09.16

# Computer

## EMERGING COMPUTING PARADIGMS

ENTROPY AS A SERVICE, P. 98
GOVERNMENT IT MODERNIZATION, P. 114

IEEE

IEEE computer society
CELEBRATING 70 YEARS

www.computer.org/computer

FAIR AND TRANSPARENT APPOINTMENT NEGOTIATIONS
DHV'S HALLMARK

The Universität der Bundeswehr München (Bundeswehr University Munich) is significantly expanding its Cyber Defence Research Center (CODE). CODE was established in 2013 with the objective to bring together experts from different faculties and scientific disciplines as well as expertise from industry and government agencies to conduct research in the cyber and information space. CODE pursues a comprehensive, integrated, and interdisciplinary approach to implement technical innovations and concepts for the protection of data, software, and ICT infrastructures in accordance with legal and commercial framework conditions. It has already established important strategic partnerships in this area. The objective of the expansion is to unite the research initiatives of the Bundeswehr and the Federal Government in the area of Cyber Defence and Smart Data and to establish the CODE Research Center as the primary point of contact in the cyber and information domain of the Bundeswehr and the Federal Government.

Research and teaching in the area of cyber security is already being carried out as part of the Bachelor's and Master's programs in the Computer Science Department. According to current planning, a new international Master's program in Cyber Security will be launched on January 1st, 2018.

The Universität der Bundeswehr München will therefore be appointing **eleven Professors** for its Computer Science Department on **October 1st, 2017.**

The Universität der Bundeswehr München is looking for personalities with outstanding scientific qualifications to fill these professorial positions, who will also contribute actively to the CODE research center. Besides excellent research work, the new professors are expected to develop demanding lectures, practicals, and seminars for the new Master's program in Cyber Security and to provide excellent teaching in their respective specialist area. Applicants are also expected to carry out teaching in the Bachelor's programs in Computer Science and Business Informatics, and to work closely with the other departments at the Universität der Bundeswehr München.

The Professorships will be provided with eight excellently equipped laboratories housed in a new building that is to be completed in the near future.

The candidates must have an excellent scientific track record, as demonstrated by a habilitation or equivalent scientific achievements, as well as significant excellent publications in academic journals. Proven teaching experience in their respective specialist area is highly desired. The new Professors should have an international perspective, e.g., based on participation in international research projects, and experience in acquiring third-party funding. The duties will also include active participation in the university's academic self-administration.

The Computer Science Department at the Universität der Bundeswehr München is seeking Professors for the following specialist areas of its Cyber Defence und Smart Data Research Center:

## University Professorship (W3) in Cryptography

When it comes to transmitting, storing, and processing data, cryptographic methods are crucial to ensure that the data remains confidential, authenticated, and uncorrupted.
The remit of the Professorship includes encryption algorithms, random number generators and key management as well as their practical application within communication protocols – both from a provider's and an attacker's perspective.

## University Professorship (W3) in ICT Threat and Malware Analysis

The complexity and heterogeneity of communication networks and ICT infrastructures requires a systematic assessment of potential attack vectors in order to derive priorities for protection mechanisms.
Besides malware as a mass phenomenon in the form of viruses, Trojan horses and encryption-based ransomware, the Professorship also focuses on constructing sandbox analysis environments to identify malicious code in third-party software, such as smartphone apps and other software downloaded from the web.

## University Professorship (W3) in Business Intelligence Security

Networked applications, IT services, and operating environments generate vast amounts of data that can be used for various purposes, including the detection of potential attacks, but the sheer amount of such data prevents any manual processing.
The research tasks involved in the Professorship in Business Intelligence Security include big data and smart data algorithms for the aggregation, the correlation, and the analysis of large data amounts associated with security events for the purpose of providing specific support for decision processes concerning the prevention, the detection, and the response to attacks.

## University Professorship (W3) in Cyber Physical System Security

The combination of networked applications with mechanical and electronic components, such as in industrial production facilities, assistance systems, and energy supply systems, has many advantages – but it also involves the risk that these cyber physical systems can be compromised or sabotaged through cyber attacks.
The Professorship focuses on the information security characteristics of cyber physical systems with their specific framework conditions, such as the constrained resources of embedded systems, real-time capabilities, and physical access by attackers.

Universität der Bundeswehr München

## University Professorship (W3) in Data Protection and Compliance

The development and procurement of complex software systems must comply with legal and sector-specific regulations, which must already be taken into account during the requirements analysis and planning stage. Ultimately, proof of compliance must also be demonstrated, e.g., by means of a certification.

The Professorship in Data Protection and Compliance focuses on the methods and tools needed for the technical implementation of requirements resulting from, for example, the European General Data Protection Regulation and the new German IT Security Act, and it will also develop IT concepts for the implementation of the privacy-by-design paradigm.

## University Professorship (W3) in Forensic Methods and IT Security

In order to fully investigate and resolve security incidents, evidence must be gathered and analyzed while maintaining the integrity of the chain of custody. In order to deal with sophisticated attacks that are increasingly leaving fewer traces in compromised IT systems, the concealment of traces, the use of encryption and, for example, requirements regarding the automation and the scalability of digital forensics, the Professorship will research new approaches to analyzing the main and background memories of mobile and stationary systems as well as networked applications, among others topics.

## University Professorship (W3) in Open Source Intelligence and Situation Assessment

The situation assessment and documentation in cyber space as a basis for planning and decision processes can no longer be completely achieved using dedicated sensors and traditional reporting channels. Instead, the analysis of public sources, such as social media and internet communities, has become crucial to ensure the timeliness and correct focus of situation reports.

The Professorship in Open Source Intelligence and Situation Assessment will develop methods for continuous information gathering in networked environments using experience and findings from previous operations, as well as methods to compile clear and concise situation reports.

## University Professorship (W3) in Privacy Enhancing Technologies

As users often act carelessly with their personal data and internet providers frequently favor commercial interests over protective measures, it is becoming easier and less risky for organized crime to commit, among other things, identity theft.

The Professorship in Privacy Enhancing Technologies will research new approaches to eliminate or at least minimize the unnecessary or unwanted disclosure of information during the use of networked applications and to ensure that stored data is not analyzed and forwarded to third parties without user consent.

## University Professorship (W3) in IT Vulnerability Management and Security Testing

Programming errors, insufficient adaptation to new operating environments, and negligent use of IT systems often lead to vulnerabilities that allow attackers to obtain unauthorized access to processed data or even take control of entire systems.

The Professorship in IT Vulnerability and Security Testing will deal with the systematic handling of such vulnerabilities in IT systems and testing methods for their identification and assessment, so that, for instance, penetration tests of networked applications can be employed to determine areas in which the security levels need to be improved.

## University Professorship (W3) in Secure Software Development

The security of software crucially depends on the priority assigned to information security attributes during the requirements analysis, system design, and programming as well as the associated testing and approval procedures.

In the Professorship in Secure Software Development, the two disciplines of software engineering and security engineering overlap. It deals with methods, algorithms, and tools required for the implementation of software according to the secure-by-design, secure-by-default, and secure-in-deployment paradigms.

## University Professorship (W3) in Usable Security and Privacy

Information security and data protection necessitate the implementation and use of technical mechanisms that are too complex to apply for many users. E-mail encryption procedures, for instance, have only been utilized by IT experts for decades because the time and effort necessary for their use is much too high compared to their subjective benefits.

The Professorship in Usable Security and Privacy will investigate user-friendly approaches to security and data protection procedures and to the implementation of graphical interfaces for human-computer interaction in order to enhance the usability and therefore ensure the extensive employment of important protective measures.

The Universität der Bundeswehr München offers academic programs directed primarily at Officer Candidates and Officers, who can obtain Bachelor's and Master's degrees within a trimester system. Depending on spare capacity, civilian students are allowed to enroll. The study course is complemented by interdisciplinary elements in an integrated program entitled „studium plus".

Preconditions of employment and the legal duty positioning of Professors are based upon the "Bundesbeamtengesetz". Employment as a "Beamtin/ Beamter" requires that the candidate is not older than 50 at the date of appointment.

The University seeks to increase the number of female Professors and thus explicitly invites women to submit applications. Severely disabled candidates with equal qualifications will receive preferential consideration.

Please submit your application documents marked as Confidential Personnel Matter to the Department Head of the Computer Science Department at *the Universität der Bundeswehr München, 85577 Neubiberg,* by **October 15th, 2016**.

NEW MEMBERSHIP OPTIONS FOR A BETTER FIT.
PREFERRED PLUS
TRAINING & DEVELOPMENT
RESEARCH
BASIC
STUDENT

# Achieve your career goals with the fit that's right for you.

## Explore your options below.

| Select your membership | Preferred Plus | Training & Development | Research | Basic | Student |
|---|---|---|---|---|---|
| | **$60** *IEEE Member*<br>**$126** *Affiliate Member* | **$55** *IEEE Member*<br>**$115** *Affiliate Member* | **$55** *IEEE Member*<br>**$115** *Affiliate Member* | **$40** *IEEE Member*<br>**$99** *Affiliate Member* | **$8** *Does not include IEEE membership* |
| *Computer* magazine (12 digital issues)* | ✓ | ✓ | ✓ | ✓ | ✓ |
| *ComputingEdge* magazine (12 issues) | ✓ | ✓ | ✓ | ✓ | ✓ |
| Members-only discounts on conferences and events | ✓ | ✓ | ✓ | ✓ | ✓ |
| Members-only webinars | ✓ | ✓ | ✓ | ✓ | ✓ |
| Unlimited access to *Computing Now*, computer.org, and the new mobile-ready myCS | ✓ | ✓ | ✓ | ✓ | ✓ |
| Local chapter membership | ✓ | ✓ | ✓ | ✓ | ✓ |
| Safari Books Online (600 titles and 50 training videos) | ✓ | ✓ | | | ✓ |
| Skillsoft online solutions (courses, certifications, practice exams, videos, mentoring) | ✓ | ✓ | | | ✓ |
| Two complimentary Computer Society magazines | ✓ | | ✓ | | |
| myComputer mobile app | *30 tokens* | | *30 tokens* | | *30 tokens* |
| Computer Society Digital Library | *12 FREE downloads* | *Member pricing* | *12 FREE downloads* | *Member pricing* | *Included* |
| Training webinars | *3 FREE webinars* | *3 FREE webinars* | *Member pricing* | *Member pricing* | *Member pricing* |
| Priority registration to Computer Society events | ✓ | | | | |
| Right to vote and hold office | ✓ | ✓ | ✓ | ✓ | |
| One-time 20% Computer Society online store discount | ✓ | | | | |

* *Print publications are available for an additional fee. See catalog for details.*

www.computer.org/membership

IEEE computer society

# Computer

IEEE COMPUTER SOCIETY http://computer.org // +1 714 821 8380
*COMPUTER* http://computer.org/computer // computer@computer.org

IEEE
IEEE computer society

SUSTAINABLE FORESTRY INITIATIVE Certified Sourcing www.sfiprogram.org SFI-01042

BPA WORLDWIDE™

# Computer

MULTIMEDIA

SEPTEMBER 2016

## FEATURES

SEPTEMBER 2016

# CONTENTS

## ABOUT THIS ISSUE

*As computers based on silicon and conventional architecture reach their limits, it is time to explore and harness new computing paradigms.*

## FEATURES CONTINUED

## RESEARCH FEATURES

▶ See **www.computer.org/computer-multimedia** for multimedia content related to the features in this issue

## COLUMNS

## Departments

## Membership News

**Circulation:** *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036. IEEE Computer Society membership includes a subscription to *Computer* magazine.

**Postmaster:** Send undelivered copies and address changes to *Computer*, IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in USA.

70TH ANNIVERSARY MILESTONES

# IEEE Computer Society Membership

The IEEE Computer Society traces its origins to the 1946 formation of the Subcommittee on Large-Scale Computing Devices of the American Institute of Electrical Engineers (AIEE). In 1963, the AIEE merged with the Institute of Radio Engineers (IRE) to become the Institute of Electrical and Electronics Engineers (IEEE). The combination of the predecessor organizations' computing-related groups continued to provide a means for establishing lifelong and diverse professional relationships in what was one of the first generations of computer engineers. With the Society's formal establishment in 1971, early successes were extended and deepened, ushering IEEE into the new digital computing era characterized by unparalleled technological innovation. As IEEE's largest society, the Computer Society supports the computing profession's advancement through networking and volunteer leadership opportunities, conferences, publications, standards, education, and global innovation.

The advent of *Transactions on Computers* and *Computer* (formerly *Computer News Group*) helped membership double by the end of the 1960s with 16,862 members, including 4,200 students, 158 affiliates, and 41 chapters. By the end of the 1970s, Computer Society membership had nearly tripled to 43,930, including 7,833 students and 3,943 affiliates. There were more than 100 chapters, including 30 student branch chapters. Today, the Society—with nearly 60,000 members and 350 chapters across 168 countries—is at the forefront of a profound technological revolution that is well poised to empower the next generation of computing professionals who will continue to bring unprecedented change to the modern world. C

—*Lori Cameron*

IEEE computer society
CELEBRATING 70 YEARS

# 32 & 16 YEARS AGO

EDITOR NEVILLE HOLMES
holmeswn@yahoo.com.au

## SEPTEMBER 1984

*www.computer.org/csdl/mags/co/1984/09/index.html*

**Letter** (p. 7) "The decision of whether to discard these packets [those queued for a failed link] or to requeue these packets to other links is an important but a difficult design choice. Several drawbacks can be identified in rerouting the packets."

**Introduction** (p. 8) "[T]o my knowledge, this special issue on AI for human–machine interface is the first of its kind. Active and well-known researchers in [AI] have contributed to it; the first three articles are general in nature, whereas the remaining four are applications dependent."

**Tutorial** (p. 11) "This article presents an overview of the field of knowledge engineering. It describes the major developments that have led up to the current great interest in expert systems, then presents a brief discussion of the principal scientific and engineering issues in the field, as well as of the process of building expert systems, the role of tools in that work, how expert systems perform human–computer interface functions, and the frontiers of research and development."

**User Interface Structure** (p. 29) "We have adopted the view that these [end-user] problems arise primarily when implicit knowledge about individual tasks is encoded in the user interface. What results is an unclear separation between actual computations involved in the execution of a task and data acquisitions (often from an end user) needed to execute a task. We have therefore experimented with a system design in which the user interface has a minimal a priori knowledge of individual tasks."

**Linguistic Interfaces** (p. 39) "There are, of course, many ways in which people can communicate with programs. Some are linguistic: they use a verbal language of some kind, be it English or an artificial command language. Others are nonlinguistic, relying instead on such techniques as pointing or drawing. This article concentrates on the use of natural languages, such as English."

**Diagnostic Aid** (p. 51) "Patrec is one component in a cluster of knowledge-based medical expert systems developed at Ohio State University. This system holds extensive knowledge on relevant medical data entities; it uses this knowledge to acquire and organize data about patients and to answer questions about patient data needed for diagnostic reasoning."

**Automated Teaching** (p. 61) "In this article we discuss how a deep understanding of a student can be constructed in an [AI] program and how this understanding, coupled with a facility for language generation, can be used to build a flexible machine tutor."

**Job-Shop Autonomy** (p. 76) "[M]anual planning and control methods limit our ability to utilize the flexibility afforded by robotic technology. We investigate the issues involved in constructing software systems for the planning and control of activities in the job-shop."

**Decision Support** (p. 89) "Machine representation of models involves development of an adequate description of model parameters as well as the valid conditions under which a model can be executed with meaningful outputs. The approach presented here features a flexible representation of models so that the same models can be used in different contexts."

**Process Control** (p. 108) "A workshop on reliable process controls was held at the Total Systems Reliability Symposium, sponsored by the IEEE Computer Society, the IEEE Reliability Society, and the National Bureau of Standards at NBS 12–14 December 1983. Its two workshop sessions attracted approximately 40 participants."

**International Conference** (p. 111) "Culminating some four years of planning and organizing, the Chinese Institute of Electronics Computer Society and the IEEE Computer Society held the First International Conference on Computers and Applications in Beijing, the People's Republic of China, 20–22 June 1984. The conference attracted over 250

0018-9162/16/$33.00 © 2016 IEEE

specialists in computer science and engineering from some 18 different countries."

**Replacing I/O** (p. 120) "Why not 'gigoput'? GIGO, as everyone knows, stands for 'garbage in, garbage out,' so the meaning of gigoput should be obvious to all. Nor is it a serious misrepresentation of the situation, since even for a correctly working computer, most of the input/output is irrelevant to the user."

**Irradiation** (p. 136) "In order to comply with a Federal Communications Commission regulation issued in 1982, the office equipment industry needs improved procedures and techniques that will minimize the amount of unwanted electromagnetic radiation given off by this equipment and, at the same time, prevent the equipment from being affected by environmental [electromagnetic] radiation."

## SEPTEMBER 2000

*www.computer.org/csdl/mags/co/2000/09/index.html*

**Letter** (p. 6) "As far as I can tell, software productivity is still on the same slow-moving upward curve it has been on for the past 40 years. Overall industry productivity has been largely unaffected by previous alleged revolutions such as CASE [computer-aided software engineering], RDBMSs [relational database management systems], and Expert Systems. Each of these, and others, was at one time the latest fad, and each was highly oversold."

**3D Graphics Chips** (p. 12) "The highly competitive graphics-chip marketplace stands in stark contrast to the microprocessor marketplace, which is dominated year after year by Intel. In contrast, some graphics-chip makers that led their market one year became minor players the next because they didn't continue to innovate or at least keep up with technology trends."

**Embedding Databases** (p. 27) "To arrive at the best embedded-database-system solution, you must select the product that best matches your specific needs, then integrate that solution with your application. Start this process by evaluating which services your embedded application will provide."

**Networks at Home** (p. 35) "The home audio/video interoperability (HAVi) architecture is a set of [APIs], services, and an on-the-wire protocol specified by an industry initiative. HAVi facilitates multivendor interoperability between consumer electronics devices and computing devices and simplifies the development of distributed applications on home networks."

**Overlaying** (p. 44) "High-performance parallel computers gave researchers the ability to implement inherently parallel techniques such as cellular automata, neural networks, and genetic algorithms—significant new mathematical models for describing complex scientific phenomena. This article explains how cellular automata offer a powerful modeling approach for complex systems in which global behavior arises from the collective effect of many locally interacting, simple components."

**IEEE and Its Computer Society** (p. 64) "Recognizing the influence and control the IEEE wields over our Society and in turn the power of Society members' votes to influence the IEEE leadership, we posed four questions to this year's candidates for IEEE president-elect."

**Olympic Reform** (p. 91) "If the Olympic Movement is failing to meet its goal through its Games system, professional system designers should as a matter of professional responsibility—given the Games' global importance—consider how the Games system could be improved."

**Project Failure** (p. 94) "One of the most frequently cited software project statistics comes from the Standish Group's 1995 Chaos report … : 'A staggering 31.1 percent of [software] projects will be canceled before they ever get completed.' The Chaos report, and numerous documents citing it, label these canceled projects as 'failed' and imply that all 31.1 percent of them were canceled because of poor software management. This implication is both false and hazardous."

**The Internet** (p. 98) "The prevailing tendency to exploit computation to achieve better communication is swiftly redefining the global information ecology by making every published item instantly visible in any part of the world. However, the inverse trend—harnessing global communication to achieve more powerful computation—is also developing before our eyes."

**Consumer Casting** (p. 103) "The future lies in narrowcasting. A hundred broadcast channels aren't worth as much to a consumer as a single channel that plays exactly what that individual wants to hear. Extreme personalization of content is the next big wave. Look to companies such as BlueWireless to provide content that adapts to your personal play list—delivered on a single channel. Thanks to personalization, broadcast will be dead by the time DARS [Digital Audio Radio Satellite] reaches the masses." C

# ELSEWHERE IN THE CS

EDITOR LEE GARBER
lgarber@computer.org

## *Computer* Highlights Society Magazines

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge computing topics including scientific applications, Internet computing, machine intelligence, pervasive computing, security and privacy, digital graphics, cloud computing, and computer history. Here, we highlight recent issues of other Computer Society magazines.

### IEEE Software

**Software architecture principles** epitomize architecture's function: to clearly define a system design's necessary constraints without prescribing all design details. According to "Harnessing the Power of Architectural Design Principles," from *IEEE Software*'s July/August 2016 issue, a good set of principles can provide context and justification for design decisions and can foster team collaboration and communication.

### IEEE Internet Computing

As the Internet grows in size and complexity and plays a more important role in modern society, measuring the network becomes increasingly critical to guide its continued evolution. However, the difficulty and ethical implications of conducting Internet experiments make it challenging to obtain an accurate and representative understanding of the network's behavior. These issues are discussed in *IEEE Internet Computing*'s July/August 2016 special issue on **measuring the Internet**.

### computing in SCIENCE & ENGINEERING

The author of "**The Power to Create Chaos**," which appears in *CiSE*'s July/August 2016 issue, says computers are the only research tools that, by design, exhibit chaotic behavior. A minimal change to a computation's input can change its output in many significant ways. Scientific software developers and users should be aware of this and set up safety nets for protecting themselves against unfortunate surprises.

### IEEE SECURITY & PRIVACY

*IEEE S&P*'s July/August 2016 **security smorgasbord** special issue covers a range of topics, including changes necessary to keep pace with new technologies, biometrics as an identity-verification method in airports, the benefits and repercussions of collecting privacy-sensitive data in gaming, security implications for creating mobile software-defined networks, and new types of malware.

### IT Professional

**Smart cities** are becoming an established way to apply information and communications technology to improve urban life. Recent advances have accelerated smart-city development, and the pervasiveness of digital sensors and control systems has enabled many applications. *IT Pro*'s July/August 2016 special issue provides examples of systems that enable the optimization or redesign of existing municipal services and identify new improvement opportunities.

0018-9162/16/$33.00 © 2016 IEEE

## IEEE Intelligent Systems

*IEEE Intelligent Systems*' July/August 2016 special issue on **new directions** includes articles on semantics-based intelligent human–computer interaction, minimizing fatigue damage in aircraft structures, satisfiability degree analysis and deductive reasoning, and transfer learning for heterogeneous one-class collaborative filtering.

## IEEE CLOUD COMPUTING

Cloud computing continues to grow in complexity. Autonomic computing helps deal with this complexity by enabling the self-management of systems and applications, as addressed in *IEEE Cloud Computing*'s May/June 2016 special issue on **autonomic clouds**.

## IEEE Computer Graphics AND APPLICATIONS

Perceptual computer graphics is a rich research area, mainly because it focuses on the human element. *IEEE CG&A*'s July/August 2016 special issue on **quality assessment and perception in computer graphics** includes four articles that explore innovative techniques in areas such as computational aesthetics in games.

## IEEE MultiMedia

The volume of multimedia data handled daily is growing rapidly. **Computational quality modeling** is thus becoming an important, yet challenging, research topic. Artificial intelligence has proven effective in dealing with this preponderance of data, and in the past decade, researchers have proposed a rich variety of quality models. The articles in *IEEE MultiMedia*'s July–September 2016 special issue cover different computational-quality-modeling techniques and applications.

## IEEE Annals of the History of Computing

Introduced with the 1972 extension of Japan's Tōkaidō Shinkansen high-speed rail line, the **COMTRAC (computer-aided traffic control) command-and-control system** efficiently runs the train's complex, high-frequency operations. "History of COMTRAC: Development of the Innovative Traffic-Control System for Shinkansen," from *IEEE Annals*' April–June 2016 issue, introduces the system's technological features and reviews its 50-year history.

## IEEE pervasive COMPUTING

The key challenge for designers of future displays will be creating innovative systems that deliver value. The research presented in *IEEE Pervasive Computing*'s July–September 2016 special issue showcases some of the challenges—and potential solutions—in making **pervasive displays** more useful and user friendly. As new display and sensing technologies become available, pervasive displays could emerge as one of the core components of a future ubiquitous computing infrastructure.

## IEEE micro

**Nonvolatile processors** (NVPs) integrate nonvolatile memory to preserve on-chip state during power emergencies. NVPs can work on tasks even when there are only short periods of power, making them a promising solution for energy-harvesting scenarios in which the available power supply is unstable and intermittent. The authors of "Nonvolatile Processor Architectures: Efficient, Reliable Progress with Unstable Power," from *IEEE Micro*'s May/June 2016 issue, explore NVP design across different architectures and power sources, and propose an efficient, heterogeneous microarchitecture. C

Selected CS articles and columns are also available for free at **http://ComputingNow.computer.org.**

SPOTLIGHT ON TRANSACTIONS

# Modern Computer Arithmetic

**Paolo Montuschi,** Polytechnic University of Turin
**Jean-Michel Muller,** Centre National de la Recherche Scientifique

*This installment of Computer's series highlighting the work published in IEEE Computer Society journals comes from* IEEE Transactions on Computers.

A 2009 *IEEE Transactions on Computers* (*TC*) guest editorial called computer arithmetic "the mother of all computer research and application topics." Today, one might question what computer arithmetic still offers in terms of advancing scientific research; after all, multiplication and addition haven't changed. The answer is surprisingly easy: new architectures, processors, problems, application domains, and so forth all require computations and are open to new challenges for computer arithmetic. Big data crunching, exascale computing, low-power constraints, and decimal precision are just a few domains in which advances are implicitly pushing for rapid, deep reshaping of the traditional computer-arithmetic framework. *TC* (www.computer.org/web/tc) has long published regular submissions as well as special sections on this topic, including one scheduled for 2017. Here, we focus on three recently published papers.

In "Parallel Reproducible Summation," James Demmel and Hong Diep Nguyen (*IEEE Trans. Computers*, vol. 64, no. 7, 2015, pp. 2060–2070) address result reproducibility in cases where it's a requirement. They present a technique for floating-point reproducible addition that doesn't depend on the order in which operations are performed, which makes it appropriate for massively parallel environments.

Mioara Joldeş and her colleagues deal with manipulation of floating-point expansions in "Arithmetic Algorithms for Extended Precision Using Floating-Point Expansions" (*IEEE Trans. Computers*, vol. 65, no. 4, 2016, pp. 1197–1210). Such expansions, which are unevaluated sums of a few floating-point numbers, might be used when one temporarily needs to represent numerical values with a higher precision than that offered by the available floating-point format. The authors introduce and prove new algorithms for dividing and square-rooting floating-point expansions, as well as for "normalizing" such expansions.

In "On the Design of Approximate Restoring Dividers for Error-Tolerant Applications" (*IEEE Trans. Computers*, vol. 65, no. 8, 2016, pp. 2522–2533), Linbin Chen and his colleagues propose several approximate restoring-divider designs. Their simulation results show that, compared with nonrestoring division schemes, their designs had superior delay, power dissipation, circuit complexity, and error tolerance. Most striking, the approximate designs offer better error tolerance "for quotient-oriented applications (image processing) than remainder-oriented applications (modulo operations)."

