professor/Software, Software Engineer

The Computer Science Dept. at New Jersey Institute of Technology (NJIT) seeks to hire faculty for a tenure-track position beginning Fall 2009. Applications are invited from candidates with research & teaching interests in multiple aspects of software, such as Software Engineering & Web Technologies & Services. Experience with practical software building and/or Open Source projects a plus.

Applicant should have a PhD (or expect to receive one by summer 2009) in computer science. Applicant should have demonstrated potential for original research, a commitment to excellence in teaching & familiarity with practical aspects of software. Salary is competitive & commensurate with appointment rank & qualifications.

NJIT is a public research university. The dept. offers programs at the undergraduate, masters & PhD levels in Computer Science. The dept. also offers undergraduate & graduate degree programs in Bioinformatics. NJIT is located in Newark's University Heights, a multi-institutional campus shared with Rutgers University at Newark, the University of Medicine & Dentistry of New Jersey & Science Park. NJIT's location in the NY metro area is ideal for research collaboration. The area is home to other universities & research labora-



Tenure-Track Full Professor in Embedded Security

Department of Electrical and Computer Engineering

The Department of Electrical and Computer Engineering at the University of Massachusetts Amherst invites applications from candidates for a tenure-track position in Embedded Security at the Full Professor level starting September 2009. We seek to aggressively grow our expertise in Embedded Security which spans Communications, Networking, Microelectronics, Cryptography and Embedded Systems Engineering. In collaboration with various government and industry organizations as well as faculty in Computer Science and Transportation Engineering, we are currently establishing a new research center in Embedded Security. Information about the ECE Department can be found at <http://www.ecs.umass.edu/ece>

Candidates should have a well-established track record of research excellence in the areas of interest for this search, which include but are not limited to: applied cryptography, RFID, embedded systems, cryptanalysis, and secure system prototyping. In addition to a demonstrated commitment to teaching at both the undergraduate and graduate levels, candidates should have an understanding of diversity issues and their educational importance, a strong publication record, demonstrated leadership and vision in their field, and a demonstrated ability to attract significant external research support.

Rank and salary will be commensurate with qualifications and experience. The committee will begin reviewing applications on November 1, 2008. The search will continue until the position is filled. Interested applicants should send a CV, statement of research and teaching interests, and the names and addresses of at least four references to: embeddedsecuritysearch@ecs.umass.edu, or printed copy to: Search Committee, Embedded Security, Electrical and Computer Engineering, University of Massachusetts, Knowles Engineering Building, 151 Holdsworth Way, Amherst, MA 01003.

The University, College, and Department are committed to increasing diversity of the faculty, student body, and curriculum, and welcome applications from women and other underrepresented groups. UMass is an Affirmative Action/Equal Opportunity Employer.

tories, as well as major financial, telecommunications & pharmaceutical companies, offering excellent opportunities for collaboration, consulting & industry sponsored research.

New Jersey enjoys a high standard of living & quality of life. Newark is minutes from New York City & close to the Jersey Shore, providing a wide range of cultural & leisure activities. To apply, visit njit.jobs & must use posting #0600306. Please include the following:

- ▶ CV
- ▶ research statement
- ▶ teaching statement cover letter

Please also ask at least three references to send letters of recommendation to faculty-search@cs.njit.edu. For more info. about the Computer Science Dept., visit: cs.njit.edu.

NJIT is an equal opportunity, affirmative action, equal access employer & especially encourages applications from minorities, women & persons with disabilities.

New Mexico State University Assistant Professor

The Computer Science Department at New Mexico State University invites applications for a tenure-track position at the assistant professor level, with appointment starting in the Fall semester 2009. We are particularly interested in candidates with expertise in computer networks and related areas. Applications from women and members of traditionally under-represented groups are

particularly encouraged. Salary and start up package will be competitive and commensurate with qualifications and experience.

The minimum qualifications are a Ph.D. degree in Computer Science or in a closely-related discipline by the time of appointment, along with demonstrated evidence of excellence in teaching and research. We particularly solicit applications from candidates with experience in inter-disciplinary research activities and candidates whose research foci complement and integrate the existing research activities in the Department, in areas like knowledge representation and software engineering. The successful candidate will be expected to develop an independent research program and teach graduate and undergraduate courses in Computer Science.

The Department has strong research and educational programs, extensive computing infrastructure (which includes several parallel and distributed platforms), and various computing and research laboratories. The Department offers B.S., M.S., and Ph.D. degrees in Computer Science; it actively participates in several interdisciplinary educational programs. The Department has received extensive funding for support of its instructional and research activities from a broad spectrum of agencies, including a \$4.5M grant from NSF to establish a Center for Research Excellence in Bioinformatics.

NMSU is located in southern New Mexico, the "Land of Enchantment", just 50 miles from the El Paso airport. NMSU is a land grant institution, with strong research programs and a tradition in serving a diverse student population (NMSU is a

Minority-serving Institution). The NMSU campus houses the Physical Science Laboratory and has close ties to Sandia and Los Alamos National Laboratories, the White Sands Missile Range, and the National Center for Genomics Research. For more information, please visit <http://www.cs.nmsu.edu/>.

Applicants should submit a letter of intent, complete curriculum vitae, a research and teaching statement, and at least three letters of reference to:


Dr. Son Cao Tran, CS Faculty Search Chair
Department of Computer Science,
New Mexico State University
P.O. Box 30001, MSC CS
Las Cruces, NM 88003, USA

Enquiries by email (faculty_search@cs.nmsu.edu) are welcome. Screening will begin November 17th, 2008. Applications will be accepted until the position is filled.

New Mexico State University is an EEO/AA Employer. Offer of employment is contingent upon verification of individual's eligibility for employment in the United States. All university positions are contingent upon availability of funding.

New York University Faculty Openings

The department expects to have several regular faculty positions beginning in September 2009 and invites candidates at all levels. We will consider outstanding candidates in any area of com-



Chairperson, Department of Electrical Engineering and Computer Science

The Department of Electrical Engineering and Computer Science (EECS) of the University of Kansas seeks an outstanding individual for the position of chairperson. The University of Kansas has approximately 26,000 students at the main campus in Lawrence, a community consistently rated as one of the most desirable places to live. EECS is the largest department in the School of Engineering, with 35 faculty members and a research volume of approximately \$7.5 million. EECS offers undergraduate and M.S. degrees in Electrical Engineering, Computer Engineering, and Computer Science, a M.S. in Information Technology at KU's Edwards Campus (metro Kansas City), and Ph.D. degrees in Electrical Engineering and Computer Science. The department has approximately 425 undergraduate and 225 graduate students. See www.eecs.ku.edu. The successful candidate should have an earned doctorate or equivalent in electrical engineering, computer engineering, computer science, or related field and have an interest in leading faculty, developing academic and research programs, and representing the department to industry, government, administration, and alumni. The appointment will be effective as negotiated. To apply (or nominate someone), visit our website at www.eecs.ku.edu/recruitment. Applications will be reviewed beginning November 15, 2008 and will be accepted until the position is filled.

EO/AA employer.



Faculty Positions

with the School Of Computer Engineering, Nanyang Technological University

A member of the College of Engineering, the School of Computer Engineering (SCE) originated from the School of Applied Science that was established in 1988. Recognizing the rapid growth in the information technology arena, SCE was formed in 2000. It offers undergraduate training leading to BEng (Hons) in Computer Engineering or Computer Science as well as full-time and part-time graduate training leading to MSc and PhD. SCE's strengths lie in constantly maintaining industrial relevance in its training of undergraduate and graduate students, as well as pioneering innovative cutting-edge research. Further information about the school can be obtained at <http://www.ntu.edu.sg/sce>.

Applications are invited for appointment as **Associate Professor** and **Assistant Professor** in the School of Computer Engineering at Nanyang Technological University, Singapore. Successful applicants will have the opportunity to work in a creative and intellectually stimulating environment within a young, vibrant university.

For Associate Professor and Assistant Professor positions, preference will be given to candidates with expertise in one or more of the following areas:

- **Human Computer Interaction**
- **High Performance Computing**
- **Agents, Service Computing and Text Mining**

Candidates for appointment at an Associate Professor level must possess an outstanding track record of research through publication in top rank journals, obtaining grants and academic leadership, as well as a willingness and demonstrated ability to teach at the undergraduate and graduate levels. Candidates for appointment at Assistant Professor level must demonstrate strong research potential and a willingness and ability to teach at the undergraduate and graduate levels.

Candidates applying for Associate Professor and Assistant Professor positions are expected to carry out research in one of the research centres hosted by the School and teach a broad range of undergraduate subjects in the Computer Engineering / Computer Science areas.

The University offers highly competitive salaries and start-up funding, and faculty have access to significant research grants provided by the various funding agents in Singapore. Informal enquiries about the post can be made to A/P Yeo Chai Kiat, Associate Chair (Academic), on +65 6790 4929 or email VD-SCE-ACAD@ntu.edu.sg.

Submission of application forms can be made to VD-SCE-ACAD@ntu.edu.sg. Guidelines for application submission and application forms can be obtained from <http://www.ntu.edu.sg/ohr/Career/SubmitApplications/Pages/Faculty.aspx>

Closing Date: **15 Dec 2008**

www.ntu.edu.sg

puter science with systems and machine learning being high-priority areas.

Faculty members are expected to be outstanding scholars and to participate in teaching at all levels from undergraduate to doctoral.

New appointees will be offered competitive salaries and startup packages, with affordable housing within a short walking distance of the department. New York University is located in Greenwich Village, one of the most attractive residential areas of Manhattan.

The department has 34 regular faculty members and several clinical, research, adjunct, and visiting faculty members. The department's current research interests include algorithms and theory, computational biology, computer vision, cryptography, distributed and parallel computing, graphics and multimedia, machine learning, natural language processing, networking, scientific computing, and verification and programming languages.

Collaborative research with industry is facilitated by geographic proximity to computer science activities at AT&T, Google, IBM, Bell Labs, NEC, Siemens and Telcordia.

Please apply online at http://webern.cs.nyu.edu/faculty_applications/

To guarantee full consideration, applications should be submitted no later than Jan. 2, 2009; however, this is not a hard deadline, as all candidates will be considered to the extent feasible, until all positions are filled. Visiting positions may also be available.

New York University is an equal opportunity/affirmative action employer.

North Carolina State University Department of Computer Science Faculty Positions

The Department of Computer Science at NC State University (NC State) seeks to fill multiple tenure track faculty positions starting August 16, 2009. Successful candidates must have a strong commitment to academic and research excellence, and an outstanding research record commensurate with the expectations of a major research university. Required credentials include a doctorate in Computer Science or a related field. While the department expects to hire faculty primarily at the Assistant Professor level, candidates with exceptional research records are encouraged to apply for senior positions.

Exceptional candidates in all areas of Computer Science will be considered, but of particular interest are candidates specializing in Computer Games and in Software Engineering. New Games faculty will play an active role in the Digital Games Research Center. New Software Engineering faculty will play an active role in the Center for Open Software Engineering and in the Secure Open Systems Initiative.

The Department is one of the largest and oldest in the country. It is placed in the NCSU's College of Engineering, which has recently received significant increases in private and public funding, faculty positions, and facilities that will assist the Department in attaining its goals. The department's research expenditures and recognitions are growing steadily. For example, we have one of the largest concentrations of the prestigious NSF

Early Career Award winners (18 total).

NCSU is located in Raleigh, capital of North Carolina, which forms one vertex of the world-famous Research Triangle Park (RTP). RTP is an innovative environment, both as a metropolitan area with one of the most diverse industrial bases in the world, and as a center of excellence promoting technology and science. The Research Triangle area is routinely recognized in nationwide surveys as one of the best places to live in the U.S. We enjoy outstanding public schools, affordable housing, and great weather, all in the proximity of the mountains and the seashore.

Applications will be reviewed as they are received. The positions will remain open until suitable candidates are identified. Applicants will receive consideration starting December 15, 2008. Applicants should submit the following online at <http://jobs.ncsu.edu> (reference position number 04-69-0808): cover letter, curriculum vitae, research statement, teaching statement, and names and complete contact information of four references, including email addresses and phone numbers. Candidates can obtain information about the department and its research programs, as well as more detail about the positions advertised here at <http://www.csc.ncsu.edu/jobs>. Inquiries may be sent via email to: facultyhire@csc.ncsu.edu.

North Carolina State University is an equal opportunity and affirmative action employer. In addition, NC State University welcomes all persons without regard to sexual orientation. Individuals with disabilities desiring accommodations in the application process should contact the Department of Computer Science at (919) 515-2858.



Windows Kernel Source and Curriculum Materials for Academic Teaching and Research.

The Windows® Academic Program from Microsoft® provides the materials you need to integrate Windows kernel technology into the teaching and research of operating systems.

The program includes:

- **Windows Research Kernel (WRK):** Sources to build and experiment with a fully-functional version of the Windows kernel for x86 and x64 platforms, as well as the original design documents for Windows NT.
- **Curriculum Resource Kit (CRK):** PowerPoint® slides presenting the details of the design and implementation of the Windows kernel, following the ACM/IEEE-CS OS Body of Knowledge, and including labs, exercises, quiz questions, and links to the relevant sources.
- **ProjectOZ:** An OS project environment based on the SPACE kernel-less OS project at UC Santa Barbara, allowing students to develop OS kernel projects in user-mode.

These materials are available at no cost, but only for non-commercial use by universities.

For more information, visit www.microsoft.com/WindowsAcademic or e-mail compisci@microsoft.com.

Northern Arizona University Sustainable Systems/ Advanced Engineering Design

The College of Engineering, Forestry, and Natural Sciences at Northern Arizona University in Flagstaff, Arizona invites applications for one (1) tenure-track faculty position in the areas of Sustainable Systems and/or Advanced Engineering Design. This is a new position created to support our recently implemented Master of Science Engineering (MSE) program. The successful candidate will start his or her duties at the beginning of the fall 2009 semester in August. Minimum qualifications are an earned doctorate in civil engineering, computer science, electrical engineering, environmental engineering, or mechanical engineering; or a closely allied engineering or computational science field. Preferred qualifications include experience or demonstrable interests in: teaching courses related to design processes, interdisciplinary approaches, and sustainable systems engineering; a research agenda compatible with the MSE themes that supports the research and funding needs of MSE students; supporting one or more of our existing engineering and computer science undergraduate programs; collaborating across the disciplines of the College; and working with diverse populations. Previous experience in industry, consulting, non-governmental organizations, or the public sector is also desirable. Please see <http://www.nau.edu/hr> for full position announcement. NAU is an AA/EEO/MWDV Employer.

Ohio Northern University Assistant Professor

The Electrical & Computer Engineering and Computer Science (ECCS) Department at Ohio Northern University seeks applicants for a tenure-track faculty position at the rank of Assistant Professor to start September 1, 2009. Appointee must have a keen interest in undergraduate education with a strong commitment to teaching. Ph.D. in computer science, computer engineering, or related field by date of employment is required. Applicants in the final stages of a Ph.D. program may be considered for appointment as an Instructor. Candidates with expertise in the area of microcontrollers and/or teaching experience in introductory computer programming will be given preference. The department offers B.S. degrees in computer engineering, computer science, and electrical engineering. The 160+ students in the three programs are well prepared and highly motivated. The facilities are well equipped and maintained by a full-time technician. Our nine faculty members are enthusiastic and dedicated to undergraduate teaching.

Founded in 1871, Ohio Northern University, located in West Central Ohio, is a private university offering a diverse, dynamic and unique learning community, with rigorous professional programs in partnership with the arts and the sciences. Its 3,600 students study for graduate and undergraduate degrees in five colleges: Arts and Sciences, Engineering, Business Administration, Pharmacy, and Law. Ohio Northern takes pride in being a student-centered, service-oriented, val-

ues-based institution. Further information about the University is available at <http://www.onu.edu>.

Applicants should submit a letter of application, vita, transcripts, and three letters of recommendation to: Dr. John K. Estell, Chair, Electrical & Computer Engineering and Computer Science Department, Ohio Northern University, 525 S. Main St., Ada, Ohio 45810. Ohio Northern University is an Affirmative Action / Equal Opportunity Employer, and women and minority candidates are encouraged to apply. The search will continue until the position is filled.

Polytechnic Institute of New York University Postdoctoral Position in Computer Engineering

The Department of Electrical and Computer Engineering at the Polytechnic Institute of New York University is seeking applicants for a post doctorate position in the area of parallel computing. Among the position requirements are a PhD in computer science, electrical engineering or computer engineering and a background in multi-processor architectures, embedded systems, microarchitectural design and operating systems.. The successful candidate will participate in the research and development and integration of a high performance computing system being developed within our department. The position is for two years starting immediately. To apply send an email with your resume and cover letter to ecesearch@poly.edu.

Rice University Department of Computer Science Faculty and/or Research Positions

The Department of Computer Science at Rice University seeks applications for several tenure-track faculty appointments to start in July 2009. We welcome outstanding candidates in all areas of computer science. We are particularly looking for candidates with interests and experience in any aspect of one of the following two areas:

Analyzing, designing, and verifying complex systems. These systems may be biological, mechanical or information-based. Examples of complex systems include but are not limited to swarm-based robotics, multi-agent artificial intelligence, social networks and synthetic biology. This search is part of a larger School of Engineering search in complex systems involving multiple positions in multiple departments.

Biomedical informatics, including but not limited to work in bio-informatics, computational biology, imaging informatics, clinical informatics and public health informatics. This search is in anticipation of furthering Rice University's collaboration with the Baylor College of Medicine and Texas Children's Hospital. These positions are designed to facilitate joint research between Rice faculty and members of the Texas Medical Center.

We anticipate hiring at all ranks including Assistant Professor, Associate Professor and Full Professor. The Department and its associated research groups also have openings for research positions, including research faculty, research scientists, and postdoctoral researchers. The availability of re-



Boston University Department of Computer Science Assistant Professor

Applications are invited for two tenure-track assistant professorships beginning September 2009 (pending approval). Qualifications required of all applicants include a Ph.D. in Computer Science or related discipline, a strong research record, and a commitment to teaching. All research areas of Computer Science will be considered. Particular attention will be given to candidates pursuing research in Computing Systems, Trustworthy Computing, and/or Informatics (e.g., databases, data mining, machine learning).

Currently, the Department comprises 17 faculty members, and offers programs leading to B.A., M.A., and Ph.D. degrees. In recent years the Department has expanded in research strength with current research interests including databases, fault-tolerant computing, image and video computing, network protocols and services, operating systems, performance evaluation, programming languages, real-time systems, security, and theory of computation and algorithms. In addition, the Department maintains a close association with other university groups working in various applied computing areas including scientific computing, computer engineering, and bioinformatics.

The Department maintains a state-of-the-art computing environment and has full access to the university's supercomputing facilities, high-speed campus networks, and Internet2. Recently, the Department has received significant government and industry grants for research, research infrastructure, and for graduate student support. We anticipate that this period of growth will continue based on our recent successes and the continued strong support of the University.

Additional information on the Department is available from <http://www.cs.bu.edu>.

Qualified applicants should apply by filling out the application form available at <http://www2.cs.bu.edu/faculty-app/>, or by sending their resume, a cover letter stating one's areas of specialization, and at least three letters of recommendation to:

Faculty Search Committee
Computer Science Department
111 Cummington Street
Boston University
Boston, MA 02215

Boston University is an Equal Opportunity/Affirmative Action employer.



Northeastern University Faculty Opening in Computer Engineering



The Computer Engineering group of the ECE department at Northeastern University in Boston, MA, invites applications from candidates for several open positions. The CE group at NEU is a dynamic group of 11 faculty with established strengths in several areas including computer architecture, computer networks, VLSI design and testing, reliable computing, computer vision, machine learning, embedded systems and information assurance.

We anticipate hiring strong candidates at the Assistant Professor level, although exceptional applicants at other ranks will be considered. We have a particular interest in attracting candidates with expertise in computer security, computer networking, embedded systems, parallel/multi-core computer architecture, nanoscale VLSI design (scaled CMOS and beyond), robotics, and software engineering.

Computer Engineering is just one area of strength in the ECE Department at Northeastern University (<http://www.ece.neu.edu>), which has over 40 faculty members. The ECE department has established areas of excellence in sensing and imaging (NSF Center), nanomanufacturing (NSF Center), communications and digital signal processing, power and control systems, power electronics, microwave magnetic materials and device technology.

A Ph.D. in Computer Engineering, Computer Science or a closely related field is required. Successful candidates will be expected to develop strong independent research programs and to excel in teaching in both our undergraduate and graduate programs.