These papers are a small but representative view of trends in computer arithmetic. However, computer arithmetic also bridges the gap between architecture and application design—and thus will continue to advance in vibrant directions, provided it maintains strong connections with technology and advanced research. C

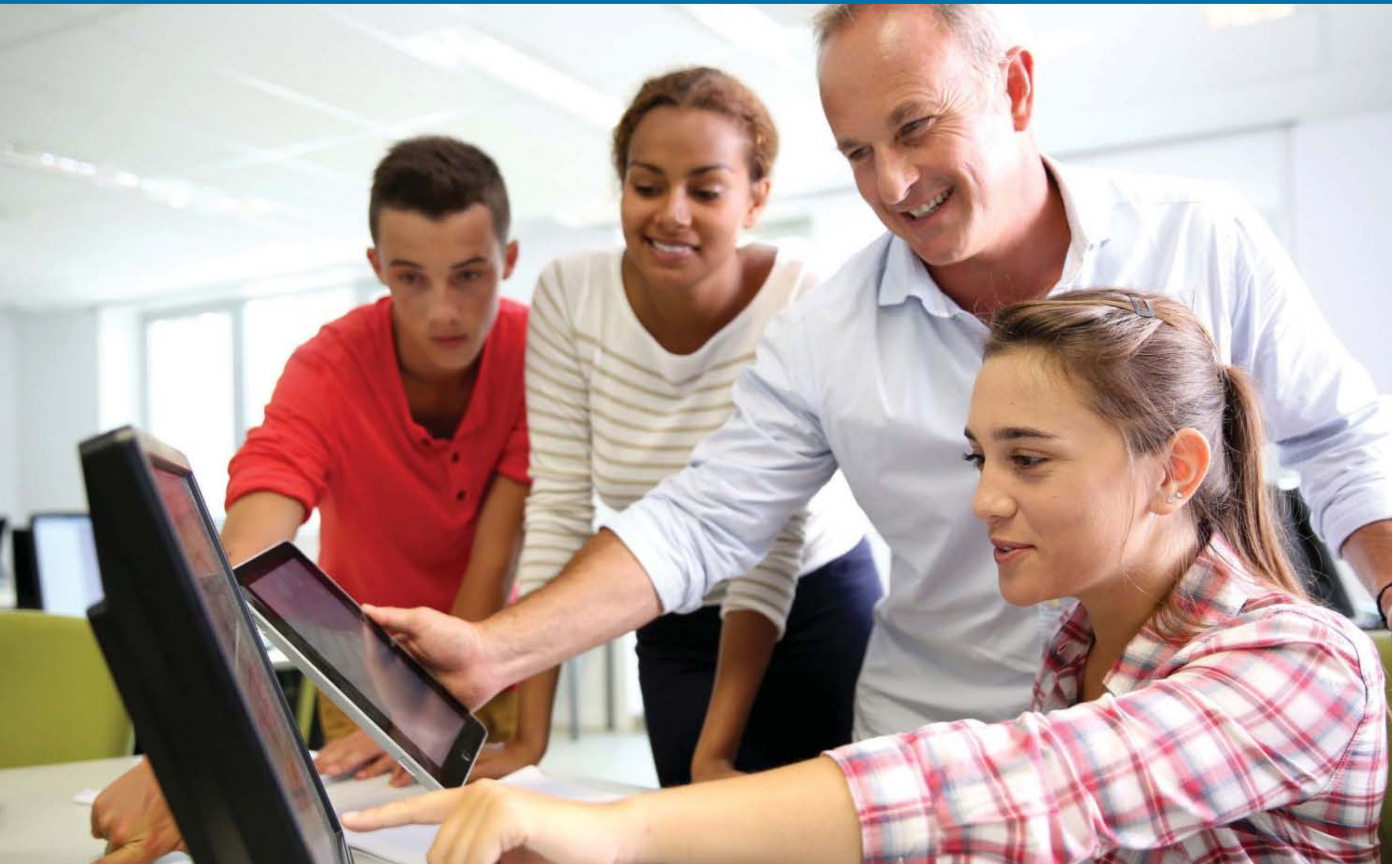
**PAOLO MONTUSCHI** is a professor of computer engineering at the Polytechnic University of Turin. Contact him at pmo@computer.org or visit http://staff.polito.it/paolo.montuschi/news-from-EIC-TC.html.

**JEAN-MICHEL MULLER** is a Senior Researcher at Centre National de la Recherche Scientifique (National Center for Scientific Research), École Normale Supérieure, Lyon. Contact him at jean-michel.muller@ens-lyon.fr or visit http://perso.ens-lyon.fr/jean-michel.muller.

0018-9162/16/$33.00 © 2016 IEEE

# CALL FOR NOMINEES

## Education Awards Nominations



## Taylor L. Booth Education Award

**A bronze medal and US$5,000 honorarium** are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

**Two endorsements** are required for an award nomination.

See the award information at:
www.computer.org/web/awards/booth

## Computer Science and Engineering Undergraduate Teaching Award

**A plaque, certificate and a stipend of US$2,000** is awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and making a statement about the importance with which we view undergraduate education.

The award nomination requires a **minimum of three endorsements**.

See the award details at:
www.computer.org/web/awards/cse-undergrad-teaching

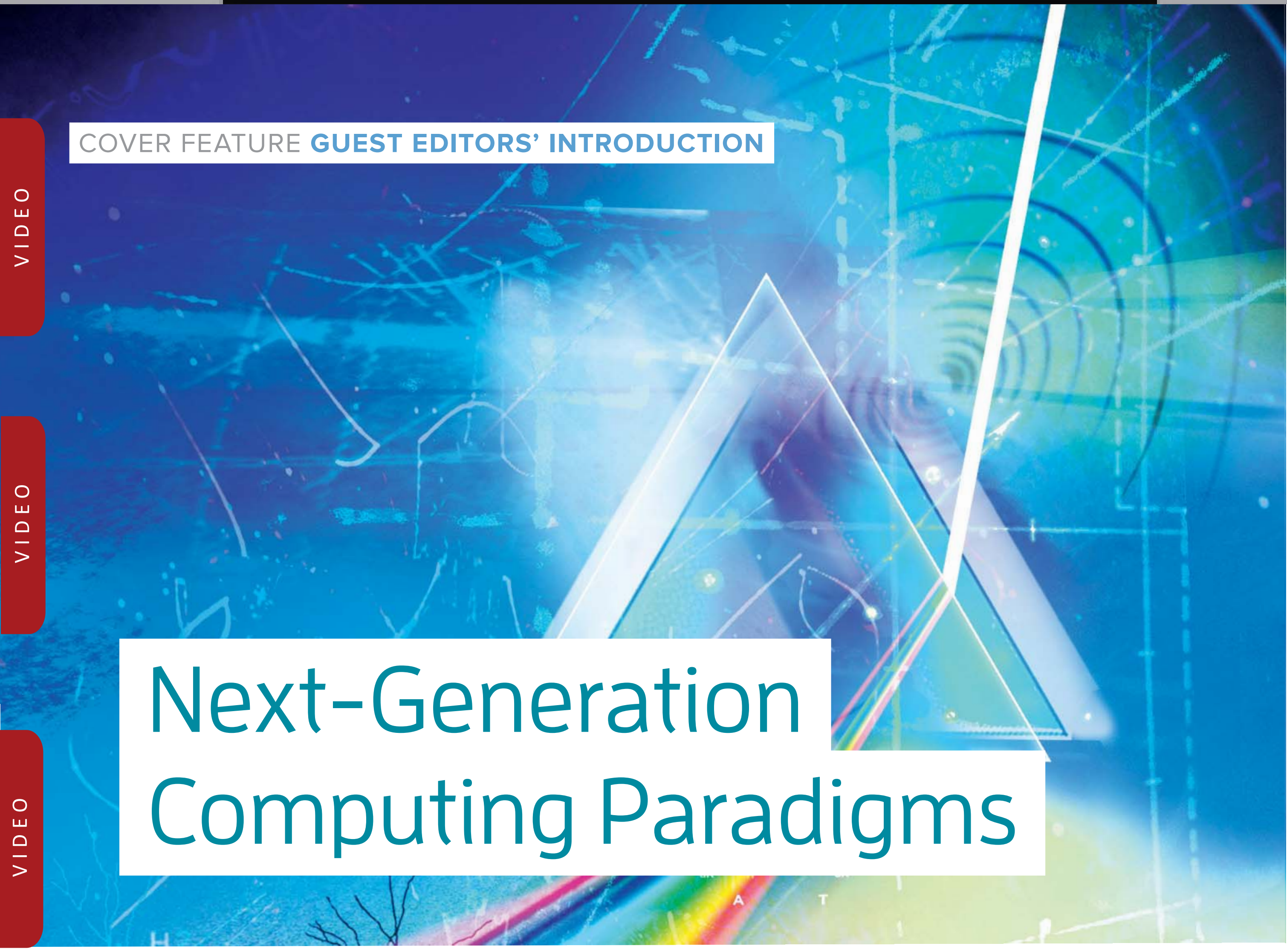
**Deadline:** 15 October 2016
**Nomination Site:** awards.computer.org

IEEE

IEEE computer society

VIDEO
VIDEO
VIDEO

COVER FEATURE **GUEST EDITORS' INTRODUCTION**



# Next-Generation Computing Paradigms

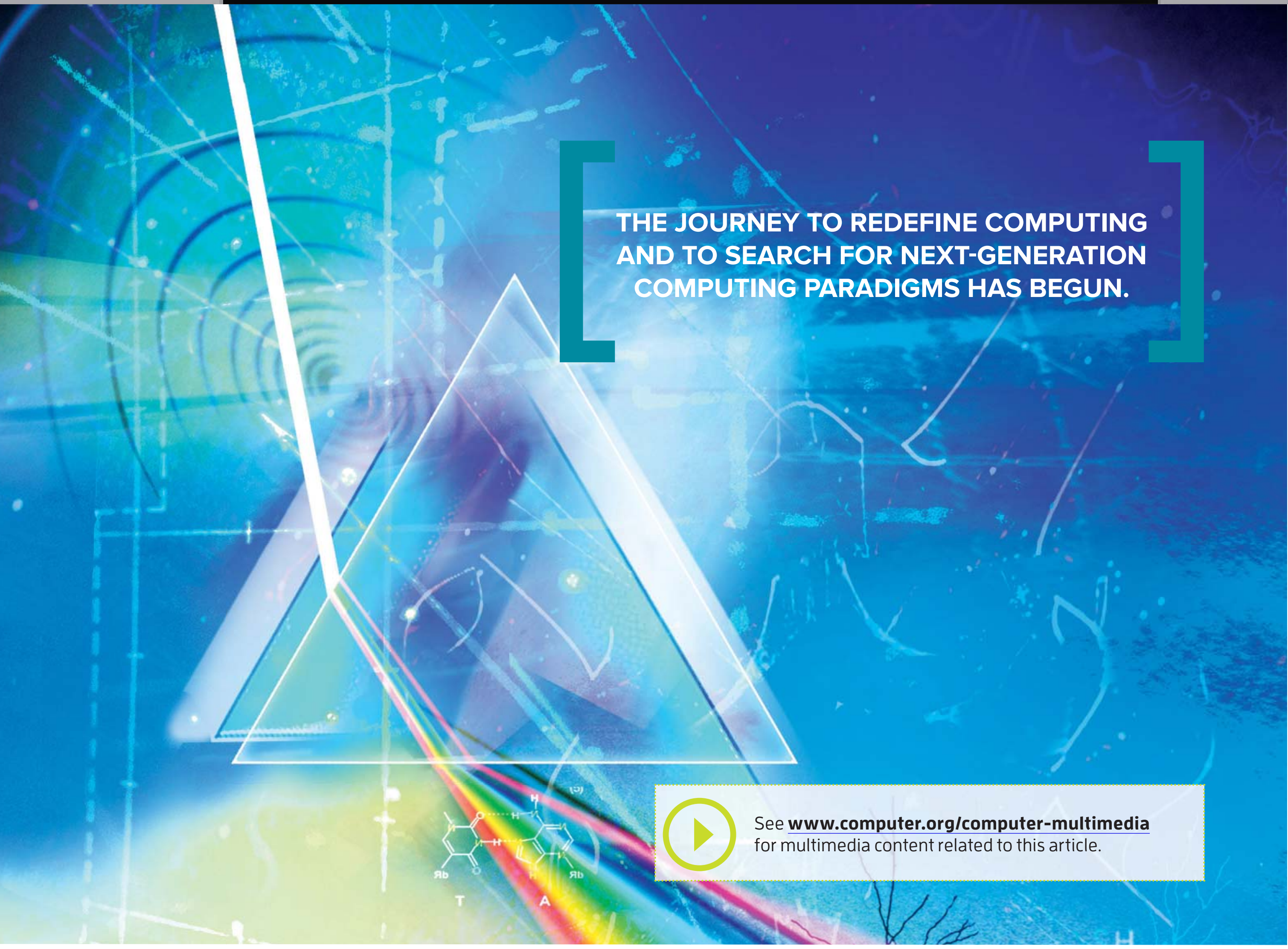
**San Murugesan,** BRITE Professional Services and Western Sydney University
**Bob Colwell,** R&E Colwell & Assocates

*Faced with challenging new applications for computing, we must pursue radical new paradigms. Through quantum computing, biologically inspired computing, and nanocomputing, we can explore novel ways to transform life for the benefit of society.*

As computers have evolved to redefine and transform almost every area of our lives in the past 50 years, they still function on the same fundamental computational concepts envisaged by Alan Turing and John von Neumann at the very beginning. As demands on computing, storage, and communication continue to escalate, digital computers based on silicon and conventional architecture approach their fundamental physical limits and face issues related to economics and reliability. Thus, certain kinds of problems in domains such as weather forecasting, bioinformatics, robotics, and autonomous systems are faced with limitations tied to the conventional computing paradigm.

Do these fundamental principles and assumptions that have shaped current conventional computing require revolutionary rethinking? Do we need to explore and harness new computing paradigms to address unresolved and as yet unforeseen challenges? The answer of course is "yes," and the journey to redefine computing and to search for next-generation computing paradigms has begun.[1–4]

0018-9162/16/$33.00 © 2016 IEEE



THE JOURNEY TO REDEFINE COMPUTING AND TO SEARCH FOR NEXT-GENERATION COMPUTING PARADIGMS HAS BEGUN.

See www.computer.org/computer-multimedia for multimedia content related to this article.

Research and industry are exploring radical new computing paradigms[3] such as quantum computing, biologically inspired computing, nanocomputing, and optical computing—all of which have the potential to bring about a variety of challenging new applications. Understanding, mastering, and applying these kinds of emerging, innovative approaches will empower us to chart the future course of computing. This special issue explores the principles of and potential for some of these paradigms and examines their current status and future prospects. We hope to inspire further study and implementation along these directions.

## IN THIS ISSUE

The five feature articles in this issue explore quantum computing, molecular computing, nature-inspired algorithms, and synergistic human–machine interaction through cortically coupled computing. These approaches help us in our quest to address current and future computing challenges through innovation. In addition, two experts offer their insights on next-generation computing and how quantum computing will impact information security (see the "Perspective: Next-Generation Computing Paradigms and the Information Revolution" and "Perspective: How Quantum Technology Will Impact Security" sidebars). Furthermore, to help readers quickly gain a better understanding of some of these new paradigms, we put together a video album to accompany this issue (www.computer.org/web/computer-multimedia).

### Quantum Computing Advances

Computational problems that are out of reach of current classical computers can in some cases be solved through devices that use the quantum mechanical properties of superposition and entanglement. This approach enables us to design devices with capabilities that exceed those of any classical computer. Recently, quantum devices and quantum techniques have attracted significant interest from researchers and industry. Quantum technologies for creating random numbers and securely encrypting communication are in fact already commercially available.

In "The Quantum Future of Computation," Krysta M. Svore and Matthias Troyer describe the principles of quantum bits, gates, and algorithms. The authors also outline the use of a quantum computer as a special-purpose coprocessor; highlight the use of quantum algorithms in a range of applications, such as cryptography, privacy, and search; and propose a software stack for quantum computing.

In "The Path to Scalable Distributed Quantum Computing," Rodney Van Meter and Simon J. Devitt present architectural models for large-scale quantum computation. They describe the classical resources needed to operate a large-scale quantum computer and explore experimental progress in a variety of different systems that support construction of a scalable quantum computer.

## Molecular Computing

The quest for radical new algorithms and physical implementation to solve computational challenges better, cheaper, and faster than conventional computers has led to some research in molecular computing. This methodology has the potential to transform conventional computation by addressing such things as information density, parallelism, and energy efficiency.

In "Embodied Molecular Computation: Potential and Challenges," Victoria Coleman describes a type of computer in which living cells can be "programmed" by biological modification to perform computational tasks. In embodied molecular computing, computation is carried out via biological systems including the use of cellular materials such as DNA molecules. This article not only describes embodied molecular computing principles and potential, but also outlines challenges associated with building and using a universal molecular computer.

## Inspiration for Computing from Nature

Nature inspires all kinds of ingenious problem-solving and optimization strategies. In fact, nature-inspired algorithms are particularly well suited

### PERSPECTIVE: NEXT-GENERATION COMPUTING PARADIGMS AND THE INFORMATION REVOLUTION

Erik DeBenedictis, Sandia National Laboratories

New computing paradigms could drive the information revolution to completion. Society had defined "computing" based on the contributions of Alan Turing, John von Neumann, and Gordon Moore. Turing showed how to describe the solution to any computable problem in what is essentially a C program. Von Neumann architected a computer for running the program, and Moore described how semiconductor scaling would make the computers grow exponentially more capable over time.

Belief in Moore's law suppressed work on alternative paradigms. If semiconductor improvements would speed up the solution of any computable problem exponentially, what more could we want? I recall people seeking funding for a new computer architecture years ago, claiming it would be ten times as efficient as a microprocessor. The counter argument was "let's do nothing for four years and then buy a regular computer, which will be ten times faster due to Moore's law."

Nevertheless, the traditional computing paradigm has several major limitations that even Moore's law does not address:

- While a computer may require infinite memory for some problems, real computers have only finite memory.
- A computer will not do anything at all until a human programs it.
- A computer may run forever and still not solve some problems, like factoring a large number.
- Gordon Moore only projected exponential growth through 1975.

So, it is now appropriate to shift our attention to addressing these limitations by other means, such as biological computing, human–computer teaming, and quantum computing.

#### BIOLOGICAL COMPUTING

Some biologically inspired computing approaches have a remedy to the problem of a computer running out of memory. Some algorithms really do need a lot of memory, but a von Neumann computer is unable to increase its own memory because it does not have the ability to fabricate memory by itself. However, a biological cell can be modified to perform computing without necessarily shutting off the cell's reproductive capability. This makes a cell equivalent to both a computer and a memory fabrication facility. A cluster of cells that is too small for a particular computation can grow bigger without human help.

## HUMAN–COMPUTER TEAMING

Computing today is the result of teamwork between the programmer and the hardware, but the nature of the teaming can change. In recent deep learning breakthroughs, humans architected a program's structure and the computer did a very large amount of simple programming within that structure.[1] Also, advances in human–computer interfaces enable a new type of team at runtime. This could lead to future human–computer partnerships with the computational throughput of a computer and the problem-solving ability (programming), motivation, and intuition of humans.

## QUANTUM COMPUTATION

Some problems demand really long run times when run on traditional computers. According to computational complexity theory, a program to solve a problem of size $N$, such as having $N$-bits of input data, is "tractable" only if the number of steps in a solution is polynomial in $N$ or less. If the number of steps is larger, such as exponential in $N$, even the exponential scaling of Moore's law could be insufficient.

Quantum computers address this limitation. As an example, consider factoring the number $N$. The best nonquantum algorithm for factorization is the number field sieve, where the number of steps is on the order of exp($1.52$ (log $N$)$^{1/3}$ (log log $N$)$^{2/3}$) operations. The expression is complicated, but the initial exponential function indicates it is of greater than polynomial order. Factoring is thus called "intractable." However, the running time of the best quantum algorithm is only of the order of (log $N$)$^2$ (log log $N$) (log log log $N$) quantum operations, which is within polynomial range. For values of $N$ typical in cryptanalysis, the first expression represents elapsed time greater than the age of the universe, whereas the second one is reasonable.

## LOOKING AHEAD

There is apparent interest in continuing the information revolution and resulting economic expansion. The initial technological driver was the implicit extension of Moore's projection of exponential growth from ten years to forever. Although the original projection has reached it limits, new models of computers and computation, including those in this special issue, could be realistic and practical alternatives for driving the information revolution further and in ways unimaginable so far.

### Reference

1. E.P. DeBenedictis, "Rebooting Computers as Learning Machines," *Computer*, vol. 49, no. 6, 2016, pp. 84–87.

**ERIK P. DEBENEDICTIS** is a technical staff member in the Non-Conventional Computing Technologies Department at Sandia National Laboratories. Contact him at epdeben@sandia.gov.

for a certain class of applications—optimization, machine learning, and multi-objective and highly complex design problems.

In "From Swarm Intelligence to Metaheuristics: Nature-Inspired Optimization Algorithms," Xin-She Yang, Suash Deb, Simon Fong, Xingshi He, and Yu-Xin Zhao describe recent developments in nature-derived algorithms and give an overview of those derived from species-based behaviors. To solve a diverse range of real-world application-based problems, they urge continuing research in a few specific areas to advance this area further.

### Synergetic Human–Machine Interaction and Teamwork

How humans and machines interact and collaborate is poised for radical improvement. In "Cortically Coupled Computing: A New Paradigm for Synergistic Human–Machine Interaction," Sameer Saproo, Josef Faller, Victor Shih, Paul Sajda, Nicholas R. Waytowich, Addison Bohannon, Vernon J. Lawhern, Brent J. Lance, and David Jangraw postulate that as machine intelligence approaches the general effectiveness of human intelligence, the need for explicit programming of machines by humans will be disrupted. Through examples of real systems, the authors explain the concept of cortically coupled computing—that is, both human and machine are actively involved in performing computational tasks in which communication is enabled through brain–computer interfaces (BCIs). Such systems use brain-derived information

# PERSPECTIVE: HOW QUANTUM TECHNOLOGY WILL IMPACT SECURITY

**Jane Melia,** QuintessenceLabs

The conversation around quantum technologies tends to focus for a large part on quantum computers and their capabilities, as well as on the threat they pose to our current cybersecurity infrastructure. What is less well known is that quantum technologies also present a security solution—they hold tremendous promise for protecting the most sensitive data.

## THE THREAT: QUANTUM COMPUTERS' SECURITY CHALLENGE

Quantum computers are touted as the next computing revolution. By relying on the principles of superposition and entanglement, some purely quantum mechanical phenomena, they could solve some previously intractable problems. At present they are known to be able to solve certain specific categories of problems (such as factorization).As research continues, additional quantum-appropriate algorithms might well be discovered. This can have many positive ramifications, for example in medical research.

However, quantum computers also challenge our security infrastructure's status quo. Current strategies for sharing encryption keys rely in part on the difficulty in factoring a large multiplication back into its prime constituents, a problem that is beyond the reach of classic computers in a reasonable timeframe. Once quantum computers mature, they will be able to crack this mathematical challenge quickly, rendering the process of sharing keys through public-key infrastructure insecure.

Symmetric encryption is itself expected to remain safe, as long as the key length is increased (doubled) and fully random. This is because a quantum-based search using Grover's algorithm is only expected to have a quadratic speedup, and an exponential speedup for search algorithms has been shown to be impossible. Unfortunately, in a post-quantum world in which public key sharing is insecure, this quantum-resistance of symmetric encryption becomes irrelevant unless we find a way to securely exchange them.

The US National Institute of Standards and Technology (NIST) estimates that mature quantum computers will be able to crack our public-key infrastructure within 15 years.[1] This may seem far out, but we are in fact in a race for time: upgrading infrastructure takes years, and a lot of sensitive data needs to be kept secure for long periods of time, making it vulnerable to being captured and stored for later decryption as quantum computers become available. Any organization that handles personal or financial information with a long shelf life needs to get ready as soon as possible.

## NOW, FOR THE PLUS SIDE

Quantum technology also delivers capabilities that can be used to enhance data security—both from today's attacks, and those from future quantum computers. This is typically known as quantum cybersecurity.

Aside from the quantum computing–related threat, poor quality or insufficient quantity of random numbers also present a security risk. Generating high-quality random numbers at high rates has proven a surprisingly hard problem to solve. Fortunately, quantum technology provides an elegant and powerful solution.

Many processes in quantum physics are random, and this inherent randomness has been harnessed into commercial quantum random-number generators capable of producing fully random numbers at high rates and cost-effectively, putting this issue to rest. These devices are starting to be integrated into cloud security infrastructure, in finance and beyond—a trend that is expected to increase over the coming years. As a bonus, the use of longer, higher-quality keys was identified by the National Security Agency (NSA) as a strategy for protecting data from the threat of quantum computers,[2] so using a high-quality quantum random bit generator enables security-aware companies to get a head start in that direction. True random bits are also necessary prerequisites to using one-time pad (OTP) encryption. This is a type of encryption for which the encrypted text provides no information about the clean text, so that it is safe, independent of the processing power of the attackers, including from quantum attacks.

At a more advanced level, quantum key distribution (QKD) uses the laws of quantum mechanics to enable private and secrete key sharing between two parties, even if they have no control over their communication link. It therefore solves the thorny key-exchange problem mentioned before. Its security is based on a fundamental characteristic of quantum mechanics: that is, the process of measuring a quantum system disturbs the system. An attacker trying to intercept the key exchange will inevitably leave detectable traces, allowing that information to be discarded. QKD has proved to be informationally secure, meaning it remains safe independent of the processing power of the attackers, and is not vulnerable to quantum computers.[3] This developing technology has challenges to overcome, but corporations are beginning to roll out commercial implementations, and there is development under way to move beyond point-to-point capability and emancipate this from its current fiber-optic constraints to free space and ultimately mobile devices. It is certainly worth watching.

## SO, WHAT WILL THE FUTURE LOOK LIKE?

In addition to these technology-driven solutions, a search is also underway for algorithms believed to be secure from both classical and quantum-computing attacks. These quantum-resistant algorithms will have challenges: they can't serve as a drop-in replacement for current solutions (thus they will require changes in current protocols), and they may be vulnerable to new attacks or advances in mathematical knowledge as they emerge. It will also take many years to reach standardization around new algorithms. However, they will provide flexibility, and an important element to an overall quantum safe security approach.

In the race to protect our data from the power of quantum computers, it is likely that hybrid solutions will emerge. Keys will be stronger, with what we call "full entropy" or true randomness. Crucial links will be protected using a global, flexible QKD network, invulnerable to quantum computers. Finally, for shorter, less-exposed links, improved algorithms could provide enhanced protection, regularly updated against growing threats.

Whereas the quantum computer is certainly a major threat to cybersecurity, approaches such as quantum random-number generators, QKD, and quantum-resistant algorithms are ramping up to take on this challenge, allowing us to reap the benefits of that technology while remaining secure.

## FINDING OUT MORE

There is currently a lot of interest, activity, and development in quantum-safe security—from enterprises and government institutions seeking to protect confidential information, standardization bodies looking to structure new safer ways of communicating, and companies and research institutions developing solutions to these challenges. If you are interested in finding out more, I recommend connecting with the Quantum Safe Security Working Group (QSS-WG), which was formed within the Cloud Security Alliance at the end of 2014. QSS-WG is a forum for interested corporations, organizations, and individuals; its mission is to stimulate the understanding, adoption, use, and widespread application of quantum-safe cryptography to commercial institutions, policy makers, and all relevant government bodies.

**JANE MELIA** is the vice president of strategic business development at QuintessenceLabs and co-chair of the CSA Quantum Safe Security Working Group. Contact her at jm@quintessencelabs.com.

### References

1. D. Moody, "Post-Quantum Cryptography: NIST's Plan for the Future," report, Nat'l Inst. Standards and Technology (NIST); http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/pqcrypto-2016-presentation.pdf.
2. K. Kennedy, "NSA Recommendations Include High Entropy and Longer Keys to Protect Against Quantum Computer Developments," *CTOVision.com*, 12 Oct. 2015; https://ctovision.com/2015/10/nsa-recommendations-include-high-entropy-longer-keys-protect-quantum-computer-developments.
3. C.H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," *Proc. IEEE Int'l Conf. Computers, Systems, and Signal Processing*, 1984, pp. 175–179; www.cs.ucsb.edu/~chong/290N-W06/BB84.pdf.

## ABOUT THE AUTHORS

**SAN MURUGESAN** is director of BRITE Professional Services and an adjunct professor at the Western Sydney University. He is the editor in chief of *IT Professional*; an editorial board member of *Computer* and *IEEE Transactions of Cloud Computing*; a co-editor of the *Encyclopedia of Cloud Computing* (Wiley-IEEE, 2016) and *Harnessing Green IT: Principles and Practices* (Wiley-IEEE 2102); and the editor of *Computer*'s Cloud Cover column. Murugesan is a Fellow of the Australian Computer Society and the Institution of Electronics and Telecommunication Engineers (IETE). Contact him at san@computer.org or via http://bitly.com/sanprofile.

**BOB COLWELL** is an independent consultant at R&E Colwell & Associates, and he served as director of DARPA's Microsystems Technology Office from 2012–2014. Previously, Colwell was Intel's chief IA32 (Pentium) microprocessor architect from 1992–2000. He is the Perspectives editor for *Computer*, an author of the "At Random" column from 2002–2005, and an author of *The Pentium Chronicles*. He is a Fellow of IEEE. Contact him at bob.colwell@gmail.com.