To apply, please submit complete curriculum vitae, research and teaching statements, and names and contact information for at least four references. All materials should be clearly marked with the applicant's name. Screening for the applications will start on January 1, 2009 and will continue until the position is filled. Materials should be sent electronically (all in PDF) to cefachire@ece.neu.edu.

Northeastern University is an Equal Opportunity/Affirmative Action, Title IX, educational institution and employer. For additional information visit <http://www.neu.edu>.



Northeastern
UNIVERSITY

<http://www.neu.edu>

search positions is contingent on external funding.

Applicants for both tenure-track faculty and research positions should hold a Ph.D. degree or equivalent in computer science or a related discipline, or expect to complete such requirements prior to assuming an appointment. A commitment to excellence in both research and teaching is required for a tenure-track appointment. Early applications will be appreciated.

The Department has access to superb research facilities, including parallel and multiprocessor systems laboratories, three terascale computers, large networks of workstations, and a high-speed network test bed. The university is located across the street from the Texas Medical Center, one of the premiere centers for medical research in the country. Houston's oil, medical, aerospace, and technology communities all combine to make it a center for many kinds of computation, from high-performance computing through real-time and embedded systems.

Rice University is a private university with a strong reputation for academic excellence in both undergraduate education and in research. Rice attracts outstanding undergraduate and graduate students from across the nation and around the world. Rice provides a stimulating environment for research, teaching, and joint projects with industry. Teaching loads are low to accommodate faculty research.

Please submit a resume, a statement of research and teaching interests, and the names and addresses of at least three references through the Computer Science website <http://csfacultyapplications.rice.edu>

The deadline for applications is January 15,

2009, but earlier submissions are appreciated. Please specify whether you are applying for a tenure-track faculty position or a research position. More information can be found on our web site, <http://www.cs.rice.edu> or by contacting Ms. Darnell Price at 713-348-5200 or by email at darnell@rice.edu.

Rice University is an Equal Opportunity/Affirmative Action Employer.

Santa Clara University Department of Operations & Management Information Systems

The Department of Operations and Management Information Systems of the Leavey School of Business is seeking a qualified candidate for Fall 2009 tenure-track appointment in Management Information Systems.

The department offers a graduate degree in IS (MSIS) and undergraduate major and minor degrees in MIS. In keeping with the University's teacher-scholar model, the department seeks candidates who are highly committed to teaching and actively engaged in research in their fields of specialization in areas of Information Systems. We are seeking individuals with a Ph.D. in MIS, or a related field, and a specialization in MIS, Large Enterprise Systems (ERP, CRM), or Information Systems for Financial and Managerial Reports. A record of high quality scholarship and superior teaching is required.

Santa Clara University is an equal opportunity/affirmative action employer and welcomes applications from women, persons of color, and

members of historically under-represented U.S. ethnic groups.

Application Deadline: Open until position is filled.

Please send a letter of application, vita, three letters of reference and teaching evaluations to:

Chair, MIS Search Committee
Operations and MIS Department
Leavey School of Business
Santa Clara University
Santa Clara, CA 95053-0382
or email to: eturner@scu.edu

Seattle University Tenure-track faculty position

The Department of Computer Science and Software Engineering invites applications for a tenure-track faculty position at the assistant professor rank to begin in September of 2009. Applicants are required to have a Ph.D. in Computer Science, Software Engineering, or a closely allied field, and should be capable of teaching a broad range of computer science and/or software engineering courses. Seattle University offers Bachelor of Arts and Bachelor of Science degrees in Computer Science and a Master of Software Engineering degree (the country's first MSE degree, now in its 30th year).

Applications must include a curriculum vita, statements of teaching philosophy and research plans, and a separate statement addressing how you could contribute to the Seattle University mission. We also require three letters of reference sent directly to the Department. Please send your application to Dr. Richard LeBlanc, Faculty Search Committee Chair, Department of Computer Science and Software Engineering, Seattle University, 901 12th Avenue P.O. Box 222000, Seattle, WA 98122-1090. Information about Seattle University, a statement of its mission, and job announcement can be found at www.seattleu.edu/home/about_seattle_university/ and www.seattleu.edu/scieng/comsci/. Applications preferred by December 1st and will close on December 31, 2008.

Seattle University, founded in 1891, continues a more than four hundred and fifty year tradition of Jesuit Catholic higher education. The University's Jesuit Catholic ideals underscore its commitment to the centrality of teaching, learning and scholarship, of values-based education grounded in the Jesuit and Catholic traditions, of service and social justice, of lifelong learning, and of educating the whole person. Located in the heart of dynamic Seattle, the University enrolls approximately 7,700 undergraduate and graduate students in eight colleges and schools. Students enjoy a university ethos characterized by small classes, individualized faculty attention, a strong sense of community, a commitment to diversity, and an outstanding faculty.

Seattle University is an equal opportunity employer.

St. Lawrence University Department of Mathematics, Computer Science and Statistics Assistant Professor

St. Lawrence University invites applications for a tenure-track position in computer science in the



香港大學
THE UNIVERSITY OF HONG KONG

Founded in 1911, The University of Hong Kong is committed to the highest international standards of excellence in teaching and research, and has been at the international forefront of academic scholarship for many years. Of a number of recent indicators of the University's performance, one is its ranking at 18 among the top 200 universities in the world by the UK's *Times Higher Education Supplement*. The University has a comprehensive range of study programmes and research disciplines, with 20,000 undergraduate and postgraduate students from 50 countries, and a complement of 1,200 academic members of staff, many of whom are internationally renowned.

Associate Professors/Assistant Professors in the Department of Computer Science (Ref.: RF-2008/2009-207)

Applications are invited for appointments as Associate Professor/Assistant Professor (2 posts) in the Department of Computer Science, tenable from as soon as possible on a three-year fixed-term basis, with consideration for tenure during the second three-year contract.

The Department offers programmes at both undergraduate and postgraduate levels, and has excellent computing resources, well-equipped teaching and research facilities and support. Information about the Department can be obtained at <http://www.cs.hku.hk/>.

Applicants should have a Ph.D. degree in Computer Science, Computer Engineering, or related fields, and a strong interest in research and teaching. A solid track record in research is essential. Applicants should be active and committed to research in bioinformatics, theoretical computer science, data engineering, computer graphics and vision, and systems or computer security. Those in other research areas will be considered exceptionally.

Applicants should indicate clearly which level (preferably with reference number) they wish to be considered for. Successful applicants with higher qualifications and experience will be appointed at a higher level. Those who have responded to the previous exercise (Ref.: RF-2007/2008-21) need not re-apply, as they will be reconsidered together with the new applicants.

Annual salaries will be in the following ranges (subject to review from time to time at the entire discretion of the University):

Associate Professor : HK\$622,740 – 963,060
Assistant Professor : HK\$474,600 – 733,440 (approximately US\$1 = HK\$7.8)

The appointments will attract a contract-end gratuity and University contribution to a retirement benefits scheme, totalling up to 15% of basic salary. At current rates, salaries tax does not exceed 15% of gross income. The appointments carry leave, and medical/dental benefits. Housing benefits will be provided as applicable.

Further particulars and application forms (152/708) can be obtained at <https://www.hku.hk/apptunit/>; or from the Appointments Unit (Senior), Human Resource Section, Registry, The University of Hong Kong, Hong Kong (fax: (852) 2540 6735 or 2559 2058; e-mail: senrapp@hku.hk). **Applications will be accepted until the positions are filled. Candidates who are not contacted within 6 months may consider their applications unsuccessful.**

The University is an equal opportunity employer and is committed to a No-Smoking Policy

Department of Mathematics, Computer Science and Statistics at the assistant professor level to begin in August 2009. Qualifications include a Ph.D. in computer science, a strong commitment to teaching undergraduates in a liberal arts setting, evidence of excellence in teaching, and evidence of research potential. The teaching load is three courses per semester with a reduced load possible for supervising senior research projects. Please visit the University's webpage at <http://www.stlawu.edu/> for more information about our students and programs.

Review of applications will begin on November 15, 2008, and will continue until the position is filled. Candidates should send a letter of application, a CV, a statement of teaching philosophy, and arrange for three letters of recommendation to be sent to: Ed Harcourt, Computer Science Search Committee Chair, Department of Mathematics, Computer Science, and Statistics, St. Lawrence University, Canton, NY 13617.

St. Lawrence University is an Affirmative Action/Equal Employment Opportunity employer. Women, minorities, veterans, and persons with disabilities are encouraged to apply.

Stanford University Department of Computer Science Faculty Opening

The Department of Computer Science at Stanford University invites applications for a tenure-track faculty position at the junior level (Assistant or untenured Associate Professor). We give high priority to the overall originality and promise of the candidate's work rather than the candidate's sub-area of specialization within Computer Science.

We are seeking applicants from all areas of Computer Science, including Foundations, Systems, Artificial Intelligence, Graphics, Computer Vision and Perception, Databases, and Human-Computer Interaction. We are also interested in applicants doing research at the frontiers of Computer Science with other disciplines, such as Biology, Neuroscience, Economics, Education, and Art, with potential connections to Stanford's main multidisciplinary initiatives: Human Health, Environment and Sustainability, the Arts and Creativity, and the International Initiative.

An earned Ph.D., evidence of the ability to pursue a program of research, and a strong commitment to graduate and undergraduate teaching are required. A successful candidate will be expected to teach courses at the graduate and undergraduate levels and to build and lead a team of graduate students in Ph.D. research. Further information about the Computer Science Department can be found at <http://cs.stanford.edu/>. The School of Engineering website may be found at <http://soe.stanford.edu/>.

Applications should include a curriculum vita, brief statements of research and teaching interests, and the names of at least four references. Candidates are requested to ask references to send their letters directly to the search committee. Applications and letters should be sent to: Search Committee Chair, c/o Laura Kenny-Carlson, via electronic mail to search@cs.stanford.edu.

The review of applications will begin on January 2, 2009, and applicants are strongly encouraged to submit applications by that date; however, applications will continue to be accepted until February 1, 2009 or until the position is filled.

Stanford University is an equal opportunity employer and is committed to increasing the diversity of its faculty. It welcomes nominations of and applications from women and members of minority groups, as well as others who would bring additional dimensions to the university's research and teaching missions.

University of California, Irvine Endowed, Distinguished Faculty Positions Donald Bren School of Information & Computer Sciences

The Donald Bren School of Information and Computer Sciences was endowed with a transformational gift that included ten "Bren Chairs" to be filled with scholars who are internationally-recognized as leaders in emerging issues of any area of information and computer sciences, including cross-disciplinary research integrating information and computer sciences with other disciplines. The Bren Professors are some of the most distinguished appointments at UC Irvine.

Candidates should bring an integrative outlook to the discipline, enthusiasm in engaging with professional and business communities and the general public, collaborating with UCI scholars who study issues of information and computing technology, and support for the development of innovative technologies and applications. We envision Bren chair-holders to serve as catalysts on campus to establish educational and research programs that foster an interdisciplinary perspective. Accordingly, candidates should not only have a strong disciplinary background with a distinguished record of scholarly publications and extramural funding, but also a proven track record of innovation, collaboration, stimulation and leadership in both education and research.

Appointments will be in the Department of Computer Science, Informatics, or Statistics at the rank of senior, distinguished professor. Scholars doing truly cross-disciplinary research may be jointly appointed with another school at UC Irvine. Four chairs are currently filled.

Candidate screening will begin immediately upon receipt of materials. Applications or nominations should include a cover letter indicating the area of primary research, a CV, up to five recent publications, and identification of five or more references. Electronic submission is preferred; please refer to the following web site for instructions:

http://www.ics.uci.edu/employment/employ_faculty.php

Paper applications should be sent to:

ATTN: Faculty Recruiting – Bren Chair
Donald Bren School of Information and
Computer Sciences
University of California, Irvine
Irvine, CA 92697-3425

The Bren School of ICS has excellent faculty, innovative programs, high quality students and outstanding graduates as well as strong relationships with local and national high tech industry. As one of eleven academic units at UC Irvine, an independent school with three departments— Computer Science, Informatics, and Statistics— the Bren School has a unique perspective that provides a broad foundation from which to build initiatives that explore the full extent of the computing and

information disciplines. With 71 regular-rank faculty members, seven full-time lecturers, approximately 275 doctoral, 130 masters, and 1000 undergraduate students, ICS is one of the largest computing programs in the country. Many faculty in the school engage in interdisciplinary research through various organizations such as the California Institute for Telecommunications and Information Technology (Calit2), the Institute for Genomics and Bioinformatics (IGB), ACE (Arts Computation Engineering), to name but a few. The Bren School of ICS just dedicated a contemporary high-tech building designed to enhance collaborative research and education, and continues to grow. Outstanding candidates in all relevant areas and at other ranks are encouraged to contact us.

UC Irvine (<http://www.uci.edu>) is targeted as a growth campus for the University of California. It is one of the youngest UC campuses, yet consistently ranks among the nation's best public universities. UCI is located three miles from the ocean in southern California with an excellent year-round Mediterranean climate. The area surrounding campus offers numerous outdoor and fine arts opportunities and the public school system in Irvine is ranked one of the highest in the nation.

UCI is an equal opportunity employer committed to excellence through diversity and strongly encourages applications from all qualified candidates, including women and minorities. UCI is responsive to the needs of dual career couples, is dedicated to work-life balance through an array of family-friendly policies, and is the recipient of a National Science Foundation ADVANCE award for gender equity.



ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to jonathan.just@acm.org. Please include text, and indicate the issue or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will be typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Rates: \$295.00 for six lines of text, 40 characters per line. \$80.00 for each additional three lines. The MINIMUM is six lines.

Deadlines: Five weeks prior to the publication date of the issue (which is the first of every month). Latest deadlines: <http://www.acm.org/publications>

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at: <http://campus.acm.org/careercenter> Ads are listed for a period of six weeks.

For More Information Contact:

JONATHAN JUST
Director of Media Sales
at 212-626-0687 or
jonathan.just@acm.org

University of California, Santa Barbara
Tenure-Track Assistant Professor
Media Arts and Technology Graduate Program

The Media Arts and Technology Program at the University of California, Santa Barbara, invites applications for a tenure-track position at the assistant professor level, starting July 1, 2009. The department seeks candidates who will establish a vigorous research and teaching program in computer graphics, scientific/information visualization, or a related field applicable to immersive, interactive, and distributed environments, working with high-dimensional data generated in scientific and artistic domains. The successful candidate will be expected to collaborate with artists, engineers, and scientists in an interdisciplinary environment of research, creative work, and teaching.

Media Arts and Technology (MAT) is a transdisciplinary graduate program at UCSB in both the College of Letters and Science (Division of Humanities and Fine Arts) and the College of Engineering. MAT offers Master's and PhD degrees and has approximately 40 graduate students and 10 faculty, several with joint appointments in engineering and arts departments. Areas of expertise include human-computer interaction, electronic music and sound design, computational visual and spatial arts, and multimedia signal processing. Offices and labs are housed in the new California Nanosystems Institute building at UCSB, which includes a unique research facility called the Allosphere, a three-story spherical immersive environment. Additional information about the department can be found at <http://www.mat.ucsb.edu>.

Applicants are expected to hold a doctoral degree in Media Arts and Sciences, Computer Science, or a closely related field, have demonstrated excellence in research, and have a strong commitment to teaching and interdisciplinary scholarship and/or creative activity.

The department is especially interested in candidates who can contribute to the diversity and excellence of the academic community through research, teaching, and service. Primary consideration will be given to applications received by December 15, 2008; however, the position will remain open until filled. Applications must include a CV, research and teaching statements, and at least three letters of reference. See <http://www.mat.ucsb.edu/recruit> for information on how to apply.

The University of California is an Equal Opportunity / Affirmative Action Employer.

University of California, Santa Barbara
Faculty Position in Computer Science

The Department of Computer Science at the University of California, Santa Barbara, has an open position in Computer Science for the forthcoming academic year 2009-10. We seek applications from outstanding candidates in all areas of computer science to fill this tenure-track position effective July 2009.

The Department of Computer Science has grown rapidly, both in size and stature, over the past 10 years, accompanied by a five-fold increase in extramural funding. The department, with 30 faculty and more than 100 doctoral students,

is part of the College of Engineering, which is ranked among the top 20 in the nation by the 2008 US News and World Report. Additional information about the department and our graduate program can be found at <http://www.cs.ucsb.edu>. Applicants are expected to hold a doctoral degree in Computer Science or a related field, show outstanding research potential, and have a strong commitment to teaching.

Primary consideration will be given to applications received by December 15, 2008; however, the position will remain open until filled. Applications should be submitted electronically as PDF documents to: <http://www.cs.ucsb.edu/recruit>. Applications must include a detailed resume, research and teaching statements, and the names and addresses of four references.

The Department is especially interested in candidates who can contribute to the diversity and excellence of the academic community through research, teaching, and service. We are an Equal Opportunity/Affirmative Action employer.

University of Cincinnati
College of Engineering

The University of Cincinnati's College of Engineering invites applications for the position of the Head of the Department of Computer Science (CS). The Head is expected to have a strong commitment to advancing research and education, to lead the development of innovative programs, especially joint ventures with other academic units, and to foster and strengthen external research support of the faculty from national funding agencies and academic partnerships with industry.

Min. Quals.: Qualifications for the Head position include a doctoral degree in computer science or a closely related field; a distinguished record in research and education; a clear vision for the future of the discipline; and established leadership and interpersonal skills. The Head is responsible for the overall program administration, including taking a leadership role in directing the growth and development of the Department. The Head is also expected to play an active role in fostering the recruitment of high quality students and faculty, and overseeing the implementation of the ongoing revitalization of the curriculum.

To apply for position (28UC1398), please see www.jobsatuc.com

The University of Cincinnati is an affirmative action/equal opportunity employer. UC is a smoke-free work environment.

University of Iowa
Assistant Professor

The department of Management Sciences is recruiting for a tenure track faculty position at the Assistant Professor level starting in fall 2009. We invite applications in the area of Management Information Systems, which includes but is not limited to the following: database, machine learning, data and text mining, knowledge management and other similar areas.

Candidates for this tenure track assistant professor position should have a Ph.D. in MIS, Informatics, Information Sciences, Computer Science or a related field and exhibit exceptional research promise. Extensive collaborative op-

portunities across departments and colleges are available. Particularly exciting opportunities for collaboration exist with faculty in Computer Science, Information Sciences, Health Informatics and Nursing Informatics. Candidates must have the interest and ability to teach at undergraduate, MBA and Ph.D. levels.

Further details and application instructions are provided at our website <http://www.biz.uiowa.edu/mansci>. Applicants should submit a cover letter, curriculum vita, summary of research interests, contact information for three references, and 1-3 research papers online at <http://www.biz.uiowa.edu/mansci/recruit>. Salary will be commensurate with qualifications. Applications screening will begin by January 1, 2009.

The University of Iowa is an Equal Opportunity/Affirmative Action Employer; Women, minority applicants, veterans and persons with disabilities are strongly encouraged to apply.

University of Iowa
Computer Science Department
Assistant Professor Position, Fall 2009

The Computer Science Department seeks applications for one tenure-track assistant professor position commencing August 2009. Applications from all areas of computer science and informatics are invited. We welcome applicants doing research at the frontiers of computing in connection with other disciplines.

The Department and the College of Liberal Arts and Sciences are strongly committed to gender and ethnic diversity; the strategic plans of the University, College, and Department reflect this commitment.

The Department offers BA, BS, MCS, and PhD degrees in Computer Science, and in Fall 2007 added BA and BS degrees in Informatics (see <http://www.cs.uiowa.edu/Informatics>).

Candidates must hold a PhD in computer science, informatics, or a closely related discipline. Applications received by January 15, 2009, are assured of full consideration. Applications should contain a CV and research and teaching statements.

URL for additional information and on-line application: <http://www.cs.uiowa.edu/hiring/>

The University of Iowa is an equal opportunity/affirmative action institution. Women and minorities are encouraged to apply.

University of Kentucky
Computer Science Positions
Assistant Professor Level

The University of Kentucky Computer Science Department invites applications for two tenure-track positions beginning August 15, 2009 at the assistant professor level in bio/medical informatics and in vision/graphics. Specific information about each position and the application process are available at <http://www.cs.uky.edu/employment/positions.php>.

Candidates must have a PhD in Computer Science.

The University of Kentucky Computer Science Doctoral Program recently ranked in the top 20% of such programs (30 out of 157) in a nationwide analysis. The rankings -- produced by Academic

Analytics -- are based on the Faculty Scholarly Productivity Index(tm), a measure of actual faculty publication, citation, and funding rates. Among doctoral programs at public universities, UKCS was ranked 16th.

The University of Kentucky is an equal opportunity employer and encourages applications from minorities and women.