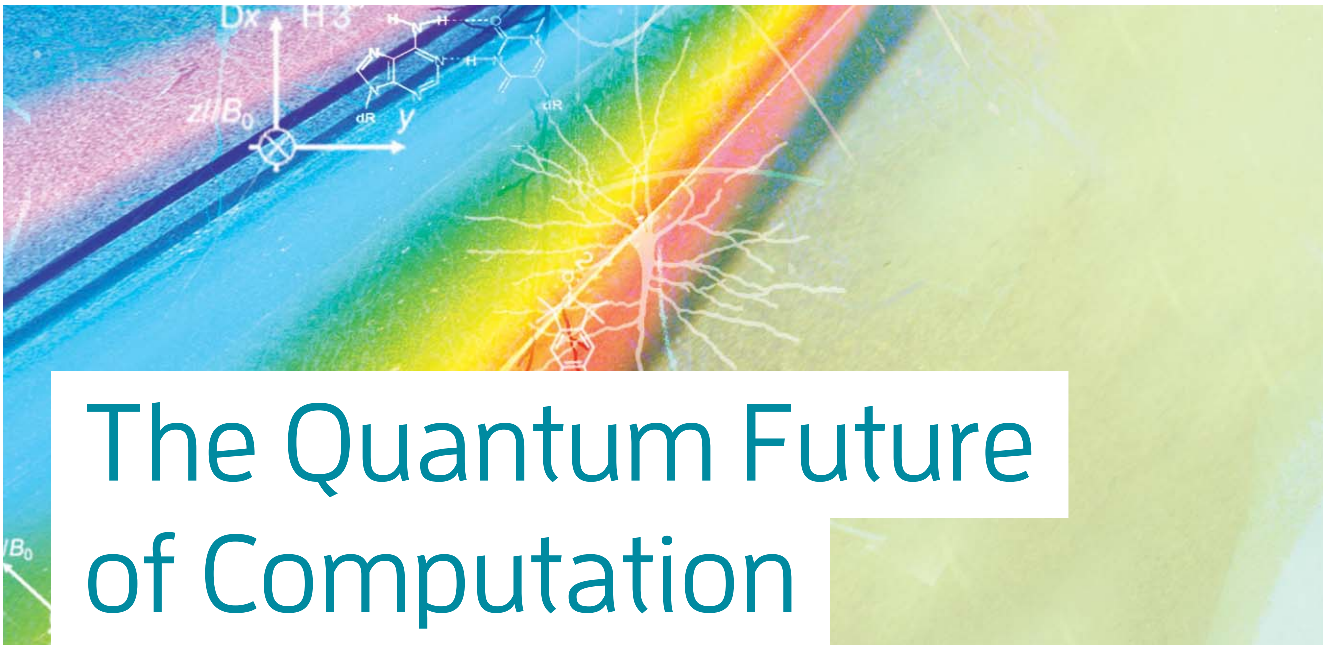
to "teach" a machine that has not been programmed a priori, thus giving rise to future possibilities in which smart computers (that have advanced artificial intelligence) and humans team up to cooperatively execute tasks and enhance human–machine synergy. Cortically coupled computing in which human–machine interaction is synergistic can be more computationally powerful than the sum of the parts.

Computing paradigms will continue to emerge and evolve to offer new capabilities that extend computing's reach and utility. To successfully embrace the potential offered by new computing paradigms, researchers, developers, and industry have to address several questions: How can we effectively address the challenges these paradigms pose? Will such paradigms be viable and evolve as next-gen computers? Are they transformational?

Through the articles in this special issue, we give you a glimpse of what is on the horizon for emerging computing technologies, and we encourage researchers and developers from multidisciplinary fields to learn from each other and work together to further advance computing. ◼

## REFERENCES

1. "Rebooting Computing,"special issue, *Computer*, vol. 48, no. 12, 2015; www.computer.org/csdl/mags/co/2015/12/index.html.
2. "Rebooting Computing Initiative," website, IEEE; http://rebootingcomputing.ieee.org.
3. S. Murugesan, "Radical Next-Gen Computing," *Computing Now*, vol. 8, no. 6, June 2015; www.computer.org/web/computingnow/archive/radical-next-gen-computing-june-2015.
4. W. Mazurczyk and E. Rzeszutko, "Security–A Perpetual War: Lessons from Nature," *IT Professional*, vol.17, no. 1, 2015, pp. 16-22; www.computer.org/cms/Computer.org/ComputingNow/issues/2015/06/T-mit2015010016.pdf.



# The Quantum Future of Computation

**Krysta M. Svore,** Microsoft Research

**Matthias Troyer,** ETH Zurich and Microsoft Research

*Still in early development, quantum computing is already overturning our contemporary notions of computational methods and devices. Understanding the applications enabled by quantum computing—and how to harness them—will alter the economic, industrial, academic, and societal landscape.*

More than a century after the discovery of quantum mechanics, we are on the threshold of an age in which quantum properties not only enable classical devices, such as the transistor, but also allow these devices to evolve beyond their present-day capabilities. Unlike classical computers, quantum devices can simultaneously be in a "superposition" of many different states and have deep connections ("entanglement") between spatially separated entities. These properties make it possible to design devices whose capabilities exceed that of any imaginable classical computer. In fact, devices to create quantum random numbers and securely encrypt communication are already commercially available.

The arguably simplest quantum device is a *quantum random number generator,* which uses the property of superposition: a quantum bit (the basic unit of information in a quantum computer, also called a qubit) can be created in a state $1/\sqrt{2}\,(|0\rangle + |1\rangle)$ that is the superposition of the values 0 and 1 of a classical bit (see the "Quantum Bits and Gates" sidebar). Upon measuring the value of the qubit, the superposition collapses into one of the two classical states, and the measurement gives either the value 0 or 1, each with probability ½. This procedure implements a true random number generator, which is impossible classically due to the deterministic nature of traditional physics.

Entangling two qubits A and B in a superposition denoted $1/\sqrt{2}\,(|0\rangle_A\,|0\rangle_B + |1\rangle_A\,|1\rangle_B)$, such that they are either both 0 or both 1, can be used for cryptography. Consider that two parties, Alice and Bob, want to exchange a secret message. Alice is assigned one of the qubits (the qubit with subscript A), and Bob is given the other (the qubit with subscript B). No matter the distance separating Alice and Bob (and their respective qubits), the two qubits can remain entangled. When Alice and Bob measure their respective qubits, they will each read a random measurement result—however, both of them obtain the

0018-9162/16/$33.00 © 2016 IEEE

same random value. This procedure can be used for quantum teleportation (to transport quantum information) or to establish a shared secret key for provably secure communication, again surpassing classical technology.

Faced with the challenge of simulating quantum systems on classical computers, in 1982 Richard Feynman suggested building computers that use quantum effects to compute, generalizing the bits and gates of classical computers to qubits and quantum gates.[1] It turns out that so-called quantum computers have applications well beyond solving physics problems, and we already have a diverse range of quantum algorithms that outperform their classical counterparts (see the "Quantum Algorithms" sidebar).

Quantum computers are most readily known for solving integer factorization,[2] where given an integer $N = p \times q$, the task is to determine the prime numbers $p$ and $q$. This problem forms a mainstay of e-commerce today, as it underlies RSA, the most widely used public-key cryptosystem. A quantum computer consisting of roughly one billion physical qubits[3] would be able to break a 2,000-bit key RSA in a short time. While quantum algorithms for cryptographic attacks are certainly an area of quantum advantage, their significance is sure to decrease, as classical "post-quantum" cryptosystems aimed at resisting quantum attacks are under development.[4] Companies and governments around the world are already beginning to actively transition from RSA and other public-key schemes to post-quantum cryptosystems because the necessary software upgrades to quantum-secure methods will require at least a decade to complete.

More globally and economically impactful applications arise from the simulation of physical systems, which was the original motivation behind Feynman's seminal paper. Quantum simulations of molecules and materials currently account for a major fraction of supercomputer usage. Quantum computers will improve the reliability of such calculations by performing accurate simulations of molecules and materials on the scale of days to months for problems that would take billions of years on today's state-of-the-art supercomputers. These simulations can be applied to help solve critical real-world problems, such as elucidating the mechanism of biological nitrogen fixation for fertilizer production[5] or designing catalysts for carbon capture and sequestration to combat global warming. Extensions of these approaches promise solutions for applications in materials modeling, including discovery of materials that have higher temperature superconductivity, which could enable, for example, lossless power transmission and maglev trains (extremely high-speed trains that use magnetic levitation, where the train moves without touching the ground).

The past decade has witnessed an explosion in classical machine learning methods for making predictions and gaining insights based on data. Machine learning has utterly transformed domains such as speech recognition, computer vision, and Web search. With its foundations in computational statistics, machine learning is an area ripe for quantum improvements.[6,7] Recent results indicate that quantum algorithms might perform classification and clustering methods such as *k*-means and nearest-neighbor more efficiently than classical methods. Boltzmann machine training, deep learning, and neural networks have also been shown to benefit from quantum computing.

By and large, quantum algorithms presently require rigorous proofs to validate any improvements in runtime complexity over classical algorithms. However, the best-performing algorithms on real-world data are often heuristic in nature. Therefore, we might look forward to having a modest-size quantum computer on which to test and master the art of quantum programming and algorithm design. The fruits of such labor will result in quantum algorithms for many real-world applications, as a generation of programmers look to quantum hardware for novel improvements in speed and accuracy.

## QUANTUM COMPUTERS AS SPECIAL-PURPOSE ACCELERATORS

Quantum computers are unlike their classical cousins. Although in principle they can perform any computation that a classical computer can perform, we should not expect quantum computers to power future personal computers or phones. Classical computers will remain cheaper, smaller, and more portable than quantum computers for many of these tasks. Quantum computers show their strength when running special quantum algorithms that solve certain computational tasks faster (in some cases exponentially faster) than any classical computer by acting on a superposition of values.

A quantum computer will thus operate as a coprocessor, receiving its instructions and cues from a stack of classical processors. To suppress noise and errors and keep the qubits in a superposition, most designs for quantum computers require the qubits

# QUANTUM BITS AND GATES

While a classical bit $x$ can take only one of the two values $x = 0$ or $x = 1$, a quantum bit $q$ can be in a superposition

$$|q\rangle = \alpha\,|0\rangle + \beta\,|1\rangle, \qquad (1)$$

where $\alpha$ and $\beta$ are two complex numbers, normalized such that $|\alpha|^2 + |\beta|^2 = 1$. Identifying the two classical states 0 and 1 with two-dimensional complex-valued vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad (2)$$

the state of the qubit can be described by the vector

$$|\,q\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \qquad (3)$$

which is also called the wave function of the qubit. Generalizing to a qubit register $|x\rangle = |x_0 \ldots x_{N-1}\rangle$ consisting of $N$ individual qubits, a $2^N$-dimensional complex vector of coefficients $c_i$ is needed to describe all possible superpositions

$$|\,x\rangle = \sum_{i=0}^{2^{N-1}} c_i\,|\,\mathrm{i}\rangle, \qquad (4)$$

where the $k$th bit of the integer $i$ corresponds to the values of the $k$th qubit.

The exponential memory required to represent the state of just $N$ qubits hints at the potential for exponentially larger computational power of quantum computers compared to classical ones. Current high-end supercomputers are barely able to represent the state of $N \approx 50$ qubits. Small-scale quantum computers with about 100 qubits already offer computational capabilities that go beyond that of any classical computer for certain applications.

Despite requiring $2^N$ complex numbers to represent a quantum state on a classical computer, when reading out a quantum register one obtains just $N$ bits of information. Of all the $2^N$ classical values in the superposition, one value $i$ is chosen at random in a measurement, according to a probability distribution given by the squares of the wave-function entries $|c_i|^2$.

**TABLE A.** Standard quantum gates.

| Symbol | Name | Matrix |
|---|---|---|
| ⊕ | NOT | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ |
| Y | Pauli-Y | $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ |
| Z | Pauli-Z | $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ |
| H | Hadamard | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ |
| T | T | $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ |
| $R_z(\theta)$ | Z-rotation | $\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$ |
| ●<br>⊕ | Controlled NOT (CNOT) | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ |

Quantum gates, similar to classical gates, typically act only on a few qubits. Their action can be described by unitary matrices operating on the vector $x$ of coefficients of the wave function. A selection of commonly used gates is shown in Table A. Unitarity of these matrices implies that all quantum gates act reversibly. Similar to classical logic, circuit lines denote qubits and boxes denote gates acting on the qubits. Quantum gates operating on $n$ qubits can be represented by unitary matrices of dimension $2n$ x $2n$. We use a basis $\{|0\rangle, |1\rangle\}$ for the two states of a single qubit and $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ for the four states of two qubits. The action of these gates on an $n$-qubit register is obtained by tensor products of the gate matrices acting on one or two qubits with identity operations on the remaining qubits.

to be cooled to very low temperatures using a dilution refrigerator to achieve near-absolute-zero temperatures. We envision that large-scale quantum computers will operate in datacenters and be accessed remotely as a cloud service. Locally, one might have just a limited number of qubits with which to establish secure communication or to use as secure "quantum money" that is impossible to forge.

Cryogenic temperatures alone are insufficient to keep qubits in superposition for the runtime of a nontrivial quantum algorithm, and quantum error correction (QEC) is needed to extend the lifetime of quantum information.[8] QEC remarkably allows an arbitrarily large quantum computation to be performed efficiently as long as the rate of errors occurring on the physical qubits are below a given threshold value. The no-cloning theorem of quantum information, which makes quantum communication provably secure, unfortunately also makes QEC harder than classical fault tolerance. QEC can incur a large overhead and thousands or more "physical" qubits might be needed to realize a single error-protected "logical" qubit.

Topological qubits, which encode quantum information in a nonlocal property of a quantum state, are a promising way of reducing the QEC overhead by remaining robust to any source of local noise. Although QEC works at the software level, topological qubits have an intrinsic, hardware level of protection that allows the qubits to achieve longer lifetimes and lower rates of errors. With a lower rate of error, the overhead of QEC could be dramatically reduced by several orders of magnitude.

Given the cost of quantum fault tolerance, qubits will be a rare commodity in quantum computers, leading to a radical shift in design from their classical counterparts where memory is cheap. Instead of moving data to computational units, early quantum computers will move operations to the data, applying gates directly to the memory

## QUANTUM ALGORITHMS

Quantum computers are superior to classical ones because of quantum algorithms, which solve a range of problems better than the best-known classical algorithms (http://math.nist.gov/quantum/zoo). Here we list a selection of these algorithms. Figure A outlines the designs of many quantum algorithms.

- *Grover search.* Searching a database of unstructured items is a key problem in computer science. Given the ability to evaluate a function $f$: $\{0,1\}^n \rightarrow \{0,1\}$, the task is to find $x$ such that $f(x) = 1$. If such an $x$ does not exist, then return that no such item exists. Grover's algorithm solves the unstructured search problem on a quantum computer with only $O(\sqrt{N})$ evaluations of $f$ in the worst case, while classically $O(N)$ evaluations are required.
- *Period finding.* Given a periodic function $f(x) = f(x + r)$, find the period $r$. Classically, this requires $O(r)$ function evaluations. Quantumly, using Shor's period-finding algorithm that is at the root of his factoring algorithm, the same problem can be solved in time polynomial in the number of bits to represent the function values and arguments, which can be exponentially smaller than $r$.
- *Linear systems.* Given an $N \times N$ matrix $A$ and an $N$-dimensional vector $b$, find $x$ such that $Ax = b$. Classically, this requires time polynomial in $N$ using methods like Gaussian elimination or iterative solvers. Quantumly, for well-conditioned matrices whose condition number $K$ grows at most polylogarithmically in $N$, it requires time polynomial in $\log N$ to sample from the solution.
- *Sampling from a distribution.* Classically, the number of samples $N$ needed to estimate the mean of a random variable with error $\in$ scales as $N \sim \in^{-2}$. Quantumly, if a wave function can be prepared to represent the distributions, then the expected value can be measured with an effort $\sim\in^{-1}$ that is quadratically smaller.
- *Quantum walks.* A (random) walk is a succession of random steps on a given graph. The quantum generalization of classical random walks, quantum walks produce a quantum superposition whose probability distribution corresponds to that of a classical random walk. They can spread significantly faster than classical random walks,

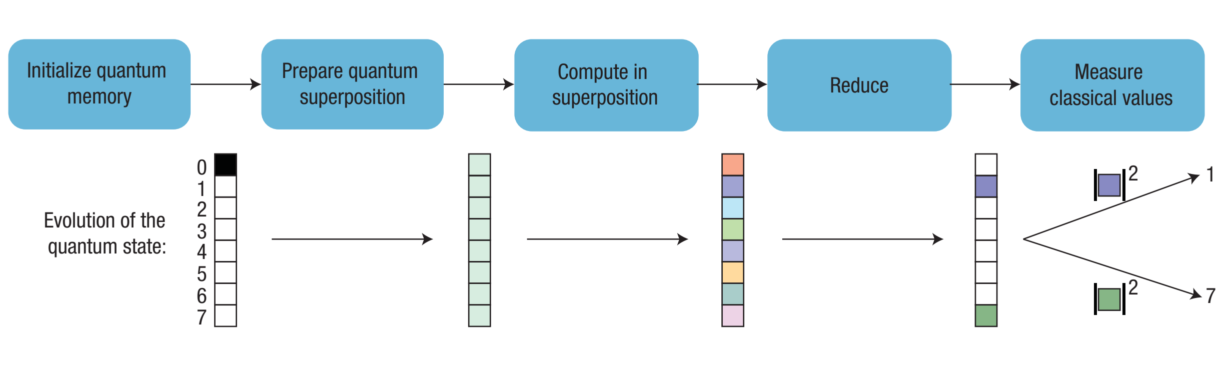

Initialize quantum memory
Prepare quantum superposition
Compute in superposition
Reduce
Measure classical values
Evolution of the quantum state:
0
1
2
3
4
5
6
7
| |2
| |2
1
7

**Figure A.** Design pattern for many quantum algorithms. Starting from a single classical initial state, a quantum superposition of many input states is prepared. A quantum algorithm is then performed on this superposition. Because reading out the superposition would only give the result for one random input, a collective reduction operation is performed before the quantum superposition is collapsed to a single classical result in a measurement process. If the final wave function is not concentrated on a single classical state, then the final result measurement will be probabilistic, determined by the squares of the wave-function entries. The lower part of the figure represents the wave function of size 2*N* (*N* = 3 qubits), illustrating the single-instruction, multiple-data (SIMD) analogy of quantum parallelism.

reducing hitting times and mixing times in walk-based algorithms.

- *Quantum tunneling.* Escaping from local minima is an important problem in heuristic optimization algorithms. Quantum systems can “tunnel” between local minima faster than classical local search algorithms can climb the barrier separating them, if the barrier is tall but narrow. It is currently unknown whether this can lead to faster solutions for hard optimization problems of practical interest.
- *Quantum simulation.* As discrete quantum devices themselves, quantum computers are by construction well suited for simulating discrete quantum systems, such as electrons in molecular orbitals. Quantum computers can be used to calculate electronic ground states and excited states. They can measure the energies of these states by mimicking interferometry and other properties by simulating experimental measurement procedures.

qubits. In many electronic systems, the firmware that controls and manipulates the computer or device is rarely updated. However, in a quantum computer, the firmware is inherently being optimized and updated continually during computation to reconfigure the device for the next steps.

An even bigger difference exists in the design of quantum algorithms themselves. To make use of the exponential parallelism inherent in a quantum superposition, computations need to be reversible; single-instruction, multiple-data (SIMD)-like; and follow certain design patterns that are discussed in the “Quantum Algorithms from a Classical Point of View” sidebar.

## NATIVE APPLICATIONS: QUANTUM CHEMISTRY AND MATERIALS SCIENCE

Simulating quantum models can be considered the native application of quantum computers. How better to understand a quantum-mechanical system than with a device that is inherently quantum mechanical?

The first applications of small quantum computers will be simple quantum models whose exact dynamics are already intractable for classical computers for about 100 lattice sites or orbitals. Accurate solutions of such model systems can be used to develop and validate new classical simulation methods which can then, in turn, simulate more complex systems than are tractable by a small quantum computer.

Recent algorithmic advances have substantially improved the scaling of algorithms for molecules or complex materials. A concrete example that has been worked out in detail is how to use quantum computers to understand biological nitrogen fixation, a process that could lead to more efficient artificial fertilizer production.[5] The iron molybdenum cofactor (FeMoco) of the enzyme nitrogenase can split a dinitrogen molecule, forming two ammonia molecules under ambient conditions, while the Haber–Bosch process, currently used in industrial processes,

## QUANTUM ALGORITHMS FROM A CLASSICAL POINT OF VIEW

Quantum computers' advantage over classical ones is sometimes oversimplified as quantum computers being able to perform computations on all possible inputs simultaneously. More accurately, many quantum algorithms can be viewed as instances of the design pattern shown in the "Quantum Algorithms" sidebar, Figure A. At a high level, such a quantum program can be seen as similar to MapReduce, but acting on all possible values on a single quantum CPU. After a quantum superposition of all input values is prepared, a computation is performed simultaneously on all values, before summary information is computed in a reduction step and read out.

While conceptually similar to MapReduce, quantum programs are different in many ways that show both the power and limitations of quantum computers. Performing operations on a superposition of exponentially many input values, a quantum computer can be viewed as an extreme form of an SIMD processor. While a quantum computer is exponentially more powerful than a classical computer, the SIMD analogy also points out some of the challenges: conditional instructions have to be implemented using masking registers (called control qubits), both branches of an if–else statement must be executed, and any loops must be iterated for the worst case.

Because quantum gates are unitary operations, they are by definition reversible. This requires that any classical function a quantum computer should perform is first converted to a reversible classical function, which can potentially incur large overheads in memory (for ancillae registers) and runtime.

Naively, one might think the next step would be the readout of the results. However, recall that when reading out an *N*-qubit quantum register, one only randomly obtains *N* classical bits of information and in turn, only one of the computational states making up the superposition results (see the "Quantum Bits and Gates" sidebar). If the computation has applied a classical function to a superposition of inputs, then this is equivalent to reading out the function value for a random input, which defeats the purpose of performing calculations in a superposition.

To profit from quantum parallelism, a quantum algorithm has to perform a global reduction function to all the qubits before reading out the quantum registers in a measurement. Such reduction operations include quantum Fourier transforms, used in Shor's period-finding algorithm, or amplitude amplification, used in Grover's search algorithm. Growing the currently limited selection of such operations is one of the challenges in designing new quantum algorithms.

is energy intensive and requires high temperatures and pressures. To elucidate how FeMoco achieves this feat, one needs to accurately calculate the energy for many configurations of the molecule along candidate reaction pathways. Although most of the protein can be simulated using approximate methods on classical supercomputers, accurate calculation of the active center of FeMoco (containing iron and molybdenum atoms) so far defies classical methods, but will be feasible using a quantum computer as a coprocessor. Given the relevance of catalytic processes in the chemical industry, quantum computers might find many commercially relevant applications in this area.

Similar algorithms have been developed for the simulation of crystalline materials, where approximate classical methods can be augmented by quantum computations to understand and design novel materials with exotic properties. These might include higher-temperature superconductors, new functional materials with switchable properties, and nontoxic dyes.

## QUANTUM-ACCELERATED CLASSICAL APPLICATIONS

The power of quantum mechanics can also be applied to solving purely classical computational problems by making use of quantum superposition and entanglement. Here we present a selection of likely and potential applications, based on the quantum algorithms in the "Quantum Algorithms" sidebar.

### Cryptography

Shor's algorithm[2] for integer factorization was the first demonstration that quantum computers could be used to solve hard computational problems outside the realm of quantum physics, and raised interest in this then-esoteric approach to computing. Integer factorization falls more broadly within the class of *hidden subgroup* *problems* (HSPs). Polynomial-time quantum algorithms exist not only for breaking RSA, but also for breaking Diffie–Hellman, the Digital Signature Algorithm (DSA), Buchmann–Williams, and other protocols, thus posing a threat to many common public-key encryption schemes.

### Search

Another famous example of a quantum algorithm outperforming classical ones is Grover's algorithm for searching the index of an item in an unsorted collection.[9] Although classically $O(N)$ queries are needed, a quantum computer can solve this problem with just $O(\sqrt{N})$ queries. Applications include finding minimal (optimal) values, determining graph connectivity, and pattern matching.

However, query complexity alone can be misleading. Indeed, for naive database lookup, each quantum query has to access the complete database, with $O(N)$ complexity, thus raising the full cost to $O(N^{3/2})$ and exceeding the classical complexity $O(N)$. Applications of Grover's algorithm will thus be implicit searches, where the value at a given index can be calculated efficiently.[10]

### Linear systems

Similar considerations apply to the linear systems algorithm,[11] which allows sampling from solutions $x$ of a well-conditioned linear system of equations $Ax = b$ with exponential speedup over classical algorithms, as long as the operation $e^{-iAt}$ can be efficiently implemented using quantum gates. For arbitrary $N \times N$ matrices, this requires at least $O(N^2)$ effort, which undoes any advantage of the quantum algorithm. Practical applications of the algorithm thus require matrices $A$ that can be described algorithmically. Such problems include finite difference and finite element discretization of partial differential equations, where one can determine, for example, solutions to electromagnetic scattering problems on exponentially fine meshes.

### Sampling

Statistical sampling using Monte Carlo methods plays an important role in many application areas, including the wide field of machine learning. Quantum computers offer two possibilities for quadratic speedup in these applications.

To accelerate sampling, qubits could be prepared in a superposition corresponding to the desired distribution. Naive readout of the qubits samples from that superposition with the error on the sample mean decreasing as $1/\sqrt{M}$ with the number of samples $M$. A quantum computer can achieve errors that decrease quadratically faster, as $1/M$, by making use of a quantum analog of the Fourier sampling theorem.

Quantum computers can also obtain a quadratic speedup in the mixing times of Markov chains when implementing them as quantum walks (see the "Quantum Algorithms" sidebar). This finds applications, for example, in stochastic optimization methods like simulated annealing,[12] where a quadratic acceleration is highly desired for hard optimization problems.

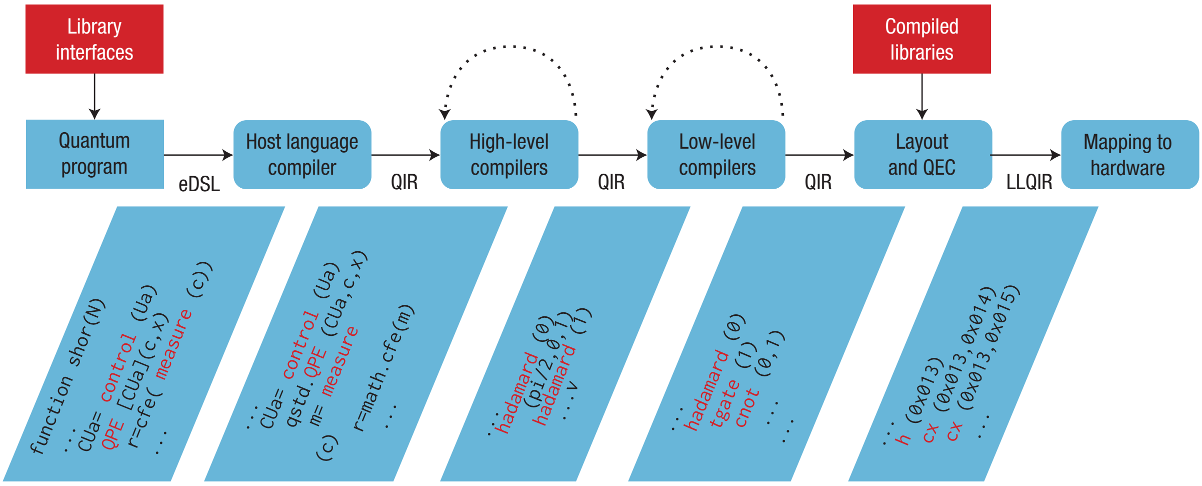
### Privacy

With increasing concern about privacy, the broadest impact of quantum devices on computing might not even come from quantum algorithms outperforming their classical counterparts but in privacy for computing. Indeed, blind quantum computing[13] allows a user to run quantum algorithms without the quantum computer operator being able to detect what is being computed. For example, a doctor might want to compute the likelihood of a patient contracting a disease, but might not want an insurance company to know about the computation as it could raise premiums. For many users, provable privacy and security of cloud computing might be more important than accelerating calculations with quantum speedup. On the contrary, in many settings one might even be willing to take a performance hit in return for better security. This could ultimately be one of the biggest applications for quantum computers.