**University of Kwazulu-Natal
Durban, South Africa**
School of Computer Science
Westville Campus
Professor/Associate Professor
Reference number: SA36/2008

The University of KwaZulu-Natal (UKZN) is one of the top 3 universities in South Africa and ranks amongst the top 500 universities in the world. The Westville campus is located in Durban on the east coast of South Africa. UKZN strives to attract and retain top quality academic staff. If you are looking for an opportunity to grow and develop professionally in a vibrant, supportive and challenging workplace, then a career at UKZN is the answer.

MINIMUM REQUIREMENTS:

FOR BOTH LEVELS:

- ▶ A PhD or equivalent degree in Computer Science
- Experience in supervision of postgraduate students or mentoring of junior staff
- Experience in curriculum development and teaching at tertiary level in Computer Science.

PROFESSOR:

- ▶ 10 years work experience at tertiary institution/s OR 10 years in appropriate industry/ies or research institute/s
- ▶ Current substantial and sustained research record in Computer Science evidenced by publications in accredited peer-reviewed journals
- Successful supervision of doctoral students.

ASSOCIATE PROFESSOR:

- ▶ 5 years work experience at tertiary institution/s OR 5 years in appropriate industry/ies or research institute/s;
- Independent research competence in Computer Science evidenced by international peer-reviewed conference and journal publications
- Successful supervision of masters and/or doctoral students.

CLOSING DATE: 28 November 2008.

Apply via e-mail at: recruitment-agsc@ukzn.ac.za quoting the relevant reference number. For full details of the post, advantages and application procedures, please access the University website at <http://hr.ukzn.ac.za/StaffVac9344.aspx>. For further details about the School of Computer Science, including research focus areas, please access the school web site at <http://www.cs.ukzn.ac.za>

University of Michigan, Ann Arbor
Department of Electrical Engineering and Computer Science
Computer Science and Engineering Division
Faculty Positions

Applications and nominations are solicited for faculty positions in the Computer Science and Engineering (CSE) Division and as part of an interdisciplinary cluster hire funded by the University President to strengthen expertise in the area of Data Mining, Learning, and Discovery with

Massive Datasets, for interdisciplinary faculty positions within the Computer Science and Engineering Division, the Medical School, School of Information, Astronomy, Ecology and Evolutionary Biology, and Statistics Departments.

Qualifications include an outstanding academic record, a doctorate or equivalent in computer engineering or computer science, and a strong commitment to teaching and research.

Candidates with a focus in the areas of artificial intelligence, security, programming languages and parallel computing, and cyber-physical systems are encouraged to apply. However, all computer science and engineering applications will be considered. Applications must be received by January 12, 2009.

To apply please complete the form at: <http://www.eecs.umich.edu/eecs/jobs/csejobs.html>

Electronic applications are strongly preferred, but you may alternatively send resume, teaching statement, research statement and names of three references to:

Professor Karem A. Sakallah, Chair, CSE
Faculty Search
Department of Electrical Engineering and Computer Science
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109-2121

The University of Michigan is a Non-Discriminatory/Affirmative Action Employer with an Active Dual-Career Assistance Program. The college is especially interested in candidates who can contribute, through their research, teaching, and/or service, to the diversity and excellence of the academic community.

University of Nebraska-Lincoln
Assistant Professor

We invite applications for a tenure track faculty position at the rank of Assistant Professor. We are looking for a faculty member who can establish a strong research and teaching program that will strengthen our programs in the area of Human Computer Interfaces and/or Software Engineering. Candidates must hold an earned doctorate in Computer Science or a closely related discipline by the date of employment.

To apply, visit <http://employment.unl.edu> and complete a Faculty/Administrative application for requisition number 080713. Attach a cover letter, a CV, and statements describing your proposed research and teaching to your application. The cover letter must include names and contact information for at least three references. Review of applications will begin on December 1, 2008, and will continue until the position has been filled. A more detailed advertisement can be viewed at <http://cse.unl.edu/search>. The University of Nebraska is committed to a pluralistic campus community through affirmative action, equal opportunity, work-life balance, and dual careers.

University of Northern Iowa
Tenure-track Assistant Professor

The Department of Computer Science at the University of Northern Iowa invites applications for a tenure-track assistant professor position to begin

August 2009. Applicants must hold a Ph.D. in Computer Science or a closely-related discipline. The department seeks candidates able to participate widely in the Computer Science curriculum, with preference given to candidates able to teach courses in Software Engineering or Computer Systems.

Detailed information about the position and the department are available at <http://www.cs.uni.edu/>

Applicants should submit a letter of application, a curriculum vitae, statements of research and teaching philosophies, and the names and contact information of at least three references to Eugene Wallingford, Search Chair, Department of Computer Science, University of Northern Iowa, Cedar Falls, Iowa 50614-0507, wallingf@cs.uni.edu.

Applications received by January 15, 2009, will be given full consideration. EOE/AA. UNI is a smoke-free campus.

University of Oregon
Department of Computer and Information Science
Faculty Position

The CIS department seeks applicants for one or more full-time tenure-track faculty positions beginning fall, 2009. We anticipate appointments at the rank of Assistant Professor; however, in the case of exceptionally qualified candidates appointments at any rank may be considered. The University of Oregon is an AAU research university located in Eugene, two hours south of Portland, and within one hour's drive of both the Pacific Ocean and the snow-capped Cascade Mountains.

The CIS department is housed within the College of Arts and Sciences and part of the recently dedicated Lorry Lokey Science Complex. The College appreciates the increasing role that computer science plays in other disciplines and supports our goals of strengthening our ties with the other sciences. Applicants interested in interdisciplinary research are encouraged to apply. We offer a stimulating and friendly environment for collaborative research both within the department and with other departments on campus. The CIS department is associated with the Cognitive and Decision Sciences Institute, the Computational Science Institute, the Neuro-Informatics Center, and the Computational Intelligence Research Laboratory.

This department recognizes that computer science is undergoing rapid change as an academic discipline, and accordingly seeks to hire faculty in emerging areas of computer science as well as more established areas including distributed computing, data mining, networking, computational science (visualization, high performance computing), and HCI (usability, accessibility, interfaces).

The CIS department offers B.S., M.S. and Ph.D. degrees. More information about the department, its programs and faculty can be found at <http://www.cs.uoregon.edu>, or by contacting the search committee at faculty.search@cs.uoregon.edu.

Applicants must have a Ph.D. in computer science or a closely related field, a demonstrated record of excellence in research and a strong commitment to teaching. The successful candidates are expected to conduct vigorous research pro-

grams, and to teach at both the undergraduate and graduate levels. Applicants should send their curriculum vitae, names of at least four references, a statement of research and teaching interests, and selected publications to: Faculty Search Committee, Dept. of Computer and Information Science, University of Oregon, Eugene, OR 97403-1202, email: faculty.search@cs.uoregon.edu.

Review of applications will begin January 5, 2009, and continue until the position is filled.

The University of Oregon is an equal opportunity/affirmative action institution committed to cultural diversity and compliant with the Americans with Disabilities Act. We are committed to creating a more inclusive and diverse institution and seek candidates with demonstrated potential to contribute positively to its diverse community.

University of Oregon

Department of Computer and Information Science

Faculty Position

The CIS department seeks applicants for one or more full-time tenure-track faculty positions beginning fall, 2009. We anticipate appointments at the rank of Assistant Professor; however, in the case of exceptionally qualified candidates appointments at any rank may be considered. The University of Oregon is an AAU research university located in Eugene, two hours south of Portland, and within one hour's drive of both the Pacific Ocean and the snow-capped Cascade Mountains.

The CIS department is housed within the College of Arts and Sciences and part of the recently dedicated Lorry Lokey Science Complex. The College appreciates the increasing role that computer science plays in other disciplines and supports our goals of strengthening our ties with the other sciences. Applicants interested in interdisciplinary research are encouraged to apply. We offer a stimulating and friendly environment for collaborative research both within the department and with other departments on campus. The CIS department is associated with the Cognitive and Decision Sciences Institute, the Computational Science Institute, the Neuro-Informatics Center, and the Computational Intelligence Research Laboratory.

This department recognizes that computer science is undergoing rapid change as an academic discipline, and accordingly seeks to hire faculty in emerging areas of computer science as well as more established areas including distributed computing, data mining, networking, computational science (visualization, high performance computing), and HCI (usability, accessibility, interfaces).

The CIS department offers B.S., M.S. and Ph.D. degrees. More information about the department, its programs and faculty can be found at <http://www.cs.uoregon.edu>, or by contacting the search committee at faculty.search@cs.uoregon.edu.

Applicants must have a Ph.D. in computer science or a closely related field, a demonstrated record of excellence in research and a strong commitment to teaching. The successful candidates are expected to conduct vigorous research programs, and to teach at both the undergraduate and graduate levels. Applicants should send their curriculum vitae, names of at least four references, a statement of research and teaching interests, and selected publications to: Faculty Search Committee, Dept. of Computer and

Information Science, University of Oregon, Eugene, OR 97403-1202, email: faculty.search@cs.uoregon.edu. Alternatively (and preferably) applications can be made on-line at <http://www.cs.uoregon.edu/Employment/application.cgi>.

Review of applications will begin January 5, 2009, and continue until the position is filled.

The University of Oregon is an equal opportunity/affirmative action institution committed to cultural diversity and compliant with the Americans with Disabilities Act. We are committed to creating a more inclusive and diverse institution and seek candidates with demonstrated potential to contribute positively to its diverse community.

University of Pennsylvania

Department of Computer and Information Science

Faculty Positions

The University of Pennsylvania invites applicants for tenure-track appointments in both experimental and theoretical computer science to start July 1, 2009. Tenured appointments will also be considered. Faculty duties include teaching undergraduate and graduate students and conducting high-quality research.

The Department of Computer and Information Science has undergone a major expansion, including new faculty positions and a new building, Levine Hall, which was opened in April 2003. Over the last few years, we have successfully recruited faculty in artificial intelligence, architecture, databases, machine vision, programming languages, security and graphics. We are now especially interested in candidates in architecture and systems, although outstanding candidates in other areas might also be considered. Successful applicants will find Penn to be a stimulating environment conducive to professional growth.

The University of Pennsylvania is an Ivy League University located near the center of Philadelphia, the 5th largest city in the US. Within walking distance of each other are its Schools of Arts and Sciences, Engineering, Medicine, the Wharton School, the Annenberg School of Communication, Nursing, Law, and Fine Arts. The University campus and the Philadelphia area support a rich diversity of scientific, educational, and cultural opportunities, major technology-driven industries such as pharmaceuticals, finance, and aerospace, as well as attractive urban and suburban residential neighborhoods. Princeton and New York City are within commuting distance.

To apply, please complete the form located on the Faculty Recruitment Web Site at:

<http://www.cis.upenn.edu/departamental/facultyRecruiting.shtml>

Electronic applications are strongly preferred, but hard-copy applications (including the names of at least four references) may alternatively be sent to:

Chair, Faculty Search Committee
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

Applications should be received by January 15, 2009 to be assured full consideration.

Applications will be accepted until positions are filled.

Questions can be addressed to faculty-search@central.cis.upenn.edu.

The University of Pennsylvania values diversity and seeks talented students, faculty and staff from diverse backgrounds. The University of Pennsylvania does not discriminate on the basis of race, sex, sexual orientation, gender identity, religion, color, national or ethnic origin, age, disability, or status as a Vietnam Era Veteran or disabled veteran in the administration of educational policies, programs or activities; admissions policies; scholarship and loan awards; athletic, or other University administered programs or employment.

The Penn CIS Faculty is sensitive to "two-body problems" and would be pleased to assist with opportunities in the Philadelphia region.

University of Rochester

Tenure Track Faculty Positions

The Department of Computer Science at the University of Rochester invites applications for tenure track faculty positions. We seek PhD level candidates in networking, HCI, graphics, and/or machine learning. In addition, we invite applications for a joint Computer Science/Electrical and Computer Engineering position in computer systems and circuits. For full job descriptions and application procedures, see <http://www.cs.rochester.edu/recruit>.

University of South Carolina

Faculty Position in Computer Science and Engineering

Applications are invited for a tenure-track position with a research emphasis in signal processing and data mining as applied to biological and ecological data. This is a tenure-track appointment in the Department of Computer Science and Engineering with a joint appointment in the School of the Environment. Candidates should have a doctorate in computer science, computer engineering, or a related discipline by fall 2009. Candidates for assistant professor positions are expected to have strong research potential as well as an interest in teaching at both the undergraduate and graduate level. For those embarking on their professional careers, department support will include low teaching loads, competitive salary and generous start-up funds. Candidates for associate or full professor positions must possess an exceptional record of high-quality funded research, teaching, and scholarship. This position is part of a cluster of interdisciplinary faculty hires in the area of forecasting ecological responses to climate change in coastal regions. Candidates will be expected to form strong research collaborations with other hires in the cluster in geography, biology, and environmental science while establishing a research record suitable for a position in Computer Science and Engineering.

The Department of Computer Science and Engineering is in the College of Engineering and Computing and offers bachelor's, master's, and doctoral degrees. We have had twelve hires since 2000 among the current faculty of 21, and our recent hires include seven CAREER award recipients. The University of South Carolina is located in Columbia, South Carolina's capital and tech-

nology center, and is the comprehensive graduate institution in the state, with an enrollment of more than 25,000 students. For more information, see <http://www.cse.sc.edu/>.

Applicants should apply to the Chair of the Search Committee, Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 or to clustersearch@cse.sc.edu and clearly indicate the cluster search in their cover letter. A curriculum vita, research and teaching statement, and the names and addresses of three references should be included. Applications will be accepted until the position is filled. The University of South Carolina does not discriminate in educational or employment opportunities or decisions for qualified persons on the basis of race, color, religion, sex, national origin, age, disability, sexual orientation or veteran status.

University of South Carolina Department Chair – Computer Science and Engineering

The Department of Computer Science and Engineering in the College of Engineering and Computing, University of South Carolina, seeks nominations and applications for the position of Department Chair. The Department offers bachelor's degrees in computer engineering, computer information systems, and computer science, M.S./M.E. and Ph.D. degrees in Computer Science and Engineering, a Master of Software Engineering, and a Certificate of Graduate Studies in Information Assurance and Security. The Department has 22 full-time faculty, an undergraduate enrollment of 313, a graduate enrollment of 88 students, and over \$1.5 million in research expenditures. New leadership in the College has made growth of the Department a high priority.

Qualified applicants are expected to have outstanding leadership and administrative skills, a strong record of research performance, dedication to education, and credentials (including a Ph.D. in computer science, computer engineering, or related field) commensurate with appointment as a full professor with tenure in the Department. Nomination letters should include statements regarding the nominee's relevant credentials. Applications should include a current resume, a statement of professional interests and vision, and the names, affiliations, and contact information (including email address and telephone number) of at least three references. Nominations and applications will be accepted until the position is filled and should be sent to Computer Science and Engineering Chair Search Committee, Office of the Dean, College of Engineering & Computing, University of South Carolina, Swearingen Engineering Center, 301 Main Street, Columbia, SC 29208. Applications may be sent by email to CSE-search@engr.sc.edu. The University of South Carolina is an Affirmative Action/Equal Opportunity Institution. Women and minorities are encouraged to apply.

University of Tennessee at Chattanooga Assistant/Associate Professor

UTC invites applications for a full-time, tenure track appointment in Computer Science and

Engineering, beginning January 1, 2009. The department seeks applicants with a Ph.D. in Computer Science or Computer Engineering, and experience/interest in high performance computing and interdisciplinary teaching and research. The CSE department (www.cs.utc.edu), part of the College of Engineering and Computer Science, offers an ABET accredited B.S. degree, a M.S. degree, and has received certification by the CNS, NSA, and DHA as a National Center of Academic Excellence in Information Assurance Education. The College is also home to the SimCenter and its graduate programs (MS/Ph.D.) in Computational Engineering.

To apply, please e-mail in Word or pdf format an application letter, resume and descriptions of teaching and research philosophies to Dr. Jack Thompson, Jack-Thompson@utc.edu. Also, please arrange for 3 letters of recommendation and a copy of your transcript listing the completion of your doctoral degree to:

Faculty Search Committee
Computer Science, Dept. 2302
The University of Tennessee at Chattanooga
615 McCallie Avenue
Chattanooga, TN 37403-2598

Screening of applicants who have provided complete information will begin immediately and continue until the position is filled. The University of Tennessee at Chattanooga is an equal employment opportunity/affirmative action/Title VI & IX/Section 504 ADA/ADEA institution, and, as such, encourages the application of qualified women and minorities.

University of Tennessee, Knoxville Tenure-track Faculty Positions

The Min Kao Department of Electrical Engineering and Computer Science (EECS) at The University of Tennessee, Knoxville seeks applications for tenure-track faculty positions in all areas of computer engineering, including but not limited to dependable and secure systems, wireless and sensor networks, embedded systems, and VLSI. The department is starting a new growth phase thanks to gifts from alumnus Dr. Min Kao and other donors plus additional state funding totaling over \$47.5 M for a new building and endowments for the department. Information about the EECS Department can be found at <http://www.eecs.utk.edu/>.

Candidates should have an earned Ph.D. in Electrical Engineering, Computer Engineering, Computer Science, or equivalent. Interested candidates should apply through the departmental web site at <http://www.eecs.utk.edu/jobs/faculty> and submit a curriculum vitae, research and teaching statement, and provide contact information for three references. Consideration of applications will begin on December 10, 2008, and the position will remain open until filled. The University of Tennessee is an EEO/AA/Title VI/Title IX/Section 504/ADA/ADEA institution in the provision of its education and employment programs and services. All qualified applicants will receive equal consideration for employment without regard to race, color, national origin, religion, sex, pregnancy, marital status, sexual orientation, age, physical or mental disability, or covered veteran status.

University of Texas at Austin Department of Computer Sciences Tenure-track Positions at All Levels

The Department of Computer Sciences of the University of Texas at Austin invites applications for tenure-track positions at all levels. Excellent candidates in all areas will be seriously considered, especially in Computer Architecture. All tenured and tenure-track positions require a Ph.D. or equivalent degree in computer science or a related area at the time of employment.

Successful candidates are expected to pursue an active research program, to teach both graduate and undergraduate courses, and to supervise graduate students. The department is ranked among the top ten computer science departments in the country. It has 46 tenured and tenure-track faculty members across all areas of computer science. Many of these faculty participate in interdisciplinary programs and centers in the University, including those in Computational and Applied Mathematics, Computational Biology, and Neuroscience.

Austin, the capital of Texas, is located on the Colorado River, at the edge of the Texas Hill Country, and is famous for its live music and outdoor recreation. Austin is also a center for high-technology industry, including companies such as IBM, Dell, Freescale Semiconductor, Advanced Micro Devices, National Instruments, AT&T, Intel and Samsung. For more information please see the department web page: <http://www.cs.utexas.edu/>

The department prefers to receive applications online, beginning November 15, 2008. To submit yours, please visit <http://services.cs.utexas.edu/recruit/faculty/>

If you do not have internet access, please send a curriculum vita, home page URL, description of research interests, and selected publications, and ask three referees to send letters of reference directly to:

Faculty Search Committee
Department of Computer Sciences
The University of Texas at Austin
1 University Station C0500
Austin, Texas 78712-0233 USA

Inquiries about your application may be directed to faculty-search@cs.utexas.edu. For full consideration of your application, please apply by January 15, 2009. Women and minority candidates are especially encouraged to apply. The University of Texas is an Equal Opportunity Employer.