## SOFTWARE ARCHITECTURE

Until recently, there has been a divide between quantum algorithm theory and experiments, with theory focused on mathematical proofs of algorithmic improvements and experiments focused on controlling qubit lifetimes. In turn, quantum computations have thus far been mostly described at the level of mathematics and logic-gate operations (see the "Quantum Bits and Gates" sidebar), compiled and manually optimized for a small quantum device or simulator. Although some quantum programming languages and compilers exist, a complete software architecture remains to be developed. As hardware steadily advances past single-qubit devices toward 100-qubit devices and beyond, computer scientists, engineers, and developers face the challenge of designing and architecting a software framework for programming and controlling a scalable, fault-tolerant quantum computer.

High-level programming languages and optimizing compilers are designed



```
function shor(N)
...
CUa= control (Ua)
QPE [CUa](c,x)
r=cfe( measure (c))
...
```

```
...
CUa= control (Ua)
qstd.QPE (CUa,c,x)
m= measure
(c)    r=math.cfe(m)
...
```

```
...
hadamard (0)
(pi/2,0,1)
hadamard (1)
...v
```

```
...
hadamard (0)
tgate (1)
cnot (0,1)
...
...
```

```
...
h (0x013)
cx (0x013,0x014)
cx (0x013,0x015)
...
```

**FIGURE 1.** A software stack for quantum computing. (T. Häner et al., "A Software Methodology for Compiling Quantum Programs," 2016; http://arxiv.org/abs/1604.01401.)

to provide layers of abstraction of the computation, creating a simple, modular programming environment. The development of software in advance of a fully scalable hardware architecture allows the verification and analysis of software components and rapid innovation in algorithm design. This idea dates back to Fortran (FORmula TRANslation), which was created to transition users from mathematical formulas and machine code to algorithm design and specification. Fortran's simplicity allowed more rapid programming and execution, improving programming speed by 500 percent and reducing the length of programs.

Quantum computing requires a similar software revolution, as the algorithms and hardware are complex systems. Novel constructs for programming, optimizing, and controlling quantum hardware are necessary to fully harness and manipulate quantum states for algorithmic advantage. Debugging tools will require innovative approaches due to the entanglement of qubits and the inability to copy. In this regard, simulators and emulators will be critical for the design and testing of quantum algorithms and circuits. Early design decisions can result in substantial cost savings on the path to finalizing an end-to-end system, and enable the verification, analysis, and redesign of key components of the system.

Figure 1 shows a software stack for quantum computing,[14] which should provide well-defined interfaces between the software and hardware systems and should enable programming of any quantum algorithm of any size for any target architecture.

At the top of the figure is a high-level language in which one writes quantum programs. Because quantum programs consist of an intimate mixture of classical and quantum instructions, a compelling case can be made for an embedded domain-specific language such as Quipper[15] or LIQUi|⟩[16] in order to maximally leverage classical language and compilation features.

To ensure flexibility and modularity, the high-level compilers are hardware agnostic and, combined with quantum metafunctions like conditional instructions and user annotations, enable a very high level of quantum code optimization. The low-level compilers of the stack will allow automatic translations to hardware instructions for specific back-end quantum devices.

The development of standard libraries for quantum subroutines, including those for arithmetic operations and complex quantum subroutines, will speed up the development of new quantum programs. Autotuning of these quantum libraries allows the best representations for a given target back end to be determined, and in turn can result in substantial cost savings in the final implementation of the quantum algorithms on any hardware.[14] Building high-level quantum applications on libraries will let developers take advantage of algorithmic improvements in those components, as in classical applications. Recent examples of such optimizations in the quantum domain include substantial complexity reductions in simulating quantum chemical systems and savings of several orders of magnitude in quantum gate synthesis.

The bottom of the stack supports a variety of back-end devices, including simulators, emulators, resource analyzers, or any proposed quantum hardware implementation. Whereas the upper layers of the stack are agnostic to whether the hardware back end is an ion trap, a superconducting system, a quantum dot, or a topological quantum computer, the lower layers allow tuning and control optimization for the specific device design, including layout and scheduling.

## OUTLOOK

Experimental development of qubits has progressed to the point where scalable qubit technology already exists—within the next few years, we can expect to see the development of small quantum computers with more than 100 physical qubits. This will allow for the creation of early applications using short quantum algorithms that do not exceed the lifetime of these qubits, and could provide the first demonstration of quantum computers outperforming classical computers on certain problems.

More importantly, these quantum computers will accelerate the path toward the development of logical qubits using QEC as well as engineering efforts to scale to bigger, robust quantum computers. Quantum software efforts need to match the hardware developments and encompass the entire stack, from low-level control of qubits to high-level applications.

Quantum algorithms for quantum simulations and factoring have been optimized to the point where they can be viewed as realistic applications on small- to medium-scale quantum computers. For other applications, including linear systems, sampling, and machine learning, algorithmic optimizations and in-depth resource estimates are still required.

Although quantum algorithms are generally much less well understood than their classical cousins, the search continues for which important problems can be solved much faster on quantum computers and which cannot. Answering this question has become increasingly important, and industrial efforts are emerging that focus on the impending quantum revolution. The quantum hardware architecture that could ultimately lead to scalability will require today's computer scientists, physicists, mathematicians, and engineers to work together to overcome the exciting challenges on the path toward universal quantum computing.

## ABOUT THE AUTHORS

**KRYSTA M. SVORE** is a senior researcher and manager of the Quantum Architectures and Computation Group at Microsoft Research in Redmond, Washington. Her research interests include the applications and fault-tolerant programming of quantum computers as well as machine learning methods for Web applications. Svore received a PhD in computer science with highest distinction from Columbia University. She is a Senior Member of ACM and was a member of the winning team in the 2010 Yahoo Learning to Rank Challenge. Contact her at ksvore@microsoft.com.

**MATTHIAS TROYER** is a professor of computational physics at ETH Zurich and a consultant for the Quantum Architectures and Computation Group at Microsoft Research. His research interests range from simulations of quantum materials to quantum computing and ecosystem modeling. Troyer received a PhD in physics from ETH Zurich. He is a Fellow of the American Physical Society, recipient of the Annesuhr Rahman Prize for Computational Physics, and a trustee of the Aspen Center for Physics. Contact him at troyer@phys.ethz.ch.

## ACKNOWLEDGMENTS

We thank Doug Carmean, Thomas Häner, and Damian Steiger for discussion and Thomas Häner for help with the figures for this article.

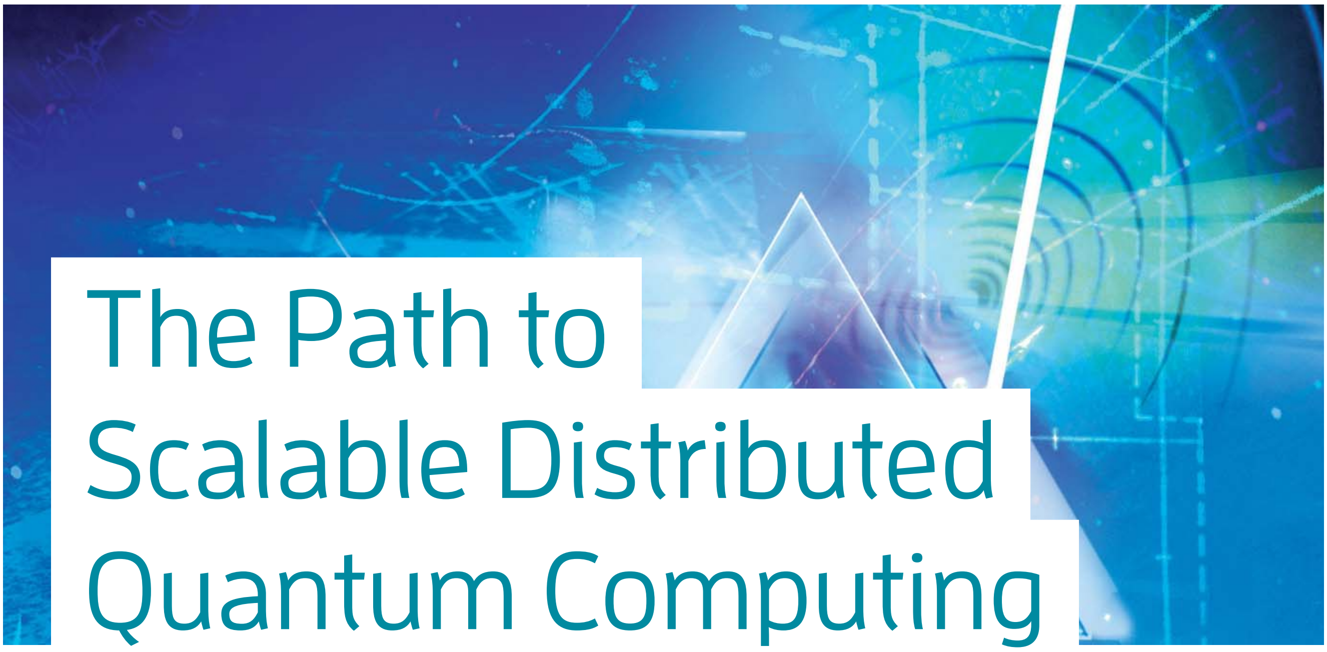
## REFERENCES

1. R. Feynman, "Simulating Physics with Computers," *Int'l J. Theoretical Physics*, vol. 21, nos. 6–7, 1982, pp. 467–488.
2. P. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *Proc. 35th Ann. Symp. Foundations of Computer Science* (FOCS 94), 1994, pp. 124–134.
3. R. Van Meter and S.J. Devitt, "Systems for Local and Distributed Quantum Computation," *Computer*, vol. 49, no. 9, 2015, pp. 31–42.
4. D. Augot et al., *Initial Recommendations of Long-Term Secure Post-Quantum Systems*, tech. report, Eindhoven Univ. of Technology, 2015; https://pqcrypto.eu.org/docs/initial-recommendations.pdf.
5. M. Reiher et al., "Elucidating Reaction Mechanisms on Quantum Computers," 2016; https://arxiv.org/abs/1605.03590.
6. P. Wittek, *Quantum Machine Learning*, Academic Press, 2014.
7. J. Adcock et al., "Advances in Quantum Machine Learning," 2015; http://arxiv.org/abs/1512.02900.
8. D. Lidar and T. Brun, *Quantum Error Correction*, Cambridge Univ. Press, 2013.
9. L.K. Grover, "A Fast Quantum Mechanical Algorithm for Database

Search," *Proc. 28th Ann. ACM Symp. Theory of Computing* (STOC 96), 1996, pp. 212–219.

10. G.F. Viamontes, I.L. Markov, and J.P. Hayes, "Is Quantum Search Practical?," *Computing in Science and Eng.*, vol. 7, no. 3, 2005, pp. 62–70.
11. A.W. Harrow, A. Hassidim, and S. Lloyd, "Quantum Algorithm for Linear Systems of Equations," *Physical Rev. Letters*, vol. 103, no. 15, 2009; http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.103.150502.
12. R.D. Somma et al., "Quantum Simulations of Classical Annealing Processes," *Phys. Rev. Letters*, vol. 101, 2008; http://arxiv.org/abs/0804.1571.
13. A. Broadbent, J. Fitzsimons, and E. Kashefi, "Universal Blind Quantum Computation," *50th Ann. Symp. Foundations of Computer Science* (FOCS 09), 2009, pp. 517–526.
14. T. Häner et al., "A Software Methodology for Compiling Quantum Programs," 2016; http://arxiv.org/abs/1604.01401.
15. A.S. Green et al., "Quipper: A Scalable Quantum Programming Language," *ACM SIGPLAN Notices*, vol. 48, no. 6, 2013, pp. 333–342.
16. D. Wecker and K.M. Svore, "LIQUi|⟩: A Software Design Architecture and Domain-Specific Language for Quantum Computing," 2014; https://arxiv.org/abs/1402.4467.

Selected CS articles and columns are also available for free at **http://ComputingNow.computer.org**.



# The Path to Scalable Distributed Quantum Computing

**Rodney Van Meter,** Keio University Shonan Fujisawa Campus
**Simon J. Devitt,** RIKEN Center for Emergent Matter Science

*Researchers are fabricating quantum processors powerful enough to execute small instances of quantum algorithms. Scalability concerns are motivating distributed-memory multicomputer architectures, and experimental efforts have demonstrated some of the building blocks for such a design. Numerous systems are emerging with the goal of enabling local and distributed quantum computing.*

Increasingly, quantum computers and networks are expanding already astonishing classical computing and communication capabilities.[1,2] As the sidebar "Key Concepts in Quantum Computing" describes, quantum computing has six underlying concepts. Each concept is simple, but collectively they imply that classical computation is incomplete and that quantum effects can be used to efficiently solve some previously intractable problems.

In the 1980s and 1990s, researchers developed several quantum algorithms and laid the foundation of quantum computational complexity, but they did not fully grasp the process of creating new quantum algorithms. From the early 2000s, researchers have begun to more deeply understand this process, which has caused an explosion of proposed quantum computing algorithms (http://math.nist.gov/quantum/zoo) in areas ranging from quantum chemistry to astrophysics to matrix operations relevant to machine learning.[3] Some algorithms offer only a polynomial speedup over competing classical algorithms; others offer super-polynomial speedups in asymptotic complexity. However, in many cases, studies have not yet investigated the algorithm's interaction with quantum computer architecture to determine constant factors, fidelity demands, and resource requirements. In short, the required size, speed, and fidelity of a commercially attractive quantum computer remain open questions.

Experimental groups are now fabricating quantum processors powerful enough to execute small instances of quantum algorithms and demonstrate quantum error correction (QEC) that extends the lifetime of quantum

## KEY CONCEPTS IN QUANTUM COMPUTING

Many quantum phenomena exhibit a set of discrete states, such as an atom's energy levels, the direction of an electron's spin relative to the magnetic field, or horizontal and vertical light polarization. Each quantum bit, or qubit, has two separate, orthogonal states: the zero and one states.

### SUPERPOSITION

Superposition in quantum systems acts in a somewhat analogous way to classical wave mechanics. Light polarized at a 45-degree angle is an even superposition of horizontal and vertical polarization. Likewise, it is possible to create superpositions of two electron spin states or two atomic energy levels. The probability that, in the end, a certain outcome will be found is related to the relative amounts of zero and one in the superposition.

### ENTANGLEMENT

When a quantum system has more than one particle or qubit, describing their states independently is not generally possible because the qubits can be entangled in such a way that their respective states become interdependent. This correlation, which is stronger than dependent classical probabilities, forms the basis of quantum communication. Entanglement cannot be used to communicate faster than the speed of light, even though entangled particles that are far apart will show correlations with no classical explanation when used appropriately.

### AMPLITUDE AND PHASE

As the quantum system grows, $n$ qubits have $2^n$ possible states, 0…0 to 1…1, just as with classical bits; the set of qubits is the register. Because the total state is described by the wave amplitude and phase of each possible state, a complete classical state description can require as much as $O(2^n)$ memory. The quantum algorithm designer's job is to shuffle amplitude from value to value, altering superposition while manipulating the phase to create interference: when phases are the same, interference is constructive, which increases a particular outcome's probability. When phases differ, interference is destructive, which decreases the outcome's probability.

### REVERSIBILITY

In a circuit-based quantum computer, the algorithm designer composes an algorithm by defining a series of gates that change one or two qubits at a time, roughly analogous to classical instructions or Boolean logic gates. In contrast to classical Boolean logic, these gates must be reversible or, in mathematicians' terms, unitary. The controlled-NOT (CNOT) is one such common building block.

### MEASUREMENT

A significant exception to the requirement for reversibility is measurement, which involves extracting a numeric value from the quantum system's register. The extraction causes the superposition to collapse into a single state. The choice of state is random, with probabilities depending on the states' relative amplitudes, taking interference into account. Measurement destroys entanglement.

### DECOHERENCE

Quantum states are very fragile and must be well isolated from the environment. However, over time, errors inevitably creep in—a process known as decoherence. The natural classical solution would be to keep extra copies of fragile data, but the no-cloning theorem, a fundamental tenet of quantum mechanics, dictates that it is not possible to make an independent copy of an unknown quantum state.

data, adding urgency to architectural investigations. Although other options continue to be explored, effort is coalescing around topological coding models as the most practical QEC implementation option on realizable microarchitectures. Scalability concerns have also motivated the proposal of distributed memory multicomputer architectures, with experimental results demonstrating some of the basic building blocks to make such designs possible. To gauge progress toward building a scalable quantum computer, we researched

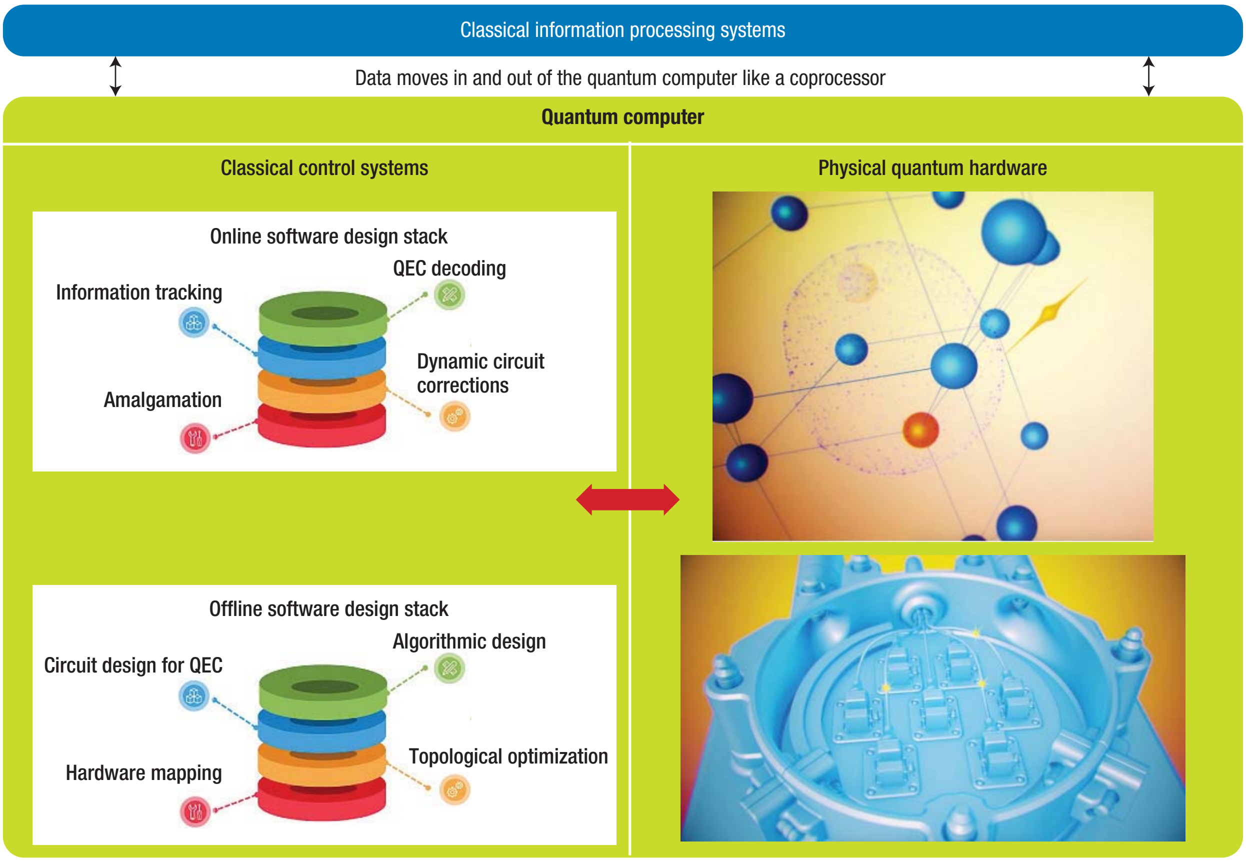

Classical information processing systems
Data moves in and out of the quantum computer like a coprocessor
Quantum computer
Classical control systems
Physical quantum hardware
Online software design stack
Information tracking
QEC decoding
Amalgamation
Dynamic circuit corrections
Offline software design stack
Circuit design for QEC
Algorithmic design
Hardware mapping
Topological optimization

**FIGURE 1.** How a quantum computer interacts with the classical information processing world. Acting akin to a classical coprocessor, a quantum computer will accept suitable outside queries and problem specifications. Aside from this interaction, an entirely separate classical system is needed to control the quantum hardware itself. QEC: quantum error correction.

the latest results from various systems with a focus on those emphasizing scalability through networks.

## BEYOND CLASSICAL COMPUTING

Figure 1 shows the basic principle of a quantum computer interacting with the classical information processing world. This interaction is akin to querying a classical coprocessor from an outside classical process. Along with this interaction is a second classical system within the quantum computer, which handles the interaction with the hardware itself.

As researchers began to explore the notion of quantum computing, David DiVincenzo, widely regarded for his work in early quantum computing, laid out five criteria that a technology must meet for architects to use it in building a basic quantum computer:

- the technology must have an extensible register of two-level systems, usable as quantum bits (qubits);
- the register must be initializable to a known state;
- the technology must have a universal gate set;
- qubits and operations on them must exhibit adequate coherence time and fidelity for long quantum computations; and
- single-shot measurement of data must be possible.

A few points are worth noting. In the first criterion, “extensible” hides substantial engineering complexity. The ability to achieve any proposed algorithm, implied in the third criterion, fits within the basic framework of quantum computation. The fourth criterion was at the center of quantum computation’s early criticism and led to the development of QEC and fault tolerance. Finally, the fifth criterion ensures that numeric data can be extracted, which is an essential function of a practical computer.

DiVincenzo later added two criteria to ensure scalability through the use of photonic interconnects or to create networks that can deliver entangled states to applications: the technology must be able to convert between stationary qubits and photons (flying qubits) as well as be able to capture and control photon routing.

To DiVincenzo’s criteria, we propose adding practical engineering constraints: systems must be small enough, cheap enough, and reliable enough to be practical and fast enough to be useful. Implementation limitations make locally distributed computation imperative, which requires system area networks that are fast, high-fidelity, and scalable.

Tightly coupling small quantum computers to form larger multicomputers so as to scale purely numerical, monolithic algorithms helps pave the path to distributed quantum algorithms and sensing. In these applications, the use of distributed quantum states will improve scientific instruments' sensitivity and accuracy and will augment classical cryptographic capabilities.

## QUANTUM COMPUTING ARCHITECTURES

Theoretical architectures for large-scale quantum computation now rely almost exclusively on topological QEC models. The two classes of topological codes dominating recent architectural designs are surface codes, which work on 2D lattices, and Raussendorf codes, which work on 3D lattices.[4] Each proposed system uses a different physical technology that defines the qubit. Adapting error-correction models to the quantum hardware's physical restrictions has led to multiple architectural designs. Because complete systems will be large scale and qubits have a far greater physical size than transistors, the system's macroarchitecture will most likely be a multicomputer design.

### Advantages of topological codes

Topological codes are so named because their structure can be defined through a repeating set of operators over a small number of closely connected qubits, while the encoded information's properties are defined through operations that act on the entire 2D (or 3D) array. These codes have been adopted broadly for three reasons:

- memory lifetimes and gate error rates remain a challenge for experimenters, and surface codes have thresholds approaching 1 percent, depending on the physical model;
- the intrinsic nearest-neighbor structure ensures that the physical hardware does not require long-range interactions; and
- the software-driven programming model for manipulating logical qubits allows runtime resource allocation suitable for any application algorithm, including (within limits) the adjustment of error-correction strength.

One perceived drawback is the high resource cost, with many physical qubits per logical qubit, but analyses suggesting large numbers were conducted assuming physical error rates above the operational thresholds of other codes.

Topological coding models allow architectures to be designed with a high level of modularity. Small repeating elements plug together to form a computer of arbitrary scale; we refer to the architecture of the unit for executing error correction as the microarchitecture. The comparative simplicity of the hardware structure makes it far easier to experimentally build, and currently the biggest challenge is engineering qubit components with the required fidelity for topological error correction to become effective.

### The topological coding model

Several detailed reviews of the topological coding model cover the functioning of both the error correction and logical computation. Although the model is complicated, the basic hardware configuration is quite simple. Figure 2 shows the model's four main elements.

**Encoding data.** As Figure 2a shows, half the qubits in 2D surface code are data qubits and half are syndrome qubits. Figure 2b shows the syndrome extraction for two circuits that run continuously in parallel over the entire surface to protect against physical system errors. Computation is achieved by temporarily switching off some of these circuits, creating defects (holes). If every circuit in every green or yellow diamond is switched on, the quantum state will be so tightly constrained that a logical qubit cannot be encoded. Switching off one (or a connected block) of these circuits introduces a degree of freedom within the system that is used as a logical qubit. As the defects' size and separation increase linearly, the information's logical error rate drops off exponentially. Consequently, *d* becomes the code distance. Changing the regions switched off from cycle to cycle essentially manipulates encoded defects, allowing the translation of a compiled fault-tolerant quantum circuit into the computer's physical control signals.

**Plumbing pieces.** Figure 2c shows a plumbing piece that links a geometrical representation of a quantum circuit to the total number of qubits needed in the surface (2D) or Raussendorf code (3D). To visualize the intertwining of the defects, we used a negative-space representation with the defect in red and the active lattice portions not shown. For the surface code, the three dimensions of the figure are the two physical dimensions of the lattice in Figure 2a and time, which flows from the front of the image to the rear.

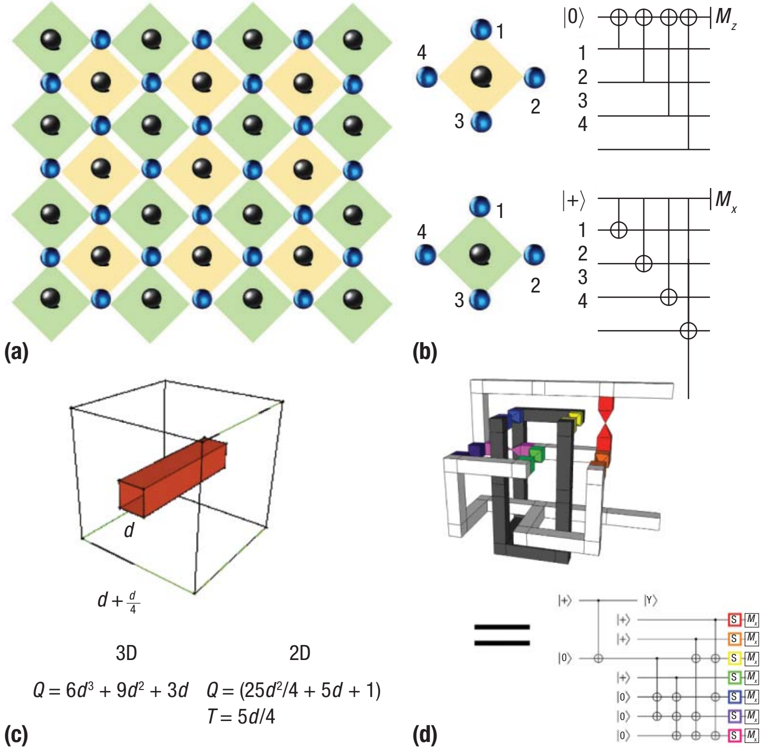


**FIGURE 2.** Elements of the topological coding model. (a) 2D structure, with data qubits (blue) and syndrome qubits to extract error-correction information (black). (b) The two circuits run continuously across the lattice to detect errors. The center black qubit accumulates the parity of the four surrounding blue qubits in one of two possible ways (denoted by the yellow or green diamond). In the circuit diagrams (right), time flows left to right; the vertical symbols represent the quantum operations (gates) used, ending each time with a measurement (either $M_Z$ or $M_X$), which gives the parity as a classical bit. (c) Negative-space representation of the defect (red). The defect is a square region of circumference *d* (measured in terms of the colored diamonds) that encodes the information. The size of the plumbing piece determines the number of qubits required for an implementation in the Raussendorf lattice or the number of qubits (*Q*) and time steps (*T*) for the surface code (equations at the bottom). The resulting logical qubit is a plumbing piece fundamental to the topological quantum circuit. (d) The geometric structure of plumbing pieces that define how regions of the computer are arranged to enact the state distillation program (lower right). Time flows from left to right.

Figure 2d shows the geometric structure of plumbing pieces that define how regions of the computer must be arranged to enact the state distillation program. The physical 3D size of this structure determines how much time and how many qubits are necessary for an error-corrected program. A large program, such as Shor's factoring algorithm, might replicate this structure millions of times.

### Older coding models

Designs for large-scale quantum computers that predate the development of the topological coding model rely on multiple layers (concatenation) of classically derived QEC codes, and some researchers continue to pursue this approach. These older codes are simpler to decode at runtime, and, if the underlying technology supports long-distance interaction between qubits, the primary technical challenge is the higher fidelity relative to the topological coding model required for physical operations.

## QUANTUM COMPUTING TECHNOLOGIES

Since the landmark 2010 review in *Nature*,[1] experimental work toward large-scale quantum computers has made progress toward scalable systems. Figure 3 shows a qualitative summary of seven major technologies receiving significant academic and industrial attention. Of the dozens of technologies under development, we chose to highlight the first six—not only because of their experimental progress, but also because relatively concrete proposals for complete, scalable architectures have been analyzed. We included anyons because they are the only technology under development that might eliminate the need for QEC.

We believe that these technologies are complementary and have a well-defined place within an emerging technology sector. Developmental time frame, cost, execution speed, and physical size are metrics that can vary by orders of magnitude among systems, and generally the systems that have the potential for higher performance are less developed.

### Ion traps

Quantum computing based on ion traps was an early experimental success story for quantum information,[5] because researchers were able to build on prior ion-trap development motivated by applications such as atomic clocks. Carefully controlled electrical fields and lasers ionize, trap, and manipulate the states of individual atoms, while holding them in place in an ultrahigh vacuum. Quantum computing using trapped ions was first proposed in 1995, and the demonstration of primitive gate operations soon followed.

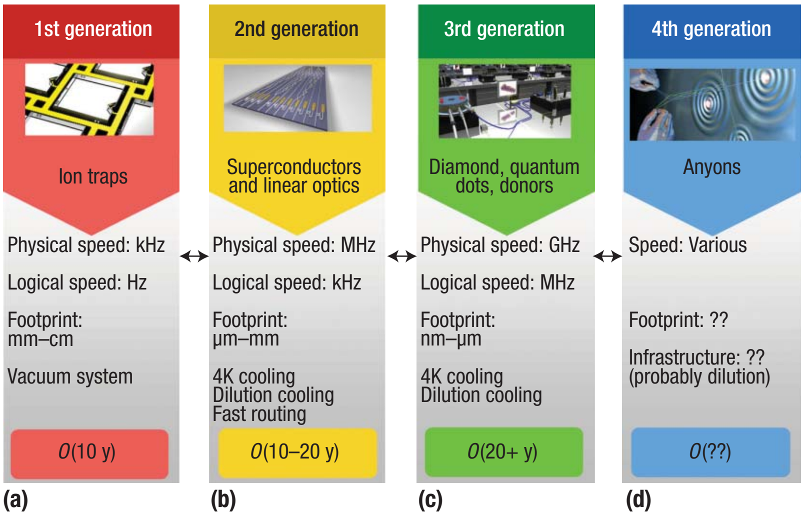


**FIGURE 3.** Generations of quantum computers. Qualitative assessment of seven major quantum computing technologies—ion traps, superconductors, linear optics, diamond, quantum dots, donors, and anyons. Although each technology is categorized in a single generation, specific implementations could straddle generations in one or more metrics. *Physical speed* is the slowest gate or set of physical gates needed to enact the physical operations that topological coding requires. *Logical speed* is the slowest encoded gate operation in a suitable universal gate set. *Footprint* is the expected size of the qubit and includes the qubit's associated control and infrastructure.

However, a large-scale quantum computer with all of the qubits in a single trap is impractical for several reasons, such as slower gate times, cross talk when applying quantum gates, limited operational parallelism, and increases in decoherence rates. To combat these problems, researchers came up with the idea of segmented traps.[5] In this approach, the microarchitectural model uses a series of DC electrodes that can move the electrostatic potential along a trapping pathway, essentially dragging the ion with it. Individual qubits can be placed into storage regions and then moved to interaction zones for gates with multiple qubits. This segmented design requires delicate control to ensure that ions can be moved without losing them around complex trapping geometries.

**Scalable design.** A simple approach to a large system based on segmented traps is a monolithic design in which individual traps are fabricated, aligned, and interconnected to form the complete computer. The advantage of this design is that physical operations for surface code are as simple as possible. The disadvantages are the need for a vacuum infrastructure surrounding the entire computer and the sheer size of a machine containing as many as 33 billion qubits.

A second approach is to further divide the computer into small elementary logic units (ELUs), where each ELU can be a segmented trap with tens to thousands of qubits. The ELUs are interconnected using probabilistic optical connections achieved by optically exciting two distant ions and using the emitted photons. This communications channel allows the connection of independent ELUs to form a larger multicomputer.

Although this approach mitigates the infrastructure issues that would plague a monolithic ion-trap computer, it introduces complications. The optical connections that allow the creation of entanglement between ions are intrinsically probabilistic. Factoring in inefficiencies in emitted-photon capture and detectors and loss through optical switches, the system must attempt entanglement many times for each successful connection. Initial experiments required tens of minutes to establish entanglement between ions, although performance has since improved to five entanglements per second.

**In summary.** Systems based on ion traps are making rapid progress and might well be the first to outperform classical quantum computers. However, size, speed, and potential cost could ultimately restrict scalability.

## Superconductors

Many of the same principles that apply to ion trapping also apply to the other five technologies, including superconductors. Superconducting quantum computers have seen explosive success in the past five years and are the principal technology of two of the first industrial players in the quantum sector, Google and IBM:[6] both are using superconducting qubits and surface-code techniques to push forward in building large-scale systems.

Superconducting qubits come in several varieties, with the most successful using a quantized amount of current in a superconductor loop. With intrinsic gate times of hundreds of nanoseconds and qubit footprints of hundreds of square microns ($\mu m^2$), superconducting qubits can be considered a generation beyond ion traps. Experiments have demonstrated the execution of one- and two-qubit gates, with initialization and measurement error rates below the fault-tolerant threshold within a single device.

**Scalable design.** There have been several proposals for large-scale quantum computer architectures. Monolithic approaches have shown the necessary building blocks for fault-tolerant computation using surface code, but a major challenge is to scale to a 2D nearest-neighbor qubit array while not degrading the individual qubit error rates. IBM demonstrated a 2 × 2 superconducting qubit array, but larger arrays will be needed and the fabrication and placement of the necessary control wiring for each qubit is an open engineering problem. As with ion traps, distributed designs could mitigate infrastructure and control issues for a large-scale machine, but they introduce more complicated protocols needed to realize fundamental gate libraries. The libraries are necessary to implement either surface code or other error-correction techniques across a slower, more error-prone interconnect.

**In summary.** Recent rapid advances have raised the strong possibility that superconducting quantum computers, not ion traps, will be the basis of the first digital quantum computer that can outperform classical machines on relevant quantum problems. Besides IBM and Google, startups such as Rigetti and Quantumcircuits are specifically targeting this platform.

## Linear optics

Linear optics was one of the first platforms to demonstrate the building blocks of quantum computation.[7] Arguably, the initial theoretical foundation for linear optical quantum computing is attributable to the 2001 *Nature* article describing the possibility of measurement-induced nonlinearities and hence of universal computation.[8]

After demonstrating the building blocks of linear optical quantum computation in bulk optics, development efforts moved to integrated optics, in which individual photons are routed through etched waveguides in a bulk material (ostensibly silicon). Early work was extremely successful, with high-fidelity circuits performing small quantum programs. Recent explorations have focused on integrating all aspects of a universal quantum computing system on chip, including the photon sources, detectors, and waveguides.

High-fidelity, high-efficiency, on-demand single-photon sources still remain the Achilles' heel of linear optics technology. There are generally two approaches to creating photon sources: use an atomic-based photon emitter to produce on-demand photons, or use a combination of spontaneous parametric down conversion (SPDC) sources and optical switching to create a multiplexed source. Using multiplexed sources translates the source problem into constructing single-photon switches with very low loss, which is the current research focus.

**Scalable design.** Architecture for linear optics has also progressed with two general approaches for realizing a topologically protected machine.

The first approach involves slowly constructing a 3D Raussendorf lattice by probabilistically fusing together larger and larger components. The downside is the need to optically store the cluster as it grows and to route together smaller subclusters that might have been successfully prepared in distant physical locations in the computer. This type of architecture has high overhead and nontrivial routing issues requiring very-low-loss single-photon switches.

The second general approach is the ballistic model, in which photons are sent through a fixed network of fusion gates, producing an incomplete (Swiss cheese) graph state. This model relies on boosted fusion—a technique that uses entanglement to increase the gate's success probability—to achieve higher than a 63 percent probability of success for individual fusion gates. This probability level ensures that the created lattice can percolate (create an unbroken line of connections reaching edge to edge across all its spatial dimensions). The ballistic model relaxes routing and storage requirements, but incurs the calculation cost of converting the incomplete lattice to a perfect Raussendorf lattice in real time.

**In summary.** Linear optical quantum computers still have significant potential, but both theoretical and experimental results are incomplete. Comparatively low infrastructure cost is a significant selling point for this technology.

## Diamond

Impurities in diamond have long been of interest as a potential technology for both large-scale quantum computing and communications.[9] The nitrogen vacancy (NV) center, which creates an electrical potential that can trap a single electron, is by far the most intensively researched type of impurity for potential use in quantum technologies. Diamond is of interest because it can be used to couple the NV center with a photon at optical frequencies, thus naturally interconnecting the quantum memories used as stationary qubits and the photons used as flying qubits on communications links.

**Scalable design.** Diamond-based quantum computing architectures have been proposed in both monolithic crystals and distributed diamond arrays. In the first approach, numerous

NV centers are fabricated within a single crystal; in the second, diamond arrays are optically connected. As is true of essentially all modern architectures, both approaches are based on the use of surface code (2D) or the Raussendorf model (3D).

An ensemble array of NV centers was successfully coupled to a superconducting flux qubit in 2011. In this system, the diamond layer was envisaged as a method to couple superconducting qubits (which themselves would couple via microwave photons) via optical photons. In 2015, researchers used diamond-based qubits in the first experimental demonstration of entanglement to close all possible loopholes.[9]

**In summary.** Although diamond-based quantum computers are not as well developed as other systems, several research groups are focusing on this technology. Efforts have identified the elements required for large-scale computation, but researchers have not yet shown high-enough-fidelity operations or a universal set of gates within a single device. Diamond has compelling advantages, however. Relative to ion traps and superconductors, its infrastructure costs are less stringent. Vacuums are unnecessary, and cooling can be limited to 4 Kelvin, rather than millikelvin temperatures. Lower potential infrastructure costs and fast operation times make diamond an ideal bridge between second- and third-generation quantum technologies.

## Quantum dots

Quantum dots trap individual electrons at the boundary between different semiconductor materials, and can be controlled optically, electrically, or magnetically.[10] They are not as experimentally advanced as superconductors or ion traps, in large part because of their sensitivity to noise, but recent work is striving to overcome that obstacle. Quantum dots have the potential for denser integration and fast operation, but this conceptual advantage is tempered by the apparent need for more development time. They also have many uses other than quantum computing, including sensing, communications, and classical computing. Even so, research groups worldwide are working toward a computer based on quantum dots.

**Scalable design.** Progress has been substantial since the original 1998 device architecture. Some of the quantum-dot groups are not in the academic sector and thus limit their public information, so exact progress is difficult to assess. As with each technology included here, researchers assume a large-scale structure based on surface codes.

Experimental demonstrations of building-block protocols have also been notable. Addressable quantum-dot qubits with fault-tolerant levels of control fidelity have been demonstrated, along with a full two-qubit logic gate. Unlike systems based on linear optics, ion traps, and superconducting qubits, significant issues remain related to the reliable fabrication of individual qubits. The material's atomic structure, including isotopes of the passive substrate, affects the quantum state held in a quantum dot, particularly its memory lifetime.

Large-scale architectures have been proposed for optically controlled dots, but current trends are toward electrically controlled dots, in which the on-chip placement of the classical control traces makes it difficult to place the dots close to each other in a scalable arrangement.

**In summary.** Multiple research groups worldwide—most notably in the US, Australia, and Japan—are perfecting the fabrication of qubits based on quantum dots, and are beginning to demonstrate the control and fidelity required for integration into larger computational arrays. However, further experimental development is needed to demonstrate the required building blocks of a scalable machine.

## Donors

Donor-based quantum computing systems use semiconductor dopants that provide an extra, unpaired electron.[11] In room-temperature semiconductor operation, the extra electron moves through the material, but in quantum computing systems, the material is cooled to millikelvin temperatures and the electron remains bound to its dopant atom. The goal is to use these individual electrons as spin qubits, sometimes in conjunction with the nuclear spin of nearby atoms. These systems are exemplified by the use of phosphorus in silicon (Si:P), which has shown significant experimental progress in the past five years. The original 1998 architectural proposal did not consider the challenges of error correction or algorithmic implementation.[12] Since then, several generations of architectures for Si:P quantum computers have been proposed.

**Scalable design.** Experimentally, there were significant challenges to simply build a functional qubit using phosphorus donors, as an atomically precise array of phosphorus donors must be embedded within an otherwise isotopically pure crystal of silicon-28. The actual placement of the phosphorus donors within the crystal can be either top-down or bottom-up. The top-down method involves directly injecting the phosphorus through a focused ion beam. Direct injection is not atomically precise and can significantly damage the silicon substrate, requiring the material to be annealed to heal the structure, which can also cause donors to move. In the bottom-up approach, the engineer grows the silicon substrate, layer by layer, placing each phosphorus donor with atomic precision before growing the silicon layer on top. This more precise method is now preferred for scalable fabrication.

**In summary.** Since 2010, Si:P technology has progressed from the readout and addressability of small phosphorous donor clusters to demonstrating the anticipated long coherence time of a single donor in isotopically pure silicon, high-fidelity readout, single-qubit control, and violations of a Bell inequality using the electron and nuclear spin of a single phosphorus donor.

The original motivation of leveraging donor technology in the classical silicon industry remains strong. Although the other technologies are likely to achieve a large-scale machine earlier, donor-based quantum computers are an attractive option because they have the potential to be smaller and cheaper.

### Anyons

In quantum mechanics, many properties of individual particles can take only specified values: either whole numbers (for particles known as bosons, such as photons) or half numbers (for particles known as fermions, such as electrons). Remarkably, under some circumstances certain materials can behave as if they are holding particles that do not actually exist, known as quasiparticles. One postulated class of quasiparticle, called anyons, allows some properties to have values corresponding to fractions other than multiples of one half.[13] The original description of the first type of topological code showed a direct correspondence between this digital QEC mechanism and the physical equations that describe anyons. We use the terminology "anyonic quantum computers" to distinguish this model from the topological coding models already described.

> **SUPERCONDUCTING QUANTUM COMPUTERS WILL LIKELY BE THE FIRST TO OUTPERFORM CLASSICAL MACHINES ON RELEVANT QUANTUM PROBLEMS.**

Anyonic quantum computing has emerged as a highly complex model of quantum computation, which makes it difficult to review in a general computing magazine. There are already excellent summaries of both the technology's theoretical foundations and possible implementations.[13]

We assigned anyonic quantum computing to the fourth generation for two reasons. First, it tries to suppress errors using the fundamental physics of the system itself. Thus, rather than embedding complicated error-correction codes on top of standard two-level quantum systems (qubits), an anyonic quantum computer exhibits quantum excitations that are naturally protected from decoherence. The result could be systems with extremely low physical error rates, mitigating (or even eliminating) the need for active error correction. The second reason for categorizing anyonic quantum computers as fourth-generation technology is that no one has yet reliably demonstrated the existence of anyonic particles within engineered systems.

## SOFTWARE CONTROL

Topological coding models of error-corrected computation are software based. Research in software control is emerging as a dedicated subfield in the development of quantum technology.[14]

Enacting quantum algorithms is a function of switching on and off sections of the computer in accordance with the overlying algorithm, while error correction is a continuous process of extracting syndrome information and decoding it to determine where physical errors have occurred. A large-scale quantum computer will require extensive classical resources to operate (in contrast to any classical system that might use the quantum computer). These resources are divided into offline control and online control, the elements of which are shown in Figures 4 and 5.

### Offline control

Offline control is the compilation and optimization of fault-tolerant quantum

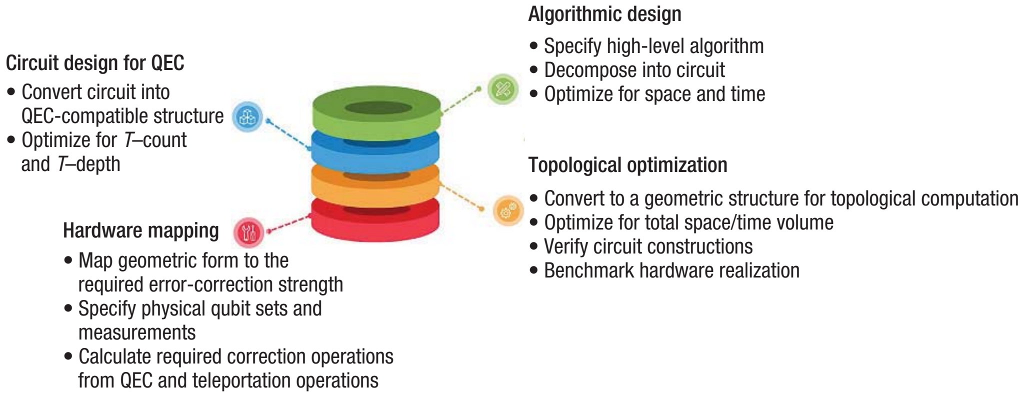

Circuit design for QEC
• Convert circuit into QEC-compatible structure
• Optimize for $T$–count and $T$–depth
Hardware mapping
• Map geometric form to the required error-correction strength
• Specify physical qubit sets and measurements
• Calculate required correction operations from QEC and teleportation operations
Algorithmic design
• Specify high-level algorithm
• Decompose into circuit
• Optimize for space and time
Topological optimization
• Convert to a geometric structure for topological computation
• Optimize for total space/time volume
• Verify circuit constructions
• Benchmark hardware realization

**FIGURE 4.** Offline design stack. There are multiple stages to compiling and optimizing a topological quantum circuit.

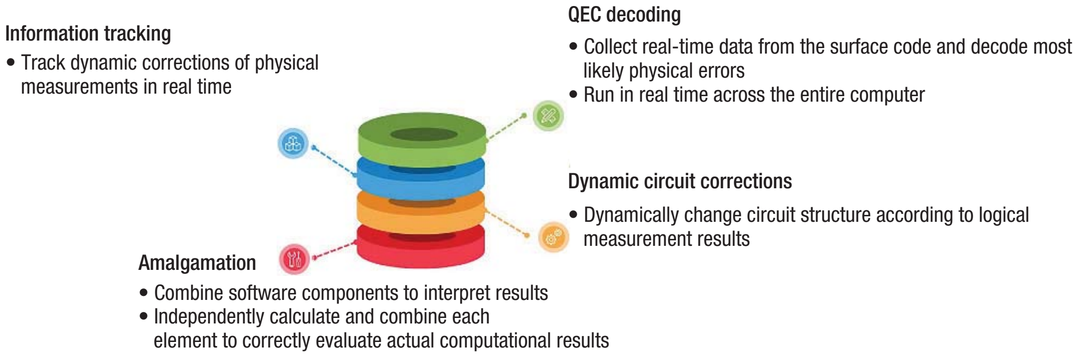

Information tracking
• Track dynamic corrections of physical measurements in real time
Amalgamation
• Combine software components to interpret results
• Independently calculate and combine each element to correctly evaluate actual computational results
QEC decoding
• Collect real-time data from the surface code and decode most likely physical errors
• Run in real time across the entire computer
Dynamic circuit corrections
• Dynamically change circuit structure according to logical measurement results

**FIGURE 5.** Online design stack. As with offline computation, there are multiple layers to online classical processing for a large-scale quantum computer. Unlike offline control, these elements must keep up with the physical clock speed of a quantum computer. Additional classical online processing is necessary so that architectures dependent on probabilistic gates, such as linear optics, can address heralded gate failures and lattice renormalization.

circuits before the computer is turned on. These software elements are needed to translate an abstract algorithm into gate sequences compatible with fault-tolerant error correction and to convert the gate lists into an appropriate control structure for the topological codes. At each stage of offline compilation, circuits and topological structures must be optimized for both physical qubits and computational time, and optimized structures must be verified against the desired computational specification.

### Online control

Online control is the set of classical software packages that run in tandem with the quantum computer. They primarily handle dynamic error decoding and the mapping of the compiled circuit onto physical control and signals to the hardware. Online control software will require extremely fast operation over a large dataset. The algorithms must be able to keep pace with the quantum hardware's physical clock rate, which for third-generation machines will be in the GHz range. They will also need to operate on qubit arrays of possibly billions of qubits. Consequently, the scaling properties of these algorithms are a serious concern and need further development.

A quantum computer cannot operate without these packages, and it is impossible to appropriately benchmark quantum algorithms without a fully developed compiler and software stack. Although some effort is focused on compiling and benchmarking topological quantum circuits and much more on higher-level software languages and circuit compilers, considerable work remains to optimize functional topological circuits to the level known to be theoretically possible and to accurately determine qubit counts and computational time for useful quantum algorithms.

## NETWORKS AND DISTRIBUTED APPLICATIONS

All seven of the technologies described above have technological limitations on the number of qubits that can be built in one chip or device—a number well below that required for applications such as Shor's factoring algorithm. The demand for high-capacity scalable

## ABOUT THE AUTHORS

**RODNEY VAN METER** is an associate professor of environment and information studies at Keio University Shonan Fujisawa Campus. His research interests include storage systems, networking, and post–Moore's law computer architecture. Van Meter received a PhD in computer science from Keio University. He is a member of ACM, IEEE, the Information Processing Society of Japan (IPSJ), and the American Association for the Advancement of Science (AAAS). Contact him at rdv@sfc.wide.ad.jp.

**SIMON J. DEVITT** is a senior research scientist at the RIKEN Center for Emergent Matter Science. His research interests include large-scale architectural designs for quantum computation and communications systems, and software compilation and optimization for topological quantum computing. Devitt received a PhD in physics from the University of Melbourne. Contact him at simon.devitt@riken.jp.

systems has forced most research groups working on large-scale systems to adopt a roadmap that includes multicomputers, groups of smaller computers connected through some form of system area network (SAN). Specific hardware platforms have been proposed, building on ion traps, quantum dots, or diamond, which offer good optical connections. For such systems to be practical, an ordinarily monolithic computation must be split into pieces for distributed computation.

Programming quantum multicomputers will require exploiting quantum teleportation and entanglement. Data can be teleported from node to node in the teledata programming style, or entanglement can be used to execute a two-qubit gate remotely in the telegate style.[15] The key constraint is to use internode entanglement efficiently because it is a scarce resource consumed during each such operation.

Metropolitan area and wide area networks are also under development, which enable distributed quantum applications in three categories: distributed numeric computation, cryptographic functions, and sensor or cybernetic services. Blind quantum computation allows client–server computation in which the server cannot determine the input data, algorithm used, or output data. Cryptographic functions include secret key generation, Byzantine agreement, and secret sharing. Sensor uses include high-precision interferometry and clock synchronization.

Over the past decade, researchers exploring various QEC implementation technologies have met the DiVincenzo criteria, including reaching the threshold at which applying error correction removes more errors than it introduces. In parallel, theorists have analyzed multicomputer architectures and developed in-depth topological methods for error correction. The process of combining these concepts with experimental work is just beginning.

The question remains: what will constrain the ability to build a quantum computing system as large as desired? The answer requires understanding what is meant by scalability. One broad, informal definition is:[15]

> *Above all, it must be possible, physically and economically, to grow the system through the region of interest. Addition of physical resources must raise the performance of the system by a useful amount (for all important metrics of performance, such as calculation speed or storage capacity), without excessive increases in negative features (e.g., failure probability).*

This definition implies that scalability is never indefinite in the real world. Systems that cost more than $1 billion or that require unavailable quantities of helium or other resources are theoretically scalable but will not be practical.

The quest for the smallest economically viable quantum computer is therefore entering a new phase: it is moving from theoretical scalability to practical application. At some point, a new article will appear in *Science* or *Nature* describing the results of quantum computing, not the latest quantum computer design—providing evidence that quantum computers have ceased *being* science and started *doing* science.

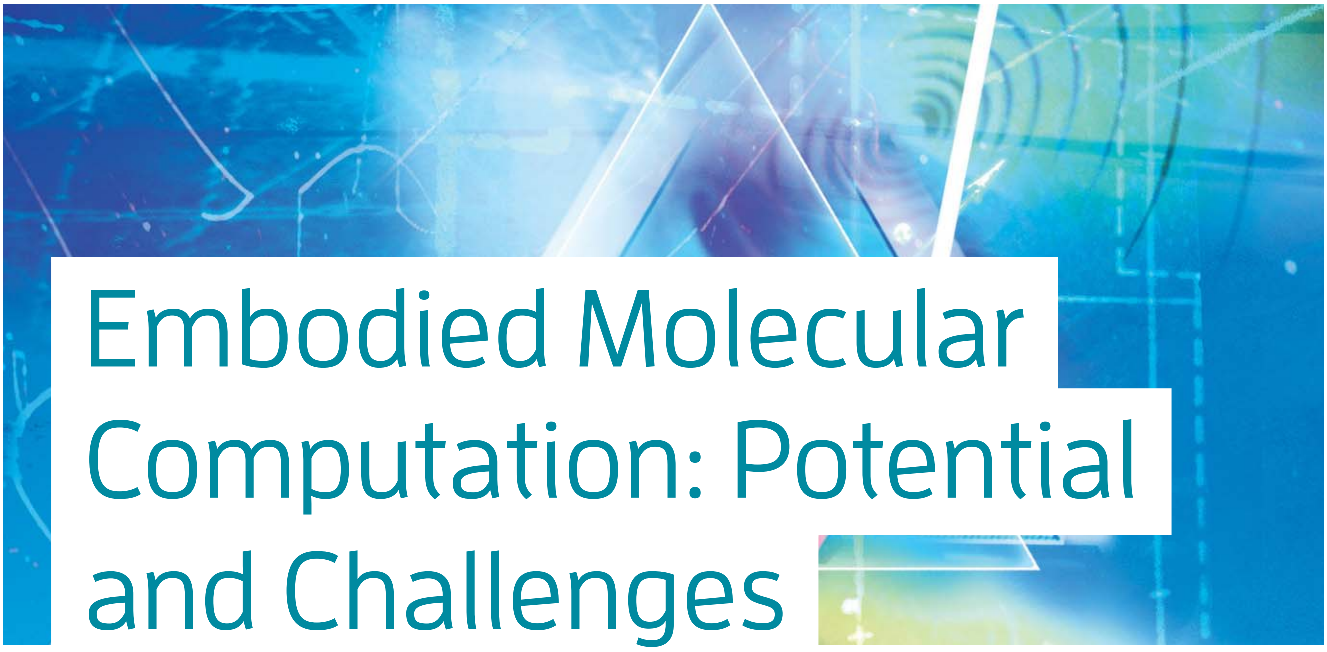
Although significant problems remain to be solved, the fundamental questions about how to build a quantum computer now have positive answers. It is clear that quantum computing is now moving from research to engineering and is poised to redefine real-world applications. ■

**ACKNOWLEDGMENTS**

This work was supported by the Japan Society for the Promotion of Science (JSPS) through KAKENHI (Grants-in-Aid for Scientific Research) Kiban B 16H02812 and the JSPS grant for challenging exploratory research. A draft of this article with more than 100 additional references is available at http://arxiv.org/abs/1605.06951.