University of Waterloo David R. Cheriton School of Computer Science Chair in Software Systems

Applications are invited for one or two David R. Cheriton Chairs in Software Systems. These are senior positions and include substantial research support and teaching reduction. Candidates with outstanding research records in software systems (very broadly defined) are encouraged to apply. Successful applicants who join the University of Waterloo are expected to be leaders in research, have an active graduate student program and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with qualifications.

surate with experience, and appointments are expected to commence during the 2009 calendar year. The Chairs are tenured positions.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: <http://www.cs.uwaterloo.ca/>

Applications should be sent by electronic mail to cs-recruiting@cs.uwaterloo.ca

or by post to:

Chair, Advisory Committee on Appointments
David R. Cheriton School of Computer Science

200 University Avenue West
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

An application should include a curriculum vitae, statements on teaching and research, and the names and contact information for at least three referees. Applicants should ask their referees to forward letters of reference to the address above. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

University of Waterloo

David R. Cheriton School of Computer Science *Faculty Position - Information Retrieval will be given High Priority*

The University of Waterloo invites applications for a tenure-track or tenured faculty positions in the David R. Cheriton School of Computer Science. The area of information retrieval (broadly defined) will be given high priority: other areas will be considered if the School is unable to recruit a very strong candidate in information retrieval. Candidates at all levels of experience are encouraged to apply. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2009 calendar year.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: <http://www.cs.uwaterloo.ca/>

Applications should be sent by electronic mail to cs-recruiting@cs.uwaterloo.ca

or by post to:

Chair, Advisory Committee on Appointments
David R. Cheriton School of Computer Science

200 University Avenue West
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

An application should include a curriculum vitae, statements on teaching and research, and the names and contact information for at least three referees. Applicants should ask their referees to forward letters of reference to the address above. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

University of Waterloo

David R. Cheriton School of Computer Science *Faculty Position in Information Systems*

The University of Waterloo invites applications for one or two tenure-track or tenured faculty positions in the David R. Cheriton School of Computer Science, in the area of information systems. Candidates at all levels of experience are encouraged to apply. Preference will be given to those who focus on health informatics as an application area. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, contribute to a newly-created Master's program in health informatics, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2009 calendar year.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied re-

search and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: <http://www.cs.uwaterloo.ca/>

Applications should be sent by electronic mail to cs-recruiting@cs.uwaterloo.ca

or by post to:

Chair, Advisory Committee on Appointments
David R. Cheriton School of Computer Science

200 University Avenue West
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

An application should include a curriculum vitae, statements on teaching and research, and the names and contact information for at least three referees. Applicants should ask their referees to forward letters of reference to the address above. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

University of Waterloo

David R. Cheriton School of Computer Science *Faculty Position in Software Engineering*

The University of Waterloo invites applications for a tenure-track or tenured faculty position in the David R. Cheriton School of Computer Science, in the area of software engineering. Candidates at all levels of experience are encouraged to apply. Preference will be given to those who focus on health informatics as an application area. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2009 calendar year.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University

has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: <http://www.cs.uwaterloo.ca/>

Applications should be sent by electronic mail to cs-recruiting@cs.uwaterloo.ca or by post to:

Chair, Advisory Committee on Appointments
David R. Cheriton School of Computer Science
200 University Avenue West
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

An application should include a curriculum vitae, statements on teaching and research, and the names and contact information for at least three referees. Applicants should ask their referees to forward letters of reference to the address above. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

Vassar College

Department of Computer Science
Two Year Visiting Assistant Professor,
Fall, 2009

Vassar College seeks applications for a two-year, full-time Visiting Assistant Professor position starting Fall, 2009. A commitment to excellence in undergraduate teaching and research is expected. The Ph.D. in computer science is required. Applicants with background in any area of Computer Science will be considered, but special consideration will be given to applicants with interest and expertise in Computer Organization, Computer Architecture, Operating Systems, and/or the "TeachScheme, ReachJava" curriculum. All candidates must be able to cover courses in the core areas of Computer Science.

Vassar College is an equal opportunity/affirmative action employer and is actively committed to diversity within its community. Applications from members of historically under-represented groups are especially encouraged to apply.

Vassar College has been successfully building a strong undergraduate program in Computer Science. Introductory courses are taught using Scheme and Java. The department has Linux laboratories for introductory and advanced instruction. Faculty are provided with Unix workstations and personal computers. For more information see <http://www.cs.vassar.edu>.

Review of applications will begin January 1, 2009 and continue until the position is filled. Send vita and three letters of reference to Nancy Ide, Chair, Department of Computer Science, 124 Raymond Avenue, Box 732, Vassar College, Poughkeepsie, New York 12604-0732. E-mail: cs-dept@cs.vassar.edu.

Washington University in Saint Louis

Multiple Tenure-track/Tenured Faculty Positions

The Department of Computer Science and Engineering (CSE) and the School of Medicine (WUSM) are jointly searching for multiple tenure-track faculty members with outstanding records of computing research and long term interest in scientific and/or biomedical problems. Appointments may be made wholly within CSE or jointly with the Departments of Medicine or Pathology & Immunology.

A key initiative in the CSE department's strategic plan is Integrating Computing and Science. As part of that initiative, we expect to make synergistic hires with a combined research portfolio spanning the range from fundamental computer science/engineering to applied research focused on science or medicine. Specific areas of interest include, but are not limited to:

- ▶ Databases, medical informatics, clinical or public-health informatics
- ▶ Theory/Algorithms with the potential for biomedical applications
- ▶ Analysis of complex genetic, genomic, proteomic, and metabolomic datasets
- ▶ Image analysis or visualization with the potential for biomedical applications
- ▶ Computer engineering with applications to medicine or the natural sciences
- ▶ Other areas of computational biology or computational science

These positions will continue a successful, ongoing strategy of collaborative research between CSE and the School of Medicine, which is consistently ranked among the top 3 medical schools in the United States. CSE currently consists of 24 tenured and tenure-track faculty members, 71 Ph.D. students, and a stellar group of undergraduates with a history of significant research contributions. The Department seeks to build on and complement its strengths in biological sequence analysis, biomedical image analysis, and biomedical applications of novel computing architectures.

Washington University is a private university with roughly 6,000 full-time undergraduates and 6,000 graduate students. It has one of the most attractive university campuses anywhere, and is located in a lovely residential neighborhood, adjacent to one of the nation's largest urban parks, in the heart of a vibrant metropolitan area. St. Louis is a wonderful place to live, providing access to a wealth of cultural and entertainment opportunities without the everyday hassles of the largest cities.

We anticipate appointments at the rank of Assistant Professor; however, in the case of exceptionally qualified candidates appointments at any rank may be considered. Qualified applicants should submit a complete application (cover letter, curriculum vita, research statement, teaching statement, and names of at least three references) electronically to recruiting@cse.wustl.edu. Other communications may be directed to Prof. Michael Brent, Department of Computer Science and Engineering, Campus Box 1045, Washington University, One Brookings Drive, St. Louis, MO 63130-4899.

Applications will be considered as they are received. Washington University is an equal opportunity/affirmative action employer.

Wayne State University

Faculty Positions in Bioinformatics/Computational Biology

The College of Liberal Arts and Sciences anticipates up to three tenure-track openings in bioinformatics/computational biology that will build on our existing strengths in computer sciences and life sciences. The new faculty will be housed in the Department of Computer Science or Department of Biological Sciences according to their background and preference. Rank will be dependent upon qualifications. As part of this cluster hire, we anticipate the creation of a Center for Computational Biology and Bioinformatics. Joint appointments in other Schools/Colleges/Centers/Institutes will be available, as appropriate.

Areas of interest include, but not limited to, development of high throughput data analysis methods and techniques (e.g., microarray, sequence analysis), systems biology, data mining, and bioinformatics/computational biology applications (e.g., genomics, proteomics, ribonomics, metabolomics, genomic medicine and disease).

Wayne State University is a, comprehensive, nationally ranked research institution that offers generous start-up packages. Applicants must have a Ph.D. degree or equivalent and an outstanding research record. Successful applicants are expected to establish and maintain vigorous, externally funded research programs and participate in education at both the undergraduate and graduate levels. In addition, successful applicants will be expected to exploit and expand the existing collaborative culture in the campus among faculty members of Biological Sciences, Computer Science, the School of Medicine's Karmanos Cancer Institute, the Center for Molecular Medicine and Genetics, the Institute of Environmental Health Sciences, the Mott Center for Human Growth and Development, Detroit Regional Institute for Clinical and Translational Research and the Henry Ford Health System.

Applications must be submitted online at <http://jobs.wayne.edu>, referring to Posting #035531. Include a letter of intent, 2-page statement of research interests, 1-page statement of teaching interests, curriculum vitae, and contact information for at least three references. All applications will be strictly confidential and references will be requested only for applicants who are shortlisted. Applicants must indicate the position(s) for which they are applying by marking the corresponding Position Number(s). Only those application materials that are submitted to this site will be considered. The search will remain open until the positions have been filled. Wayne State University is an equal opportunity/affirmative action employer.

Wayne State University

Department of Computer Science

Tenure-Track Faculty Positions

The Department of Computer Science of Wayne State University invites applications for two tenure-track faculty positions, subject to administrative approval, at the Assistant/Associate Professor level. Continuing our recent growth, we are seeking applicants in the areas of Software Engineering and Services Computing. Outstanding applications in other areas will also be considered.

Candidates should have a Ph.D. in computer science or related area. The successful candidate will have a strong commitment to research and teaching, a strong publication record, and potential for obtaining external research funding. Senior applicants should have strong publication and funding records.

We offer B.S., M.S. and Ph.D. degrees with enrollment of over 80 Ph.D. students. Our total annual R&D expenditures average between \$2-3 million.

Wayne State University is a premier institution of higher education offering more than 350 undergraduate and graduate academic programs through 11 schools and colleges to more than 33,000 students. Wayne State ranks in the top 50 nationally among public research universities. As Michigan's only urban university, Wayne State fulfills a unique niche in providing access to a world-class education. The University offers excellent benefits and a competitive compensation package.

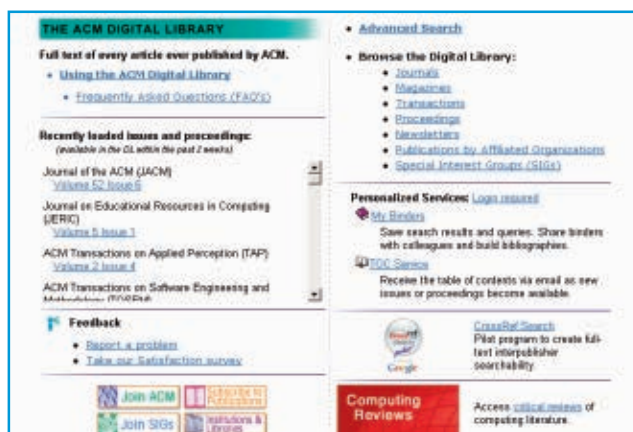
Submit applications online at <http://jobs.wayne.edu>. Please include a letter of intent, statement of research and teaching interests, CV, and contact information for at least three references. All applications received by December 1, 2008 will receive full consideration. However, applications will be accepted until the position is filled. Wayne State University is an equal opportunity/affirmative action employer.