## REFERENCES

1. T.D. Ladd et al., "Quantum Computers," *Nature*, vol. 464, 2010, pp. 45–53.
2. R. Van Meter, *Quantum Networking*, Wiley-ISTE, 2014.
3. A. Montanaro, "Quantum Algorithms: An Overview," *npj Quantum Information*, vol. 2, 2016, article no. 15023.
4. A.G. Fowler et al., "Surface Codes: Towards Practical Large-Scale Quantum Computation," *Physics Rev. A*, vol. 86, 2012, article no. 032324.
5. K.R. Brown, J. Kim, and C. Monroe, "Codesigning a Scalable Quantum Computer with Trapped Atomic Ions," arXiv preprint, 2016; arXiv:1602.02840.
6. J.M. Martinis, "Qubit Metrology for Building a Fault-Tolerant Quantum Computer," *npj Quantum Information*, vol. 1, 2015, article no. 15005.
7. D. Bonneau, J.W. Silverstone, and M.G. Thompson, "Silicon Photonics III: Systems and Applications," *Silicon Quantum Photonics*, L. Pavesi and D.J. Lockwood, eds., Springer, 2016, pp. 41–82.
8. E. Knill, R. LaFlamme, and G.J. Milburn, "A Scheme for Efficient Quantum Computation with Linear Optics," *Nature*, vol. 409, 2001, pp. 46–52.
9. A.D. Greentree, "Nanodiamonds in Fabry-Perot Cavities: A Route to Scalable Quantum Computing," *New J. Physics*, vol. 18, no. 2, 2016, article no. 021002.
10. F.A. Zwanenburg et al., "Silicon Quantum Electronics," *Modern Physics Rev.*, vol. 85, 2013, pp. 961–1019.
11. C.D. Hill et al., "A Surface Code Quantum Computer in Silicon," *Science Advances*, vol. 1, no. 9, 2015, article no. e1500707.
12. B.E. Kane, "A Silicon-Based Nuclear Spin Quantum Computer," *Nature*, vol. 393, 1998, pp. 133–137.
13. S. Das Sarma, M. Freedman, and C. Nayak, "Majorana Zero Modes and Topological Quantum Computation," *npj Quantum Information*, vol. 1, 2015, article no. 15001.
14. S.J. Devitt, "Classical Control of Large-Scale Quantum Computers," *Reversible Computation*, LCNS, vol. 8507, Springer, 2014, pp. 26–39.
15. R.D. Van Meter III, "Architecture of a Quantum Multicomputer Optimized for Shor's Factoring Algorithm," PhD dissertation, Dept. of Computer Science, Keio Univ., 2006; http://arxiv.org/abs/quant-ph/0607065.



# Embodied Molecular Computation: Potential and Challenges

**Victoria Coleman,** Potomac Institute for Policy Studies

*In its quest for more powerful computing paradigms and machines, DARPA's Defense Science Research Council looked at molecular computing, a compelling approach to unconventional computing that draws on information processing in physical, chemical, and biological systems.*

The Defense Science Research Council (DSRC), a group of prominent scientists, was created by DARPA in 1968. It helped the agency peer into the future for decades until it was disbanded in 2015. The DSRC fulfilled its mission by organizing mostly yearlong studies in which members and outside invited experts studied promising directions. The typical study entailed two or three workshops, and the results were briefed to DARPA during the DSRC annual retreat.

In 2010, the DSRC embarked on a study of impossible problems—those that conventional computing is unable to solve regardless of node size, speed, number of cores, or storage. In these workshops, we explored the potential of unconventional computing to solve computational challenges—such as climate modeling—better, cheaper, and faster than conventional computers. In the process, we found that approaches such as embodied computation, which leverages the physical properties of the compute fabric it executes on,[1] can indeed solve important problems such as massively parallel search in elegant and efficient ways. A large and growing body of work bears this out (www.ucnc2016.org; www.oldcitypublishing.com/journals/ijuc-home; http://molecular-programming.org). More interestingly, we realized that unconventional computing can drastically enlarge the boundaries of what is today considered the set of problems that can be solved computationally. For example, virus replication or drugs to arrest viral replication can be framed, specified, and computed as finite computer programs. This is embodied molecular computing. As well as allowing us to computationally solve problems not conventionally thought of as computable, molecular computing could potentially transform conventional computation itself by solving some of today's fundamental computing challenges: information density, parallelism, and energy efficiency.

## APPROACH

This series of DSRC workshops was motivated by the council's visit to the National Center for Atmospheric

Research in Boulder, Colorado, in the spring of 2008. We saw scientists striving to build better prediction models and execute them on acres of high-performance machines. The tools they use, Navier Stokes equations, have not changed in generations. All these acres of machines labor with traditional numerical models using traditional computing software and hardware and, overall, deliver poor accuracy and efficiency.

Our intuition was that tough computational problems like these could be solved better, cheaper, and faster by rethinking computing's fundamental models and processes. So we embarked on a series of workshops to explore this hunch. We interacted with many prominent and diverse researchers and scientists, ranging from logic philosophers to theoretical computer scientists, complexity theorists, mathematicians, physicists, biologists, and computer architects.

We debated approaches ranging from hypercomputation (or super Turing computation) to quantum and cellular computing to neural, molecular, and chemical computing. Although the hypercomputation discussions were especially fun (some involving armies of self-replicating robots proceeding at near light speed to the vicinity of a black hole and thus able to solve infinite problems such as the Goldbach conjecture[2]) and entailed novel theoretical formulations of computability, we ultimately chose to focus on one particularly compelling (and perhaps eventually practical) form of embodied computation: molecular computing. We felt that the advances in synthetic biology combined with the computer science community's experimental and theoretical work brought the state of the art near the tipping point for realizing the technology.

Molecular computing is the computation embodied in and performed by biological systems such as DNA molecules. Interesting analogies between computability and DNA offer insights into how biological processes can not only be harnessed for computation but also be comprehended as computational processes themselves. If true, this has profound consequences.

## EMBODIED COMPUTATION PRINCIPLES

In defining embodied molecular computation, it is important to distinguish between using DNA and other organic material as the compute fabric versus molding it to our existing silicon compute fabric paradigm. Arieh Aviram pioneered this latter approach in 1975, when he proposed substituting silicon transistors with single organic molecules. Similar approaches evolved over the past 40 years, all of them aiming to replace the fundamental unit of silicon compute, the transistor, with a single, organic molecule transistor. The most recent in the series of these efforts, the memristor, was invented in 2008 and is still being actively developed at HP Labs. Kevin Kelly and Cyrus Mody present a highly readable retrospective on the promise of replacing silicon with molecular components.[3] The insight driving this body of work is that building a computing element starting with one molecule and adding more along the way is an inherently easier way to create sophisticated, precise structures as opposed to starting with a large piece of material and etching it away to arrive at the component, which is how we build transistors today. Our point of departure here is different. Instead of crafting transistors one molecule at a time, we want to explore and exploit the physical and biochemical properties of DNA and other organic compounds to get computation "for free." In other words, our focus is on embodied molecular computation.

Bruce MacLennan states that embodied computational processes have the following characteristics:[1]

- They directly exploit physical processes for computational ends.
- They represent information and processes implicitly in the physics and other properties of the system and its environment.
- They affect computation by growth, assembly, development, transformation, reconfiguration, or disassembly of the physical system embodying the computation.

In this approach, the physical substrate performs many computations for free. Embodied computation is a physical or biological process that proceeds by interacting with other physical processes. A good example of embodied computation is diffusion. Diffusion occurs naturally in many fluids and can be harnessed for computational tasks such as broadcasting information or massively parallel search. Both tasks are computationally intensive in silicon but free in fluidic systems. A further example of embodied computation for free is Rajesh Ganapathy and his colleagues' work demonstrating how colloidal crystals can be used to conduct highly accurate simulations of epitaxial growth.[4]

The difference between Aviram's work and its successors and the work of MacLennan and Ganapathy and his colleagues is as profound as the difference between digital and analog computation. In discussing physical embodiment's role in the "grand challenge" of nonclassical computing, Susan Stepney and her colleagues write:[5]

> *Computation is physical; it is necessarily embodied in a device whose behaviour is guided by the laws of physics and cannot be completely captured by a closed mathematical model. This fact of embodiment is becoming ever more apparent as we push the bounds of those physical laws.*

However, this nonclassical approach to computing and the mathematical models required to systematically and predictably harness embodied computation must be delicately balanced. I discuss this balance next, starting with the Turing machine, the fundamental concept of computability.

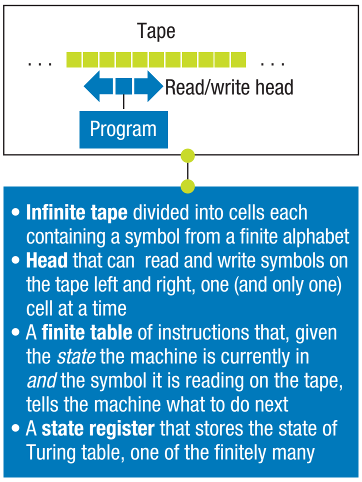
## THE TURING MACHINE AND MOLECULAR COMPUTATION

In 1936, Alan Turing began a rigorous study of the notion of computability.[6] This purely theoretical work preceded the advent of actual computers by about a decade and led to some of the major mathematical results of the 20th century. Turing invented the Turing machine, a toy computer that was intended to be a conceptual tool for mathematical investigation but has turned out to be universal—it can be programmed to compute anything that is computable. Conversely, if the Turing machine cannot be programmed to solve a particular problem, then that problem cannot be computed by any machine we can hope to build, no matter how complex or powerful. The Turing machine is the central abstraction of computing today. As Figure 1 shows, it is a simple device consisting of an infinite tape, head, finite table, and state register. (Note that although the Turing machine is not practical for implementing computation, anyone wishing to use it for practical computation would use a finite Turing machine; that is, one with a finite tape.)

Let us now tease out molecular computation's analogies with the Turing machine. Molecular computation is a form of embodied computation. Many researchers have experimented with building the basic hardware building blocks, logic gates, and circuits using technologies such as DNA strand displacement[7] or enzyme gates.[8] Though certainly interesting, the question is whether these circuits can be used to solve real problems or whether they are simply a curiosity. From a computing perspective, the first question we must ask is whether molecular computation is Turing complete. In other words, can we build a molecular computer that can solve any problem that a Turing machine can? Assuming the answer is yes, the second question concerns the right computing primitives. Are they logic gates, ones and zeros, or something altogether different? Molecular computers consisting of simple circuits constructed out of DNA-enabled gates have been built[9] and used to solve toy problems such as tic-tac-toe (see Figure 2). But the real question is whether we can build a molecular Turing machine capable of general-purpose computation. And further, should such a machine be possible, how would we program it? We next look at one attempt to do just this: Len Adleman's 1994 DNA computer.[10]

## EMBODIED COMPUTATION WITH DNA

Adleman's pioneering work not only offers an elegant solution to the nondeterministic polynomial time–complete Hamiltonian path problem (given a connected directed graph, determine whether a path exists that begins at the start node and finishes at the end node, passing through all remaining nodes exactly once) but also provides key clues into embodied computation with DNA.[10]



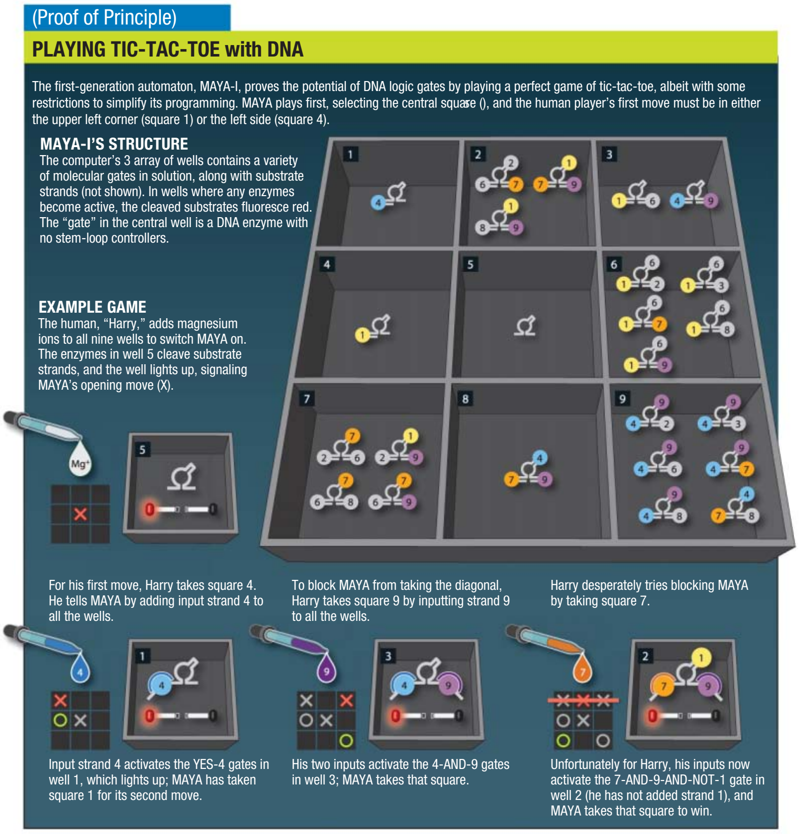
- **Infinite tape** divided into cells each containing a symbol from a finite alphabet
- **Head** that can read and write symbols on the tape left and right, one (and only one) cell at a time
- A **finite table** of instructions that, given the *state* the machine is currently in *and* the symbol it is reading on the tape, tells the machine what to do next
- A **state register** that stores the state of Turing table, one of the finitely many

**FIGURE 1.** The Turing machine.

Adleman used a simple algorithm for a graph with *n* vertices:

1. Generate a set of random paths (or "flights") through the graph.
2. For each path in the set:
   a. Check whether that path begins at the start vertex ("city of origin") and finishes at the end vertex ("destination city"). If not, remove that path from the set.
   b. Check whether that path passes through exactly *n* vertices ("cities"). If not, remove that path from the set.
   c. For each vertex, check whether that path passes through that vertex. If not, remove that path from the set.
3. If the set is not empty, report that a Hamiltonian path exists. If it is empty, report that a Hamiltonian path does not exist.

DNA represents information using the four-character genetic alphabet (A


(Proof of Principle)

PLAYING TIC-TAC-TOE with DNA

The first-generation automaton, MAYA-I, proves the potential of DNA logic gates by playing a perfect game of tic-tac-toe, albeit with some restrictions to simplify its programming. MAYA plays first, selecting the central square (), and the human player's first move must be in either the upper left corner (square 1) or the left side (square 4).

MAYA-I'S STRUCTURE

The computer's 3 array of wells contains a variety of molecular gates in solution, along with substrate strands (not shown). In wells where any enzymes become active, the cleaved substrates fluoresce red. The "gate" in the central well is a DNA enzyme with no stem-loop controllers.

EXAMPLE GAME

The human, "Harry," adds magnesium ions to all nine wells to switch MAYA on. The enzymes in well 5 cleave substrate strands, and the well lights up, signaling MAYA's opening move (X).

For his first move, Harry takes square 4. He tells MAYA by adding input strand 4 to all the wells.

Input strand 4 activates the YES-4 gates in well 1, which lights up; MAYA has taken square 1 for its second move.

To block MAYA from taking the diagonal, Harry takes square 9 by inputting strand 9 to all the wells.

His two inputs activate the 4-AND-9 gates in well 3; MAYA takes that square.

Harry desperately tries blocking MAYA by taking square 7.

Unfortunately for Harry, his inputs now activate the 7-AND-9-AND-NOT-1 gate in well 2 (he has not added strand 1), and MAYA takes that square to win.

**FIGURE 2.** MAYA I, a DNA computer. (Source: J. Macdonald, D. Stefanovi, and M.N. Stojanovic, "DNA Computers for Work and Play," *Scientific American*, vol. 299, 2008, pp. 84–91; used with permission.)

[adenine], G [guanine], C [cytosine], and T [thymine]), rather than the binary alphabet (one and zero) used by traditional computers (www.britannica.com/technology/DNA-computing). Adleman assigned a random DNA letter sequence to each city. Then he assigned a flight number to each nonstop flight between two cities by concatenating the second half of the sequence representing the city of origin with the first half of the sequence representing the city of destination. He then synthesized the DNA city names and flight numbers. He ordered the DNA sequences from an outside lab. Once the DNA sequences arrived by mail, he put a pinch (about 1,014 molecules) of each in a test tube; added water, ligase, and salt; and waited about a second for the result. He then used the polymerase chain reaction to weed out molecules that did not both begin with the start city and terminate with the last city (step 2a). Gel electrophoresis identified and discarded all molecules with the wrong length (step 2b). Finally, affinity separation removed molecules whose paths did not pass through all the intermediate cities (step 2c). Any remaining molecules encoded the Hamiltonian path (step 3).

This was an inspiring result. However, and more importantly, Adleman observed a clear connection between this kind of computation and general computability. Given a Watson-Crick strand, DNA polymerase, an enzyme responsible for synthesizing DNA molecules from deoxyribonucleotides and hence DNA replication, produces a complementary strand in which C → G, G → C, A → T, and T → A. DNA polymerase hops onto a strand of DNA and slides along it, reading each base it passes and writing its complement onto a new growing DNA strand. The Turing machine has a finite control element that moves along the machine's input tape reading data while simultaneously moving along its output tape writing data. The similarity between DNA polymerase and the Turing machine finite control element could hardly be more obvious.

Unfortunately, as Adleman observed, while DNA polymerase can make Watson-Crick complements, it is unlikely that similar enzymes have evolved in nature for solving problems such as factoring numbers. However, today we can synthesize large numbers of custom molecules quickly and cheaply. So the question we must answer is whether we can design an enzyme or define some other biological process that can inherently solve the problem of interest. In other words, can we compile a problem into an enzyme versus a collection of ones and zeros? Andrew Phillips and Luca Cardelli present a compelling approach to this problem: a programming language for designing and simulating DNA circuits in which strand displacement is the main computational mechanism.[11] The language can be compiled to nucleotide sequences and used to model various circuits. More recently, Alec Nielsen and his colleagues described Cello, a Verilog-based programming language that, given a circuit description in the form of a truth table, compiles it into plasmid DNA sequences that genetically implement the circuit's desired logic function.[12] The DNA sequences are then injected in living cells where the circuit runs. These efforts represent important first steps toward novel implementation strategies for DNA computing.

Adleman's simple but powerful DNA computing toolkit suggests that although logic gates made of transistors might be the right primitives for silicon compute, they might not be appropriate for DNA substrates. Polymerases create complementary copies of the template on the primer (copying). Ligases bind molecules together (concatenation). Nucleases cut nucleic acids (partitioning). Gel electrophoresis separates DNA strands by length (sorting). Why would we design and implement a sorting algorithm if the substrate offers the function for free?

In his seminal work on systems biology machines,[13] Cardelli offers further insights into molecular computing toolkits. He shows how the four classes of macromolecules map neatly into common data structures, possibly forming additional biochemical toolkits:

- RNA is analogous to a linked list, and DNA is analogous to a doubly linked list;
- proteins map into records or objects that are active and stateful;
- lipids (membranes) map into containers with interfaces (modules with parameters); and
- carbohydrates map into trees.

This is embodied computation's challenge and opportunity: to take advantage of what the compute fabric naturally offers versus hammering our existing models in it, or, in Christof Teuscher and Peter Dittrich's words, torturing versus respecting the medium. In this case, taking advantage of the computational elements offered by DNA intrinsically versus building simple silicon computing[14] elements like logic gates out of arguably one of the most complex and information-rich materials known to man.

How to replicate Adleman's success in solving the Hamiltonian path with DNA molecules interacting in a gel computer in a practical, systematic, predictable, and reliable way while respecting this inherently stochastic medium is one of the most interesting questions in the theory and practice of computing today. Adleman built a fascinating computer. But can we build a general model that can be used to analyze the computations it performs, for example, complexity, termination, and convergence? Cardelli's work on abstract machines of systems biology offers a good starting point for thinking about this problem.[13]

## ABSTRACT MACHINES OF SYSTEMS BIOLOGY

The compute's proximity to the underlying fabric, while very powerful, does not absolve us of the need for embodied computation theories that allow us to analyze and predict the computation's properties. In recent years, our understanding of biology has been greatly enhanced by the systems approach, in which discovering how a cell works requires understanding not only the biological system's structure but also the biological processes it relies on and participates in—essentially understanding the cell's function in terms of information processing instead of chemistry. Cardelli showed how the macromolecule classes can be represented as abstract machines,[13] with each such machine consisting of a collection of discrete states and a collection of operations or events that cause discrete transitions between these states. This is a profound observation: to understand the interactions between multiple machines is to understand how the cell works. Programming interactions between these machines enables manipulation of the cell in well-defined, predictable ways to affect therapeutics. These interactions can thus be understood as the result of symbolic (abstract) computation, which effectively means that, for example, developing a drug to halt viral replication is equivalent to writing a program.

As shown in Figure 3, Cardelli's model has three key machines:

- The *gene machine* performs information-processing tasks within the cell, regulating all other activities, including assembly and maintenance of other machines and the copying of itself.
- The *protein machine* performs all mechanical and metabolic tasks and some signal processing.
- The *membrane machine* separates different biochemical environments and operates dynamically to transport substances via complex, multistep processes.

To understand how the cell works is to understand how these machines interact, albeit in vastly different time and size scales.

Each of these machines operates with its own instruction set. Let us consider the protein machine as an example[13] for the description of the gene and membrane machine instruction sets. Cardelli characterizes its operation as follows (see Figure 4). Each protein is modeled as a collection of sites and switches that, at any given time, are either available or unavailable. Proteins can join at matching sites to form bigger complexes. The sites and switches' availability in a complex is the state of that complex. A system is a multiset of disjoint complexes, each in

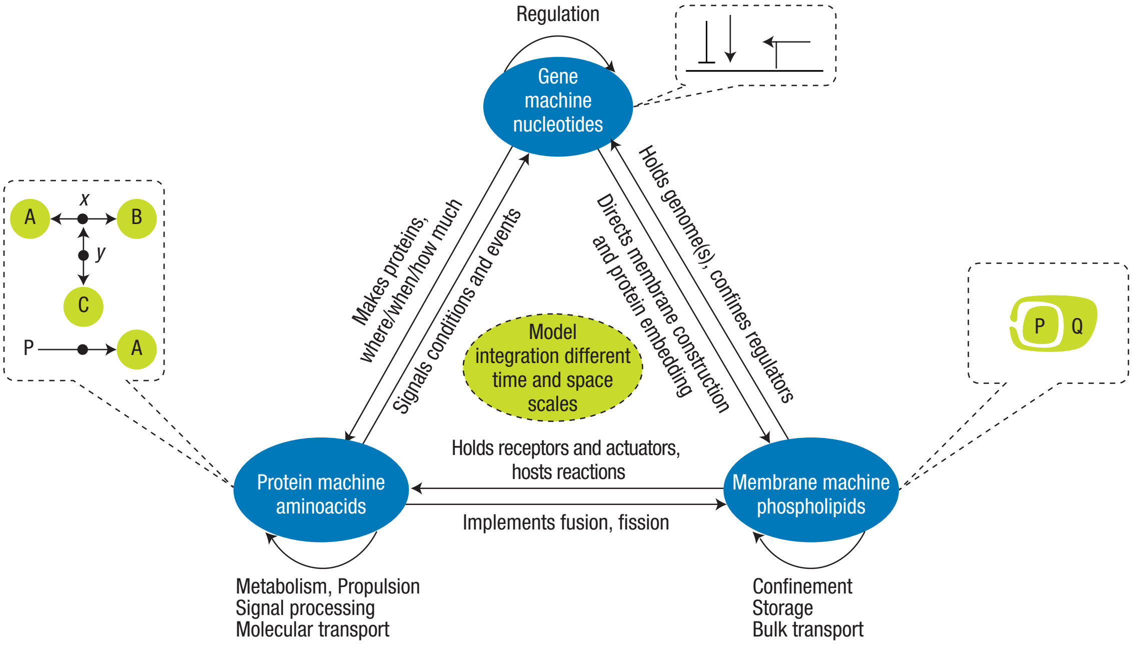

Regulation
Gene machine nucleotides
A
x
B
y
C
P
A
P
Q
Makes proteins, where/when/how much
Signals conditions and events
Holds genome(s), confines regulators
Directs membrane construction and protein embedding
Model integration different time and space scales
Protein machine aminoacids
Holds receptors and actuators, hosts reactions
Implements fusion, fission
Membrane machine phospholipids
Metabolism, Propulsion
Signal processing
Molecular transport
Confinement
Storage
Bulk transport

**FIGURE 3.** Cardelli's abstract machines of biology. (Source: L. Cardelli, "Abstract Machines of Systems Biology," *Trans. Computational Systems Biology III*, LNCS 3737, C. Priami et al., eds., Springer, 2005, pp. 145–168; used with permission.)

a given state. The protein machine has two operations: an available switch on a complex can be turned on or off, resulting in a new state in which a new collection of switches and sites is available; and two protein complexes can combine at available sites or one complex can be split into two, resulting in a new state in which a new collection of switches and sites is available.

How does a protein find its binding partner in the cell? This is a hard computational problem that cells perform with utmost ease. How do we capture that capability in terms of the protein machine instruction set? Switching and binding are driven by other proteins. There are tens of thousands of proteins in a cell, so the protein machine has tens of thousands of primitive instructions, each with a specific way of acting on other proteins. For each cellular subsystem to be modeled, we need to list the proteins involved alongside the ways that the proteins interact with one another in terms of switching and binding.

The three machines continuously interact. They are, in fact, highly interdependent. Genes instruct the production of proteins and membranes, and direct the embedding of proteins within membranes. Some proteins act as messengers between genes, and others perform various gating and signaling tasks when embedded in a membrane. Membranes confine cellular materials and bear proteins on their surfaces.

The execution of these processes can be modeled as programs written in the instruction set of each machine, and the interactions between machines can be modeled as interdependent, concurrent processes. In classical computing, especially in parallel/concurrent computation, comprehending the complexity of these interactions necessitated many specialized models, including process algebras. Process algebras are a tool for the high-level description of interactions, communications, and synchronizations among a collection of processes. They provide algebraic laws that allow process descriptions to be manipulated and analyzed and that permit automated formal reasoning about equivalence and other relationships and interactions among processes. Leading examples of process algebras include the π calculus enriched with stochastic semantics, the Ambient calculus that extends the π calculus with compartments, and the Brane calculus that embeds biological invariants.[13] These algebras lend themselves nicely to the modeling and analysis of gene, protein, and membrane machine interactions, thus yielding a powerful theoretical framework in which these machines and, by extension, the semantics of molecular computations carried out by Adleman's gel computer can be comprehended and defined. This, then, paves the way for models that can be used to predict the computation's properties, to systematically derive programs from, to argue about their correctness, and more.

## IF BIOLOGY IS COMPUTATION, THEN BIOLOGICAL PROCESSES ARE PROGRAMS

Cardelli's machines allow us to comprehend cell function in terms of information processing rather than chemistry. Stated differently, they allow us to discover the algorithms of biology. In fact, they enable us to go even further and design algorithms for biology. Cardelli presents an elegant example of this in his description

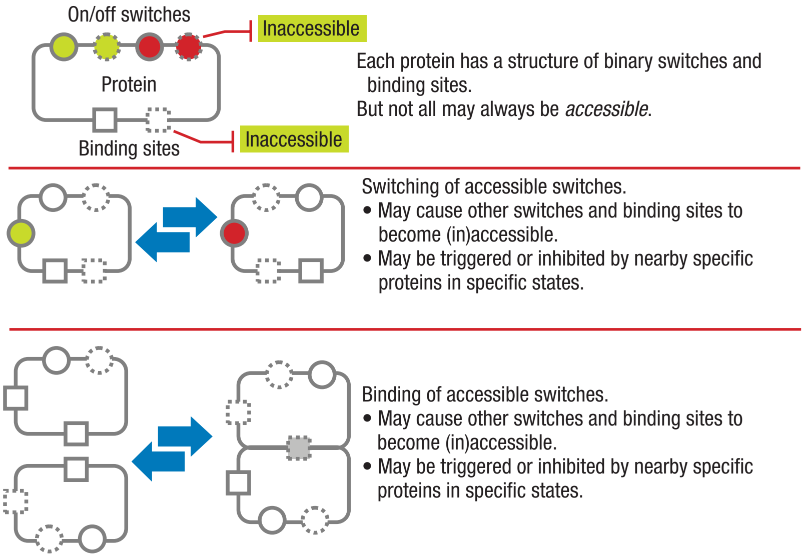

On/off switches
Inaccessible
Protein
Binding sites
Inaccessible
Each protein has a structure of binary switches and binding sites.
But not all may always be *accessible*.
Switching of accessible switches.
• May cause other switches and binding sites to become (in)accessible.
• May be triggered or inhibited by nearby specific proteins in specific states.
Binding of accessible switches.
• May cause other switches and binding sites to become (in)accessible.
• May be triggered or inhibited by nearby specific proteins in specific states.