Willamette University Assistant Professor

We invite applications for a tenure-track position at a small, select liberal arts college, located in Salem, Oregon, one hour from the Pacific, the Cascades, and the Silicon Forest. The position will complete a department of four full-time faculty. The teaching load is two courses and two labs per term. Startup funding, junior sabbaticals and conference travel are available. Qualifications include a Ph.D. in Computer Science, a commitment to high-quality teaching, an interdisciplinary focus in teaching and scholarship, and an ability to conduct research with students. Please send a letter of application, a statement of teaching philosophy and vitae (including names of three references) to: cs-search@willamette.edu. Applications will be reviewed starting 11/21/08. Believing that diversity contributes to academic excellence and to rich and rewarding communities, Willamette University is committed to recruiting and retaining a diverse faculty, staff and student body. We seek candidates, particularly those from historically under-represented groups, whose work furthers diversity and who bring to campus varied experiences, perspectives and backgrounds.

ACM Digital Library

www.acm.org/dl



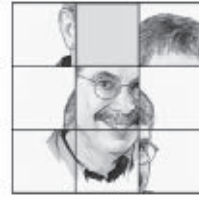
The Ultimate Online INFORMATION TECHNOLOGY Resource!

- Over 40 ACM publications, plus conference proceedings
- 50+ years of archives
- Advanced searching capabilities
- Over 2 million pages of downloadable text

Plus over one million bibliographic citations are available in the ACM Guide to Computing Literature

To join ACM and/or subscribe to the Digital Library, contact ACM:

Phone: 1.800.342.6626 (U.S. and Canada)
+1.212.626.0500 (Global)
Fax: +1.212.944.1318
Hours: 8:30 a.m.-4:30 p.m., Eastern Time
Email: acmhelp@acm.org
Join URL: www.acm.org/joinacm
Mail: ACM Member Services
General Post Office
PO Box 30777
New York, NY 10087-0777 USA



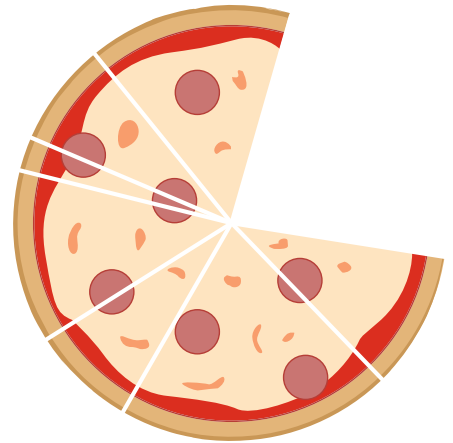
DOI:10.1145/1400214.1413439

Peter Winkler

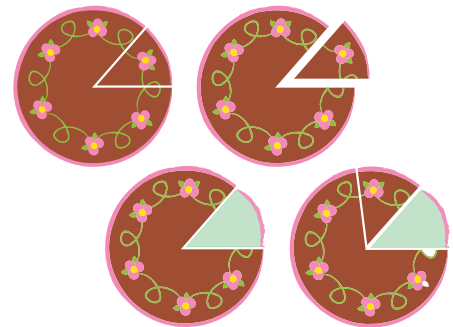
Puzzled: Circular Food

Welcome to three new challenging mathematical puzzles. Solutions to the first two will be published next month; the third is as yet unsolved, so you may need extra luck with that one. Here, I concentrate on circular food, so you might want to eat something before jumping in.

1. Two hungry Icelanders, Alta and Baldur, are sharing a pizza that is radially sliced into pieces of various sizes. To start, Alta chooses any slice to eat; thereafter, she and Baldur alternate taking slices (Baldur first) but must always take a slice that is adjacent to some previously taken slice. I call this the “polite pizza protocol”; the uneaten portion is always connected, thus, in theory, staying hot longer. Is it possible for the pizza to be cut in such a way that, no matter what Alta does, Baldur can get more than half the pie?



2. A cylindrical ice-cream cake with the most scrumptious chocolate frosting on top is sitting on a table. As an expert cake cutter, you choose an arbitrary angle x and proceed to cut one wedge after another, counterclockwise, around the cake, each of angle exactly x . However, each time you cut a wedge, you turn that piece upside-down and slide it back into the cake. This puts the frosting on the bottom at first, but as you work your way around and around the cake, the frosting comes back up to the top, then returns to the bottom, and so forth. Your mission is to prove that after some finite number of slices, all the frosting will be back on top of the cake.



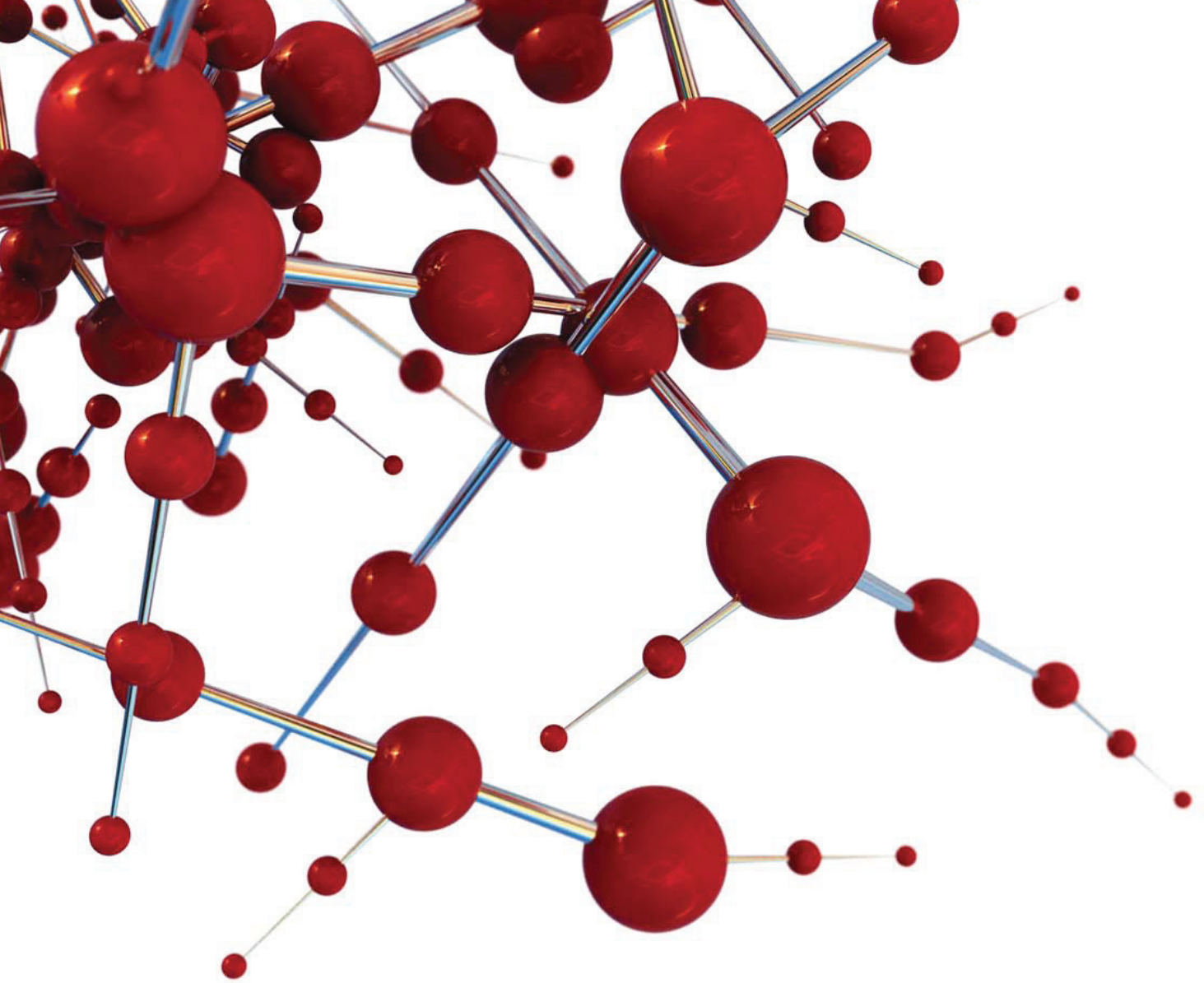
3. You need to bake circular tarts of various sizes and total area 1. Can you always fit them into a circular pie pan of area 2? Amazingly, no one knows.

(This puzzle is from Alexander Soifer of the University of Colorado, Colorado Springs; related work, including a proof for square tarts in a square pan, can be found in the journal *Geombinatorics*, www.uccs.edu/~geombina/.)



Readers are encouraged to submit prospective puzzles for future columns to puzzled@cacm.acm.org.

Peter Winkler (puzzled@cacm.acm.org) is Professor of Mathematics and of Computer Science and Albert Bradley Third Century Professor in the Sciences at Dartmouth College, Hanover, NH.



**CONNECT WITH OUR
COMMUNITY OF EXPERTS.**

www.reviews.com



Association for
Computing Machinery

Reviews.com

They'll help you find the best new books
and articles in computing.

Computing Reviews is a collaboration between the ACM and Reviews.com.

introducing...

ACM's *Newly Expanded* Online Books & Courses Programs!

Helping Members Meet Today's Career Challenges



3,000+ Online Courses from SkillSoft

The ACM Online Course Collection features **unlimited access to 3,000+ online courses** from SkillSoft, a leading provider of e-learning solutions. This new collection of courses offers a host of valuable resources that will help to maximize your learning experience. Available on a wide range of information technology and business subjects, these courses are open to ACM Professional and Student Members.



SkillSoft courses offer a number of valuable features, including:

- **Job Aids**, tools and forms that complement and support course content
- **Skillbriefs**, condensed summaries of the instructional content of a course topic
- **Mentoring** via email, online chats, threaded discussions - 24/7
- **Exercises**, offering a thorough interactive practice session appropriate to the learning points covered previously in the course
- **Downloadable content** for easy and convenient access
- **Downloadable Certificate of Completion**

"The course Certificate of Completion is great to attach to job applications!"

ACM Professional Member

600 Online Books from Safari

The ACM Online Books Collection includes **unlimited access to 600 online books** from Safari® Books Online, featuring leading publishers including O'Reilly. Safari puts a complete IT and business e-reference library right on your desktop. Available to ACM Professional Members, Safari will help you zero in on exactly the information you need, right when you need it.



Association for
Computing Machinery

Advancing Computing as a Science & Profession

500 Online Books from Books24x7

All Professional and Student Members also have **unlimited access to 500 online books** from Books24x7®, in ACM's rotating collection of complete unabridged books on the hottest computing topics. This virtual library puts information at your fingertips. Search, bookmark, or read cover-to-cover. Your bookshelf allows for quick retrieval and bookmarks let you easily return to specific places in a book.



pd.acm.org
www.acm.org/join