**FIGURE 4.** Cardelli's protein machine instruction set. (Source: L. Cardelli, "Abstract Machines of Systems Biology," *Trans. Computational Systems Biology III*, LNCS 3737, C. Priami et al., eds., Springer, 2005, pp. 145–168; used with permission.)

of the Semliki Forest virus replication. Figure 5 shows the algorithm that the virus uses to replicate itself in terms of the protein, gene, and membrane machine instruction sets. Cardelli models each stage of the replication process, infection, replication, and progeny in the Brane calculus. This theoretical model lends itself to mathematically sound analysis and prediction, meaning we can predict as well as reason about its behavior. If we could, in turn, implement this model in a molecular programming language, itself implementing the semantics of the three machines, we could also program the virus replication in "mechanical" ways and, crucially, modify its algorithm and behavior. This would be virus replication expressed as symbolic computation.

The implications of this are fundamental: a drug to stop replication would be "just" a program—moving therapeutics and drug discovery from a purely experimental discipline to a symbolic one. Not only can embodied molecular computing help us solve problems like Adleman's Hamiltonian path, it can also help us solve biochemical problems using symbolic computation. Why do I emphasize symbolic? Accurate simulations of physical viral behaviors are extremely complex and, in most cases, impractical. Symbolic computation, and the abstraction it affords, brings analysis of the most complex biochemical processes into the realm of the possible.

Hence, we could expand the universe of computationally solvable problems per the promise of embodied computation. We could not only solve known computing problems in fundamentally more efficient ways but also approach problems that, today, are solved experimentally by instead using computational principles, theories, and tools. To fulfill this promise, we need a universal molecular computer.

## IS A UNIVERSAL MOLECULAR COMPUTER PLAUSIBLE?

What does DNA bring to the embodied computation toolkit? First, it stores information, albeit in strings of As, Ts, Gs, and Cs, instead of ones and zeros. Second, it offers powerful embodied computation primitives:

- DNA polymerase binds onto a strand of DNA and produces its Watson-Crick complement, thus offering a free copying operation;
- ligases bind molecules together, thus offering free concatenation; and
- nucleases cut nucleic acids, so segmentation operations are available in the fabric.

Is this enough to build a general-purpose computer? As Turing showed with his all-powerful toy machine, only two things are necessary to build a computer: a method for storing information (DNA is a great way to store information) and a few simple operations for acting on that information (enzymes like polymerase operate on that information). So, in theory, we can be confident that a general-purpose molecular computer is feasible.

In addition, molecular assembly languages have been designed and implemented that can translate DNA molecule operations into chemical reactions. Routine DNA synthesis allows for the creation of large numbers of molecules from DNA sequences. Certain cell constituents map into key data structures such as lists and trees. Process algebras are very well-suited to capturing molecular computation's operational semantics; thus, the promise of a molecular logic theory that allows the building of predictable programs is very real. Gel-based DNA computers have been used to solve Hamiltonian problems as well as reasonably complex satisfiability (SAT) problems (for example, 6-variable, 11-clause SAT).

These tantalizing analogies offer a great deal of optimism. But several formidable technical challenges must be solved before we can build and use a universal molecular computer:

- First and foremost, we must define the problem sets we hope

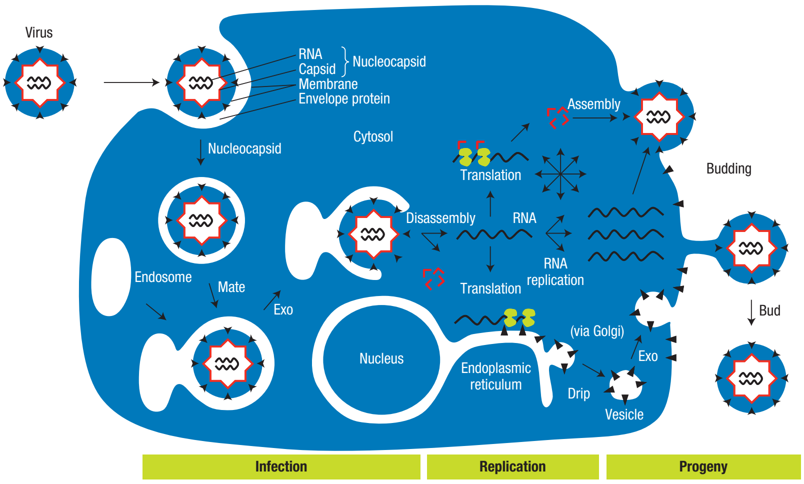


**FIGURE 5.** Replication algorithm of the Semliki Forest virus. (Source: L. Cardelli, "Abstract Machines of Systems Biology," *Trans. Computational Systems Biology III*, LNCS 3737, C. Priami et al., eds., Springer, 2005, pp. 145–168; used with permission.)

to solve with embodied molecular computation. Some might be classical computing problems such as Hamiltonians and SAT. Others might be more recent, including machine learning and its variations such as deep learning. Yet others might be problems that we do not think of today as computational, such as drug discovery, drug delivery, and nanotherapeutics.

- The Turing machine, powerful as it is in defining what is computable, is far from a practical computing device. Our molecular programs need to be compiled to effect the computation. What does a high-level molecular programming language look like?
- How do we bridge the vastly different rates at which different aspects of the computation proceed? In other words, how do we synchronize multiple, overlapping molecular clocks?
- How do we define molecular assembly languages that can translate our intent to programmable, controllable, and predictable biochemical reactions?
- With a molecular computing process occurring in a test tube, there is nowhere to attach crocodile clips to read the result of the computation. How do we implement I/O mechanisms?
- What theoretical frameworks are needed to model embodied molecular computation, and what are the complexity limits of the problems we hope to solve?
- How do we gain confidence in both the molecular computer and our molecular software? How do we validate our code? What robustness and fault tolerance mechanisms are feasible?

## A TRANSFORMATIONAL PAYOFF

What if we succeed in building a universal molecular computer? We would have liquid computers and programmable drugs that could be ingested by a patient to cure all manner of diseases in highly specialized, noninvasive, targeted ways. We could build molecular robots for payload transport or build memories. And we could solve the fundamental challenges of practical computing today:

- *Unprecedented information density and persistence.* One cubic millimeter of DNA can store an exabyte of information with an observed half-life of 500 years.[15] The digital universe (all digital data worldwide) is forecast to grow to more than 16 zettabytes in 2017. This exponential growth rate easily exceeds our ability to store it, even with the predicted improvements in storage technologies.
- *Massive parallelism.* In Adleman's study, 1 teaspoon of DNA solution can carry out 1,014 computations simultaneously. Our ability to leverage the parallelism offered by multicore machines continues to be limited by the complexities of concurrent programming.
- *Extreme energy efficiency.* 1 joule is sufficient for $2 \times 10^{19}$ operations against a theoretical maximum of 34 x $10^{19}$. In contrast, the most power-efficient supercomputer topping the Green500 list in 2015, RIKEN's Shoubu in Japan, delivers 7.03 Gflops/W.

Embodied molecular computation hints tantalizingly at the possibility of machines that vastly exceed conventional computing capabilities in storage, performance, and energy efficiency. Moreover, it promises to bring a class of fundamental problems with strictly empirical solutions, such as drug discovery, into the symbolic computation realm. Pioneering work by Adleman, Cardelli, Erik Winfree, Paul Rothemund, and many others has served to build our confidence in the approach. But formidable obstacles remain, and our ability to systematically and predictably harness the embodied molecular computation's potential is, at present, a vision only. In addition to supporting expeditionary research such as that of the National Science Foundation–funded Molecular Programming Project, we must accelerate the pace and growth of the community of computer scientists, mathematicians, biochemists, biologists, chemists, physicists, and others who have united to build the key enabling technologies of molecular languages and compilers and to solve other challenges entailed by the theory and fabric of molecular computation. Catalyzing this community will require a concerted effort by funding agencies as well as the research community to build the research agenda, choose the most promising enquiry avenues, and focus the endeavor on a set of challenge problems.

We cannot afford to ignore molecular computation's potential. With the last decade's advances in the theory of molecular computation and synthetic biology, the opportunity in front of us is obvious.

## ABOUT THE AUTHOR

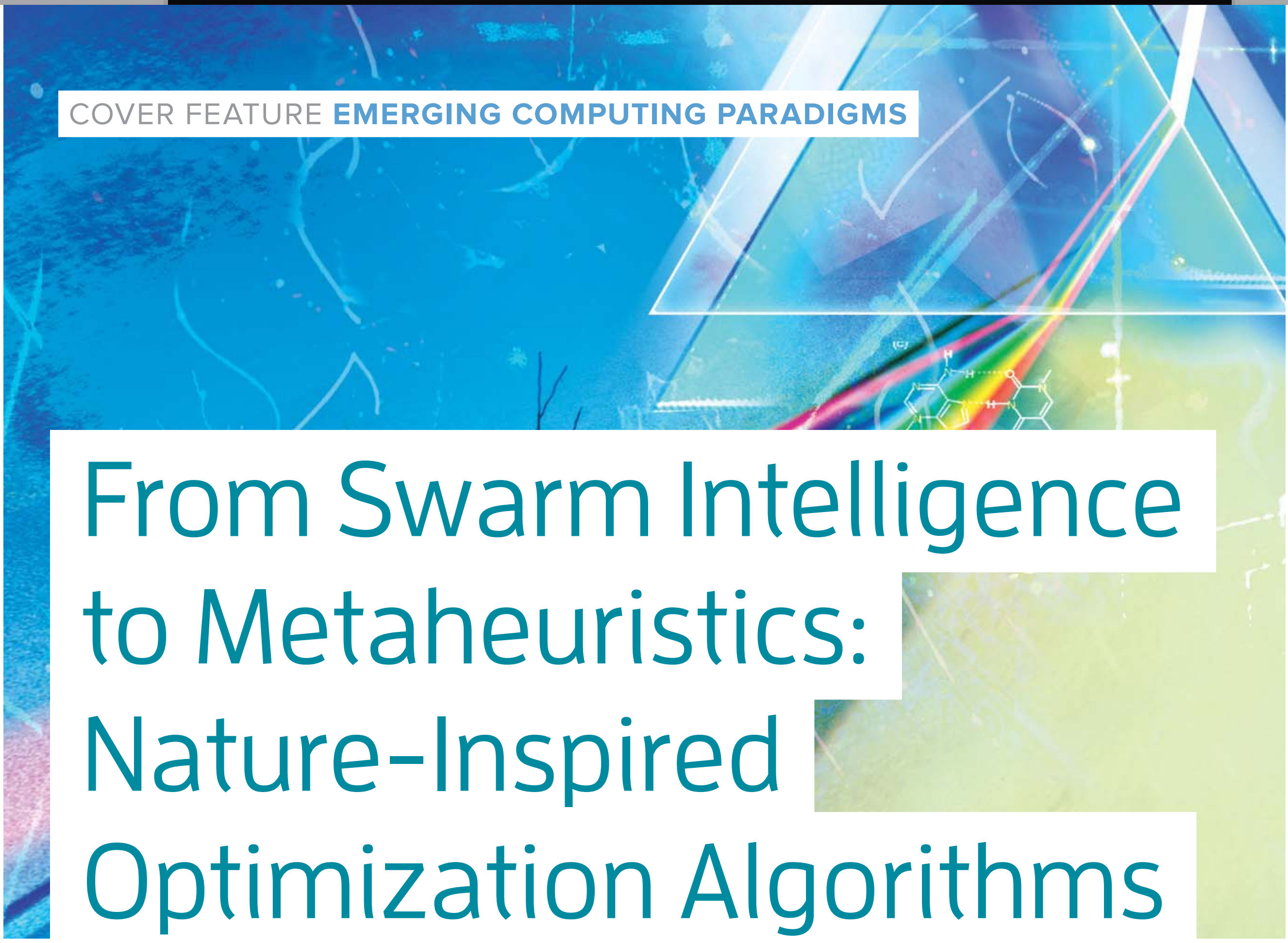
**VICTORIA COLEMAN** is a Senior Fellow at the Potomac Institute for Policy Studies. She is a member of the Defense Science Board, Lockheed Martin's Technology Advisory Group, and the advisory board of Santa Clara University's Department of Computer Engineering. Coleman previously served on DARPA's Information Science and Technology advisory group and Defense Sciences Research Council. She received a BSc in electronic computer systems and an MSc in computer-aided logic design from the University of Salford, and a PhD in computer science from the University of Manchester. Coleman holds four patents and has authored more than 60 articles and books. Contact her at vcoleman@potomacinstitute.org.

### ACKNOWLEDGMENTS

I am truly grateful to my DSRC colleagues who participated and animated the study that gave rise to this article: Howard Davidson, Craig Heller, Sandip Tiwari, Ed Yu, and George Whitesides. The following DARPA colleagues supported us and participated in the study: Todd Hylton, Gil Pratt, Ben Mann, Mita Desai, Mike Fritze, Duane Ashton, and Mark Rosker. And, of course, the study would not have been possible had it not been for the participation and contribution of our guest speakers: Bob Colwell, Lotfi Zadeh, Kwabena Boahen, Jonathan Mills, Itai Cohen, Son Dao, Martin Rinard, Michael Wehner, Jonathan Hiller, Nachum Dershowitz, Grisha Mints, Herbert Edelsbrunner, Christos Papadimitriou, Bruce MacLennan, Darko Stefanovic, David Meyer, Victor Eliashberg, and Martin Davis. Finally, I would like to thank the editors and anonymous referees for their comments that significantly helped improve the clarity and flow of ideas in this article.

### REFERENCES

1. B.J. MacLennan, "Aspects of Embodied Computing: Toward a Reunification of the Physcial and the Formal," tech. report UT-CS-08-610, Dept. of Electrical Eng. and Computer Science, Univ. of Tennessee, 2008.
2. H. Andréka, I. Németi, and P. Németi, "General Relativistic Hypercomputing and Foundation of Mathematics," *Natural Computing*, vol. 8, no. 3, 2009, pp. 499–516.
3. K.F. Kelly and C.C.M. Mody, "Whatever Happened to the Molecular Computer?," *IEEE Spectrum*, 25 Sept. 2015; http://spectrum.ieee.org/biomedical/devices/whatever-happened-to-the-molecular-computer.
4. R. Ganapathy et al., "Direct Measurements of Island Growth and Step-Edge Barriers in Colloidal Epitaxy," *Science*, vol. 327, no. 5964, 2010, pp. 445–448.
5. S. Stepney, "Journeys in Non-classical Computation," *Grand Challenges in Computing Research*, T. Hoare and R. Milner, eds., British Computer Society, 2004, pp. 29–32.
6. A.M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proc. London Mathematical Society*, series 2, no. 42, 1936, 230–236; correction no. 43, 1937, pp. 544–546.
7. L. Qian and E. Winfree, "Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades," *Science*, vol. 332, no. 6034, 2011, pp. 1196–1201.
8. J. Zhou et al., "Enzyme-Based NAND and NOR Logic Gates with Modular Design," *J. Physical Chemistry*, vol. 113, no. 49, 2009, pp. 16065–16070.
9. J. Macdonald, D. Stefanovic, and M.N. Stojanovic, "DNA Computers for Work and Play," *Scientific American*, vol. 299, 2008, pp. 84–91.
10. L. Adleman, "Molecular Computation of Solutions To Combinatorial Problems," *Science*, vol. 266, no. 1587, 1994, pp. 1021–1024.
11. A. Phillips and L. Cardelli, "A Programming Language for Composable DNA Circuits," *J. Royal Society Interface*, 2009; doi:10.1098/rsif.2009.0072.focus.
12. A.K. Nielsen et al., "Genetic Circuit Design Automation," *Science*, vol. 352, no. 6281, 2016; doi:10.1126/science.aac7341.
13. L. Cardelli, "Abstract Machines of Systems Biology," *Trans. Computational Systems Biology III*, LNCS 3737, C. Priami et al., eds., Springer, 2005, pp. 145–168.
14. A. Adamatzky et al., eds., *Unconventional Computing 2007*, Luniver Press, 2007.
15. J. Bornholt et al., "DNA-Based Archival Storage System," *Proc. 21st Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 16), 2016, pp. 637–649.



COVER FEATURE **EMERGING COMPUTING PARADIGMS**

# From Swarm Intelligence to Metaheuristics: Nature-Inspired Optimization Algorithms

**Xin-She Yang,** Middlesex University
**Suash Deb,** IT and Educational Consultant
**Simon Fong,** University of Macau
**Xingshi He,** Xi'an Polytechnic University
**Yu-Xin Zhao,** Harbin Engineering University

*Nature has provided rich models for computational problem solving, including optimizations based on the swarm intelligence exhibited by fireflies, bats, and ants. These models can stimulate computer scientists to think nontraditionally in creating tools to address application design challenges.*

Most if not all engineering tasks involve decisions about the product, service, and system design, which are related in some way to optimizing time and resources as well as achieving balance between maximizing performance, profit, sustainability, quality, safety, and efficiency and minimizing cost, energy consumption, defects, and environmental impact. As the sidebar "Elements of an Optimization Problem" describes, many of these design problems have multiple objectives bound by highly complex constraints. The traditional approach of specializing design method to problem type does not fit well with complex, nonlinear problems, such as multicriteria engineering designs and multiple complex feature extraction in big data.

This lack of suitability has motivated interest in more novel optimization approaches. One emerging trend is to combine heuristic search with multiagent systems to solve real-world business and engineering problems.[1] Such a combination has accuracy, efficiency, and performance advantages over specialized methods. For example, it can be used to more efficiently and accurately

0018-9162/16/$33.00 © 2016 IEEE

## ELEMENTS OF AN OPTIMIZATION PROBLEM

A typical optimization problem has a main design objective such as cost or energy efficiency, but is usually subject to many design constraints, which necessitates two optimization stages: problem formulation to define and prioritize constraints, and solution search to solve the problem with the optimal method.

### PROBLEM FORMULATION

To illustrate problem formulation in optimization, consider the hunt for a missing black box from a trans-Atlantic flight. The search's design constraints—budget, ocean currents, weather, and time—will be contradictory. Weather can cause delays, which might increase cost and work against the goal of finding the box within a certain time. There is also a stringent time constraint because the box's battery has a relatively short life. Thus, problem formulation must consider certain tradeoffs among cost, time, and the probability of finding the box, and these tradeoffs are subject to changes in various other factors, such as resources and location accessibility.

### SOLUTION SEARCH

Once an optimization problem is properly formulated, the main task is to find its optimal solution. Information about the box's possible location might come from the flight's last known position, and the search might start in that region. However, currents could have carried the box to another region, making the initial search quasirandom. A single agent hunting every possible square inch of the ocean region would be exhaustive but hardly practical. An alternative is to have a group search, in which members share information. The shared knowledge leads to a collective intelligence that helps narrow the search space.

In reality, most search processes are somewhere between random and focused; searching is not purely random even when there are no clues about where to look. As in the black-box example, the most likely scenario is a quasirandom search that seeks hints. It starts with a plausible location and then moves to another plausible location in the neighborhood and so on. Hints are exchanged with others in the search group. Searchers who are industrious and apply their experience to uncover more possible solutions might be rewarded or otherwise encouraged. Likewise, lazy or less motivated searchers might be dismissed. In this way, the selection of searchers and solutions will gradually improve the probability of finding the black box. This process of refined collective intelligence leading to a solution is the essence of contemporary metaheuristic optimization algorithms based on swarm intelligence.

optimize or tune parameters in artificial neural networks, which are essential to many AI tasks.

One class of novel optimization algorithms is based on swarm intelligence (SI). SI captures the idea that decision making among organisms in a community, such as ants and bees, uses local information and interactions with other agents and with their own environment, which in turn could be responsible for the rise of collective or social intelligence. One hypothesis is that complex interactions directly or indirectly contribute to the emergence of intelligence in highly developed biological species. The reasoning is that biological change results from the organism responding and adapting to alterations in its community and environment. Groups of same-species organisms have successfully carried out specific tasks through collective or swarm intelligence.[2–4] When resources are scarce, different species cooperate and can even coevolve.

SI has inspired researchers to develop an array of ingenious methods for solving challenging problems such as optimization based on insect and bird behavior. These algorithms—or *metaheuristics*—have been applied to solve both machine learning and engineering problems. The application of these SI algorithms makes sense because, in many ways, an algorithm's iterative process strongly resembles a self-organized system's evolution.[5,6] Moreover, the noise and randomness in algorithms can promote diverse solutions and avoid the pitfall of being trapped in local optimal solutions. The result is that the selection mechanism more rapidly converges on a global optimal solution. Just as in nature, iterative reorganization can lead to new structures of states, and such

```
Initialize the population and iteration counter
Evaluate the initial population and find the best solution
while (the stopping criterion is not met)
  Generate new solutions by modifying the existing population either locally or globally
  Evaluate the new solutions
  If the new solutions are better, then accept them and update the population
  If they are not better, accept them by some probabilistic criteria
        Update the iteration counter
end
Post-process results
```

**FIGURE 1.** Pseudocode of a typical nature-inspired algorithm.

states can be considered as possible solutions to the problem at hand.

To better understand both SI and metaheuristics, we surveyed the literature to identify underlying ideas. The algorithms described are representative of the more than 140 nature-inspired algorithms and their variants.[7] However, this number is likely already lower than what actually exists, as research is producing new algorithms almost weekly. Thus, the algorithms we describe can be only illustrative—showing how inspiration from nature can be turned into effective problem solving. Our hope is that this survey is a springboard to novel thinking about methods and tools to address computational challenges.

## FUNDAMENTAL CONCEPTS

Nature-inspired algorithms are based on SI, which in turn forms the foundation of metaheuristics.

### Swarm intelligence

SI is a complex process and no one is yet certain what mechanisms are required to ensure the emergence of collective intelligence. However, extensive recent studies suggest that individual agents such as ants and bees follow simple rules, act on local information, and have no centralized control.[8] Such rule-based interactions can lead to the emergence of self-organization, resulting in structures and characteristics at a higher system level. Individuals in the system are not intelligent, but the overall system exhibits collective intelligence. Such emerging self-organization can explain key swarming behavior, whether in ants or people.

Certain conditions are necessary for systems to self-organize: feedback, stigmergy (when individual system parts intercommunicate indirectly by modifying their local environment), multiple interactions, memory, and environmental setting.[2,8] However, what role each condition plays and how seemingly self-organized structures can arise from these conditions are unclear, and different studies typically focus on one or a few of these influences.

### Metaheuristics and monkeys

Most metaheuristic algorithms were developed from observing nature. Although quality solutions to a difficult optimization problem can be found in a heuristic way, there is no guarantee that the optimal solutions can be found. Such heuristic approaches might work most of the time, but there is no guarantee that they will. In contrast, metaheuristics tend to carry out higher-level search in a vast design space and usually employ some memory to record historical best solutions in combination with some form of solution selection. One of the earliest metaheuristics was a quasi-random approach that simulated metals annealing[9] by assigning some probability linked to a system's energy consumption. When the temperature is high, the probability approaches one; when the temperature is low (nearly zero), the probability approaches zero. Thus, when the system cools, the algorithm is less likely to accept worse solutions, only better ones.

Because of their ability to conduct wider searches and record best solutions, metaheuristics tend to outperform simple heuristics.[7,10] The Infinite and Finite Monkey Theorems provide insights into the way these algorithms work.

**Infinite Monkey Theorem.** This theorem assumes that if each monkey in a group is given a typewriter, the probability that the group will produce any given text—for example, William Shakespeare's complete works—will almost surely be one if the group contains an infinite number of monkeys randomly typing for an infinitely long time.[11,12] On a smaller scale, suppose the task was to reproduce the eight characters in "computer" from a random typing sequence of $n$ characters on a keyboard of 101 keys. The probability that a consecutive eight-character random string will be "computer" is $p = (1/101)^8 \approx 9.23 \times 10^{-17}$, which is extremely small, but not zero.

Indeed, the central idea in this theorem is that the probability of reproducing any text given infinite monkeys is not zero. In fact, probability theory can be used to formally prove that, for an infinitely long sequence, the probability of reproducing any text (including this article) is almost surely one.[13]

**Finite Monkey Theorem.** It is impractical to assume that infinite resources are available, so a more reasonable perspective is the Finite Monkey Theorem: the probability is greater than zero that a finite number of monkeys typing randomly for a fixed time can

**TABLE 1.** Advantages and disadvantages of different algorithms.

| Algorithm type and categories | Advantages | Disadvantages |
|---|---|---|
| Gradient-based<br>Deterministic | Higher convergence rate<br>Efficient for local search | Gradients are not easy to estimate<br>No guarantee for global optimality |
| Nature-inspired metaheuristics<br>Stochastic<br>Swarm intelligence | Global optimality<br>Simple and flexible<br>Can address hard problems<br>Can handle diverse range of problems<br>Easy to implement in parallel | Higher computational costs<br>Exact solutions are not repeatable<br>Less efficient for local search and simple problems |

reproduce some text. Transitioning that idea to people, a few billion people typing on keyboards daily has not yet produced another collected works of Shakespeare, but the probability that it will someday appear is greater than zero.

**Defining differences.** Metaheuristics and the monkey theorems have three crucial differences. First, monkeys type randomly without learning from or remembering their prior actions; metaheuristics try to learn from history to generate better solutions. Second, monkeys do not select what to type; metaheuristics tend to select the best or fittest solutions. That is, most use some form of randomization to enhance the search capability and thus increase the probability of finding truly optimal solutions. In theory, if such algorithms can be executed for a sufficiently long time with multiple runs, the best design solutions are likely to be found. However, an infinitely long search time is not possible, so most methods use thousands and even millions of design evaluations or iterations. Third, monkey typing is at most equivalent to a random search on a flat landscape of objectives, whereas metaheuristics use landscape information to guide the search process and generate new solutions—akin to hill climbing as the algorithm identifies better and better solutions.

## COMMON ALGORITHMIC CHARACTERISTICS

Although algorithms can have different sources of inspiration and different mathematical equations, they share certain characteristics. For example, they all use a solution population, and some solutions in that population will be replaced as its probability of success decreases. Solutions are generated by modifying the population either locally or globally. A new solution is accepted if its fitness is higher than that of existing solutions, and accepting a new solution will initiate a population update.

Thus, loosely speaking, all nature-inspired algorithms can be represented by the unified pseudocode in Figure 1.

As Figure 1 implies, all nature-inspired algorithms include both exploration and exploitation capabilities. To explore the design space more effectively, the population of candidate solutions must have sufficient diversity. On the other hand, for the algorithm to converge quickly using the fewest iterations, it must use landscape information such as gradients and modal shapes to guide its search in more promising regions. Such information exploitation can accelerate convergence. However, it can also reduce population diversity too quickly. The result is that all solutions might appear to be similar or the same and the algorithm becomes stuck.

## COMPARISON WITH OTHER ALGORITHMS

As Table 1 shows, relative to gradient-based algorithms, nature-inspired metaheuristics have distinct advantages, such as simplicity and flexibility and the ability to address hard problems. The drawback is that their computational costs tend to be higher because metaheuristics require multiple evaluations of design options and their stochastic nature makes it difficult to obtain repeat solutions.

## ALGORITHM TYPES

We selected seven examples of nature-inspired algorithms. There are many, many more, including gravitational, harmony, and wolf search algorithms and optimizations based on biogeography and the immune system.[14]

### Ant colony optimization

Ants are social insects, living in organized colonies, which can have as many as 25 million ants in each colony. Ants communicate through pheromones, which are chemicals that indicate a particular ant's presence. Each ant can follow pheromone trails and lay scent pheromones to communicate with other ants. The pheromone deposition acts as positive feedback mechanism; pheromone evaporation acts as an escape mechanism from any randomly chosen trail.

The ant colony optimization algorithm, developed in 1992, exploits these local interactions and pheromone variations, essentially mimicking the ants' primary behavior and social characteristics.[8] Surprisingly, this system with its simple local rules is quite effective at solving optimization problems—from vehicle routing to the well-known traveling-salesman problem (TSP). The algorithm codes paths the ants visit as actual paths. The choice of which route for a particular ant at the junction of actual paths depends on the pheromone concentration on each possible path. Pheromone deposition signifies a favored route, and pheromone evaporation ensures that nonoptimal routes discovered in the earlier search stages will not converge too quickly.

### Bee colony optimization

A honeybee colony typically consists of about 20,000 to 80,000 bees with one queen and a few hundred male bees or drones, with the rest being female worker bees. The workers can be scouts, onlookers, guards, or nectar collectors. A scout communicates to other workers through a waggle dance that it has found a new nectar source. An algorithm that mimics the main characteristics of the bees' foraging behavior—the artificial bee colony algorithm—was developed in 2005.[15] The algorithm divides bees into employed, scout, and onlooker bees; codes the nectar source's location as a solution vector; and links the nectar amount with the landscape of objectives. Although this description is somewhat simplistic, it captures the main characteristics of foraging behavior. It has been used to solve unconstrained numerical optimization problems and constrained optimization problems as well as to train neural networks.

### Bat algorithm

Most of the 800+ microbat species use echolocation for navigation, which involves emitting about 10 to 20 ultrasonic bursts per second. Each burst lasts only a few thousandths of a second, has a frequency of 20 to 200 kHz (humans can hear at most a 20-kHz burst), and can be as loud as 120 dB (the noise level of a jumbo jet taking off). When a bat finds an insect and is homing in on its prey, the pulse emission rate can accelerate to 200 pulses per second with a higher frequency. Echolocation allows the bats to more accurately gauge a flying insect's size, position, range, speed, and direction.

SIMPLICITY, FLEXIBILITY, AND THE ABILITY TO ADDRESS HARD PROBLEMS ARE THE RELATIVE ADVANTAGES OF NATURE-INSPIRED METAHEURISTICS.

The bat algorithm, developed in 2010, uses characteristics of pulse emission and frequency tuning[16] and considers the bat's location as a solution vector in the search space. The frequency tuning lets the bat explore the search space on a larger scale, while the speedup of the pulse emission focuses on the neighborhood of local promising solutions. Among the whole bat group, there is a global best solution, and other bats tend to swarm toward it. Consequently, convergence is relatively rapid, controlled by frequency tuning, pulse emission, and loudness. The bat algorithm has been applied in many real-world applications such as engineering optimization, training neural networks, image processing, and solving the TSP.

### Cuckoo search

Some cuckoo species, such as Old World cuckoos, engage in brooding parasitism, in which the cuckoo lays eggs in the nest of a host bird such as a warbler. The eggs then hatch and the host bird raises the cuckoo chicks. Because cuckoos are adept at mimicry, the texture, color, and size of the cuckoo's eggs look very similar to those of the host birds' eggs. Even so, some host birds can recognize cuckoo eggs and then get rid of them or abandon the nest, creating a kind of evolutionary arms race between the two species.

The cuckoo search algorithm, developed in 2009, considers a cuckoo's egg as a solution vector and the nesting field as the search space.[17] There is some evidence that both the cuckoo's and host bird's flight paths can obey Lévy flights—flights with occasional long jumps followed by many local random steps—which makes the search more effective over a large region. The similarity of eggs can be converted into the similarity of solutions, which helps the iterative search process reach convergence, and the discovery probability aids in global exploration. The cuckoo search algorithm has been successfully applied to engineering optimization and image-processing problems.

### Particle swarm optimization

Birds and fish form swarms as they move, in part because there is safety in numbers. Each bird follows simple rules and often only tracks the flying status of the seven birds adjacent to it. Newtonian mechanics govern flight motion, and, amazingly, birds almost never collide. The overall swarm shows some organized structures.

Particle swarm optimization, developed in 1995, is based on these simple rules and swarming characteristics.[18] It considers a particle's position as a solution vector, so each particle has a historical best solution, and the entire swarm produces the current best solution. The simple rules are used to update each particle's position and velocity, and all particles tend to swarm toward the centroid—the global best solution. The system can converge very quickly in many cases. On one hand, rapid convergence makes this method an effective optimizer; on the other hand, convergence might be too early, thus leading to premature convergence. Because many approaches are possible in resolving premature convergence, this algorithm has quite a few variants. Particle swarm optimization has been applied to almost every area in science and engineering, including design optimization, image process, and scheduling.

### Firefly algorithm

Fireflies in tropical regions use bioluminescence to communicate, and each firefly species can have a distinct flashing pattern that serves as a unique signaling system. Other fireflies of the same species will be attracted by the flashing light and thus might swarm to it. Because light intensity decreases with distance and increased air pollution, the flashing light is visible only a few hundred meters away.

The firefly algorithm, developed in 2008, is based on these flashing characteristics. A firefly's position corresponds to a solution vector, and the objectives landscape determines its brightness or attractiveness. Because short-distance attraction is stronger than long-distance attraction, the whole firefly algorithm can be automatically subdivided into subgroups or subswarms, and each subswarm will swarm around a local model in the landscape with peaks for maximization and valleys for minimization. Consequently, the algorithm can find multiple optimal solutions simultaneously and, under certain conditions—such as the right attraction range and the monotonic decrease of its randomness—it can converge more quickly than genetic algorithms and particle swarm optimization. Like particle swarm optimization, the firefly algorithm has also been widely applied—in areas from scheduling and classification to image processing and design optimization in engineering.

### Flower pollination algorithm

Obviously, not all algorithms are based on swarming behavior. Plants also provide ingenious problem-solving models, such as the pollination process. Of the quarter million flowering plant species, 90 percent are biotic, requiring pollen transfer through some agent, such as bats, birds, ants, and bees. Some pollinators, such as hummingbirds, tend to visit certain flower species exclusively, forming a flower constancy. In abiotic pollination—which is characteristic of the remaining 10 percent of flowering plant species—pollination tends to be local, carried out by wind or water. In contrast, biotic pollination can be global because pollinators tend to move across longer distances.

The flower pollination algorithm, developed in 2012, encodes a pollen gamete as a solution vector to a problem, and uses biotic pollination to simulate global search and abiotic pollination for local search.[19] The switch between global and local search is controlled by a probability to simulate the percentage of biotic and abiotic pollination. The algorithm has been applied to solve multiobjective optimization problems.

## APPLICATION AREAS

Nature-inspired algorithms have been applied to a wide range of diverse applications, with literally hundreds (according to Web of Science) or even thousands (according to Google) of new articles published each year describing work to explore and solve diverse problems in real-world applications across domains.

### Hard problems

A classic example of a hard problem is the TSP, which has been solved using metaheuristic approaches such as ant colony optimization.[8] The TSP is representative of how nature-inspired methods can be a powerful alternative for dealing with hard problems because existing approaches cannot solve hard problems on a realistic timescale. Although nature-inspired algorithms are not guaranteed to produce optimal solutions consistently, suboptimal solutions can still be very useful and are certainly better than no solutions at all.

### Telecommunications

Nature-inspired algorithms have also been used to optimize local access networks to maximize quality of service and minimize overall energy consumption.[20] These algorithms enable better designs than those obtained by traditional methods; new design options tend to have improved efficiency with lower energy consumption and less cochannel interference.

### Image processing

Image processing often concerns time-consuming computational tasks. When traditional techniques are combined with nature-inspired algorithms, features can be extracted more accurately for many applications including image segmentation, classification, and deep-belief networks.[21]

### Engineering design

Many engineering design problems are highly nonlinear—such as structural design and wireless sensor networking—and traditional methods do not handle such nonlinearity well. Recent studies show that nature-inspired algorithms can often produce better design options more efficiently because they use landscape information to search design-space regions and share information among different agents (or solution sets).[14,17,22]

### Vehicle routing

Vehicle routing is a challenging problem with many applications in logistics and combinatorial optimization. Several vehicle routing problems—for example, deploying vehicles under various constraints—have been solved using nature-inspired algorithms to enhance performance. In one study, the use of metaheuristic approaches was shown to be more effective than traditional algorithms in solving vehicle routing problems and transport costs were lower. [23] The scheduling of aircraft, departure slot allocation, and airspace management can also be solved satisfactorily by nature-inspired metaheuristics, with solutions leading to reduced running costs and more effective use of departure slots.[24]

As applications become more complex and are tasked with handling big-data volume, velocity, and variety, researchers will increasingly look at how nature handles similar problems, motivating studies in important new directions. One area is theoretical analysis—understanding how nature-inspired algorithms work. Any theoretical insight gained will be extremely useful in guiding the proper use of current algorithms and in designing new algorithms that can solve challenging problems more effectively. Optimization problems remain in all areas of science, engineering, industry, and business applications, and there is little doubt that application explosion will continue.

Algorithms that move beyond mimicking nature will also be important in gaining new perspectives on problem-solving strategies. Biological systems evolve in a Darwinian manner with no high-level objective; they merely respond and adapt to environmental changes. Thus, survival of the fittest does not require global optimality as long as the fittest can survive as a population or a species. Given that nature is not always optimal, research must explore ways to give algorithms more robustness in adapting to change.

Hybrid algorithms will be of interest because they combine the best of several algorithms. How to design a better hybrid is itself a higher-level optimization problem—the optimization of optimization algorithms. Research must move beyond the trend of developing hybrids solely to claim gains in efficiency and robustness over nonhybrid algorithms. The comparative study of any hybrid algorithm should involve only hybridized counterparts.

In the Internet of Things age, algorithms must become self-adaptive and intelligent. There are far too many applications to find the best algorithm for a particular problem. In contrast, self-adaptive approaches enable the automatic choice of algorithms that can adapt for a given task set, carrying out tasks with little or no user intervention. Self-adaptation also means being able to control performance and self-tune parameters to maintain the highest efficiency. The ultimate goal is to develop an intelligent toolbox that can solve the problem at hand with no more user effort than pressing a button.

Obviously, these are only a few possible directions. A combination of new algorithms coupled with traditional techniques can be very useful to pursue. After all, traditional techniques have been well established and widely tested, and they are among the most useful to a specific class of problems. New methods will be most needed when traditional methods do not work well. However, given application demands, we expect to see more research in nature-inspired algorithms to evolve more efficient tools for solving the diversity of problems in real-world applications.

## REFERENCES

1. X.S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley and Sons, 2010.
2. L. Fisher, *The Perfect Swarm: The Science of Complexity in Everday Life*, Basic Books, 2009.
3. P. Miller, "Swarm Theory," *Nat'l Geographic*, July 2007; ngm.nationalgeographic.com/2007/07/swarms/miller-text.
4. J. Surowiecki, *The Wisdom of Crowds*, Anchor Books, 2004.
5. W.R. Ashby, "Principles of the Self-Organizing System," *Trans. Symp. Univ. of Illinois: Principles of Self-Organization*, H. Von Foerster and G.W. Zopf, Jr., eds., Pergamon Press, 1962, pp. 255–278.

## ABOUT THE AUTHORS

**XIN-SHE YANG** is a reader in modeling and optimization at Middlesex University and an elected Bye-Fellow in Downing College at Cambridge University. His research interests include nature-inspired computation, swarm intelligence, modeling, and optimization. Yang received a DPhil in applied mathematics from the University of Oxford. He is chair of the Task Force on Business Intelligence and Knowledge Management of the IEEE Computational Intelligence Society. Contact him at x.yang@mdx.ac.uk.

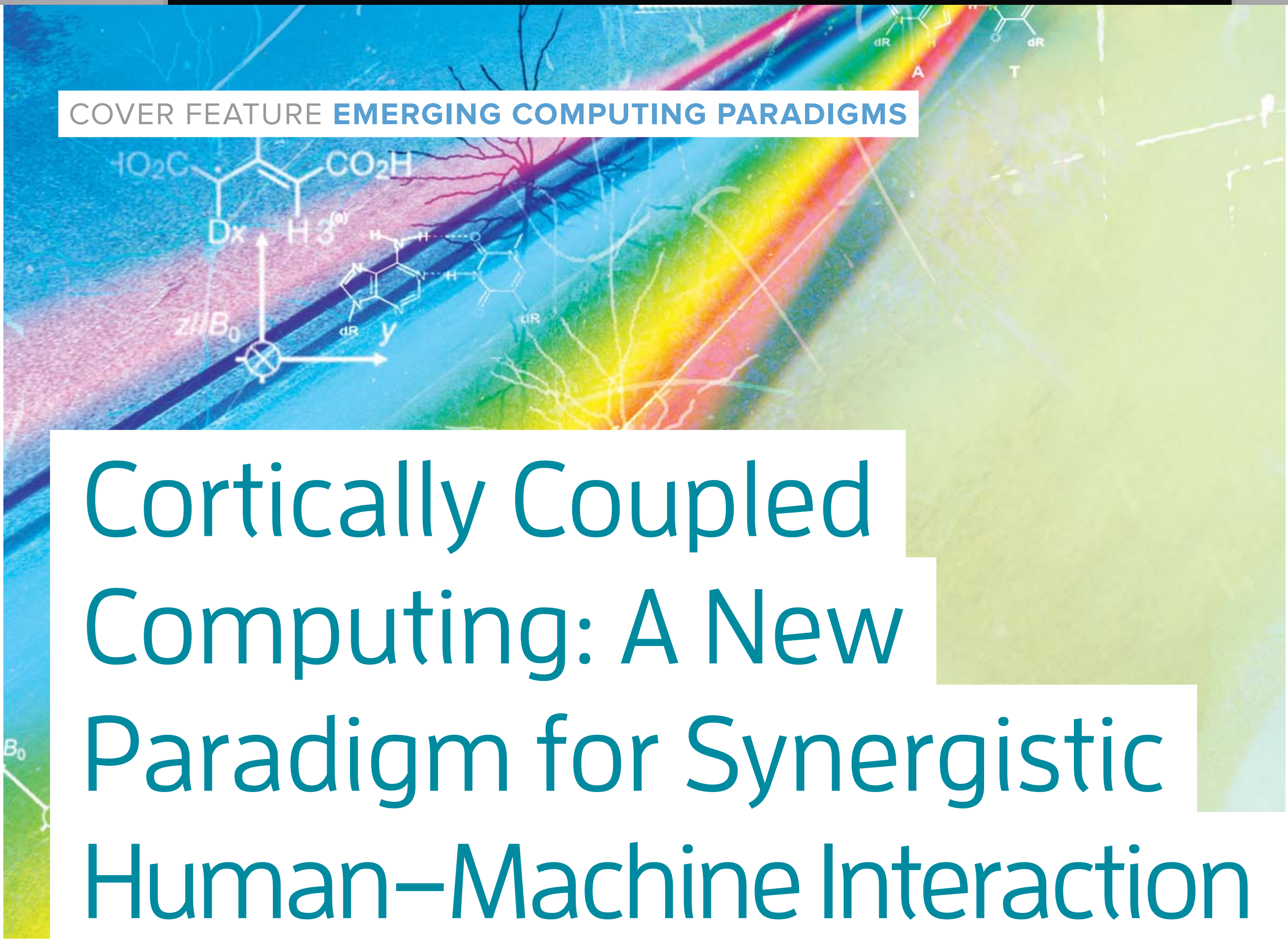
**SUASH DEB** is an IT and educational consultant. His research interests include metaheuristics and swarm intelligence. Deb received an MTech in computer science from the University of Calcutta and a United Nations Fellowship in computer science from Stanford University. He is a Senior Member of IEEE and founding president of the International Neural Network Society's India Regional Chapter. Contact him at suashdeb@gmail.com.

**SIMON FONG** is an associate professor in the Department of Computer and Information Science at the University of Macau. His research interests include data mining, metaheuristics, and big-data analytics. Fong received a PhD in computer science from La Trobe University. He is a member of IEEE and ACM. Contact him at ccfong@umac.mo.

**XINGSHI HE** is a professor and deputy dean in the College of Science at Xi'an Polytechnic University. His research interests include mathematical modeling, algorithms, statistics, and computational intelligence. He received an MS in mathematics from Shaanxi Normal University. He is a recipient of the 2016 Shaanxi Provincial Distinguished Teaching Achievement Award. Contact him at xsh1002@126.com.

**YU-XIN ZHAO** is a professor of control and navigation at Harbin Engineering University (HEU). His research interests include nature-inspired computation, intelligent transport systems, and marine navigation systems. Zhao received a DEng in control science and engineering from HEU. He is a member of the British Royal Institute of Navigation. Contact him at zhaoyuxin@hrbeu.edu.cn.

6. E.F. Keller, "Organisms, Machines, and Thunderstorms: A History of Self-Organization, Part Two: Complexity, Emergence, and Stable Attractors," *Historical Studies in the Natural Sciences*, vol. 39, no. 1, 2009, pp. 1–31.
7. X.S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, 2014.
8. E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford Univ. Press, 1999.
9. S. Kirpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, 1983, pp. 671–680.
10. C. Blum and A. Roli, Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, vol. 35, no. 2, 2003, pp. 268–308.
11. A. Gut, *Probability: A Graduate Course*, Springer, 2005.
12. G. Marsaglia and A. Zaman, "Monkey Tests for Random Number Generators," *Computers & Mathematics with Applications*, vol. 26, no. 9, 1993, pp. 1–10.
13. G. Lorge, "The Best Thought Experiments: Schrödinger's Cat, Borel's Monkeys," *Wired*, 22 May 2007; www.wired.com/2007/05/st-best-4.
14. X.S. Yang et al., *Swarm Intelligence and Bio-inspired Computation: Theory and Applications*, Elsevier, 2013.
15. D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, tech report TR06, Computer Eng. Dept., Erciyes Univ., 2005.
16. X.S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," *Proc. Conf. Nature-Inspired Cooperative Strategies for Optimization* (NICSO 10), J.R. Gonzales, ed., *Studies in Computational Intelligence*, vol. 284, 2010, pp. 65–74.
17. X.S. Yang and S. Deb, "Multiobjective Cuckoo Search for Design Optimization," *Computers and Operations Research*, vol. 40, no. 6, 2013, pp. 1616–1624.
18. J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int'l Conf. Neural Networks* (ICNN 95), 1995, pp. 1942–1948.
19. X.S. Yang, M. Karamanoglu, and X.S. He, "Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization," *Eng. Optimization*, vol. 46, no. 9, 2014, pp. 1222–1237.
20. X.S. Yang, S.F. Chien, and T O. Ting, *Bio-inspired Computation in Telecommunications*, Elsevier, 2015.
21. X.S. Yang and J.P. Papa, *Bio-inspired Computation and Applications in Image Processing*, Elsevier, 2016.
22. H.E.P. Espinosa, *Nature-Inspired Computing for Control Systems*, Springer, 2016.
23. N. Labadie, C. Prins, and C. Prodhon, *Metaheuristics for Vehicle Routing Problems*, John Wiley and Sons, 2016.
24. N. Durand et al., *Metaheuristics for Air Traffic Management*, John Wiley and Sons, 2016.



COVER FEATURE **EMERGING COMPUTING PARADIGMS**

# Cortically Coupled Computing: A New Paradigm for Synergistic Human–Machine Interaction

**Sameer Saproo, Josef Faller, Victor Shih, and Paul Sajda,** Columbia University

**Nicholas R. Waytowich, Addison Bohannon, Vernon J. Lawhern, and Brent J. Lance,** US Army Research Laboratory

**David Jangraw,** National Institute of Mental Health

*Unlike traditional brain–computer interfaces that use brain signals for direct control of computers and robotics, a cortically coupled computer system opportunistically senses the brain state, capturing a user's implicit or explicit computation, and then communicates this information to a traditional computer system via a neural interface.*

Most modern computers embody a variant of the von Neumann architecture, central to which is the concept of the stored program or instruction set. The task of devising the appropriate instruction set—programming—has been the exclusive domain of trained human operators. In essence, programming is the art of transferring knowledge about the world and an executable set of actions that achieve some desired outcome from a human to a computer using electromechanical interfaces such as the keyboard and mouse. In developing machines' computational abilities, human users often become a bottleneck because information transfer from human to machine is limited by our ingenuity in crafting logical instructions that capture knowledge in the human brain, as well as the interface's bit rate. However, as machine intelligence approaches the general effectiveness of human intelligence, the dynamic of how we interface or effectively collaborate with machines will likely disrupt the current need for explicit human programming of machines.

Machine learning, one of the fastest-growing areas of computer science, heavily relies on the availability of

0018-9162/16/$33.00 © 2016 IEEE

labels for data, such as object labels in real-world images, so that the machine can learn the associations between various entities in the data. Many of these labels are subjective—they are generated by humans using their knowledge of the world—and therefore hard to produce through purely automated methods, which makes them valuable. In fact, user-clickstream analyses have proved indispensable to commercial entities such as Google, Amazon, and Facebook, which utilize the data to personalize the services they offer. Although the volume of explicit labels generated by human activities—such as clicking on webpages—is impressive, it is tiny compared to the volume of labels generated by the human brain during implicit processing of sensory data. Human perception is an active process even if it does not result in obvious behavior; if it were possible to access the contents of people's cognitive processes while they stroll down a street or watch a TV show in silence, the wealth of information usable by machine-learning systems would be likely transformative.

Brain–computer interfaces (BCIs), also known as brain–machine interfaces (BMIs), decode brain content—noninvasively using electroencephalography (EEG) from scalp electrodes or invasively using electrocorticography (ECoG) from cortical electrodes—and make a computer system act upon that content to restore communication and/or control. The most common example of a BCI application is enabling a physically disabled person to move a cursor on a computer screen by merely thinking about it. Although BCI systems for active control are important for specific populations, they have so far shown very limited suitability for the general population, primarily because of the low signal-to-noise ratio (SNR) of measured brain signals. However, even low SNR signals can provide useful information when assessed over long timescales. Such information can be leveraged to augment the capabilities of other computational systems, especially their capacity to learn.

We propose a new paradigm, *cortically coupled computing*, wherein both humans and machines do computational tasks and communication between the two is enabled via a BCI. In its simplest version, cortically coupled computing attempts to use brain-derived information about the world to teach specialized knowledge to a machine that has not been programmed a priori. Fully realized, it envisions a future in which highly advanced AI and human users execute tasks cooperatively and seamlessly, and such human–machine synergy is facilitated by next-generation BCIs between humans and AI. We illustrate the concept of cortically coupled computing using examples of several real systems that implement aspects of this novel paradigm.

## CORTICALLY COUPLED COMPUTER VISION

Although substantial progress has been made in computer vision (CV), particularly the recent advances in deep learning, machines' comprehensive "understanding" of images remains a challenge in noisy and context-rich environments. Consider an image analyst assessing a potential threat in aerial reconnaissance data; the large quantity of images precludes detailed assessment of each image upfront, so the analyst must triage images according to their subjective importance, or "gist." Image comprehension in this case requires not only understanding an object's identity as inferred from the image, but also assessing its significance given contextual information. To accomplish this, the CV system must leverage the expertise of a trained image analyst.

Cortically coupled computer vision (C3V) combines the human visual system's general-purpose and context-aware capabilities with CV's fatigue-free "number-crunching" capabilities.[1,2] The overarching idea is to increase the throughput of an automated image

## IEEE BRAIN INITIATIVE

In late 2015, IEEE created the IEEE Brain Initiative (http://brain.ieee.org) with the mission to facilitate cross-disciplinary collaboration and coordination to advance research, standardization, and development of technologies in neuroscience to help improve the human condition. IEEE is in a unique position to leverage its broad expertise in electronics, communication, sensors, power management, and other technologies to bring an engineering and systems perspective to worldwide activities that are focused on understanding and interfacing with the brain. For example, the IEEE Brain Initiative is coordinating standardization and technology development activities with the US-led Brain Research through Advancing Innovative Neurotechnologies (BRAIN) Initiative (www.braininitiative.nih.gov) and EU-led Human Brain Project (www.humanbrainproject.eu), as well as initiatives that are underway or being launched in Japan, Australia, and China.

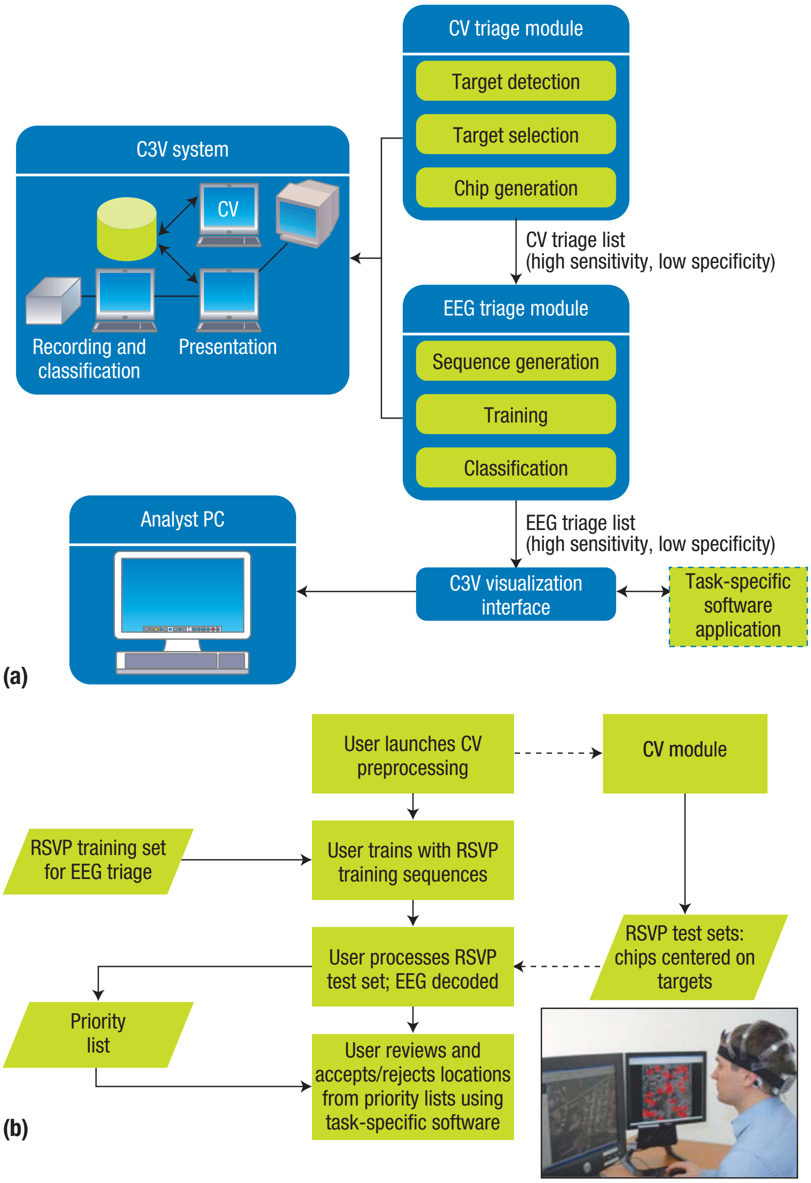

CV triage module
Target detection
Target selection
Chip generation
C3V system
CV
Recording and classification
Presentation
CV triage list (high sensitivity, low specificity)
EEG triage module
Sequence generation
Training
Classification
Analyst PC
EEG triage list (high sensitivity, low specificity)
C3V visualization interface
Task-specific software application
(a)
User launches CV preprocessing
CV module
RSVP training set for EEG triage
User trains with RSVP training sequences
User processes RSVP test set; EEG decoded
RSVP test sets: chips centered on targets
Priority list
User reviews and accepts/rejects locations from priority lists using task-specific software
(b)

**FIGURE 1.** Example cortically coupled computer vision (C3V) system: (a) system components and (b) system workflow. CV: computer vision; EEG: electroencephalography; RSVP: rapid serial visual presentation. Figure adapted from P. Sajda et al., "In a Blink of an Eye and a Switch of a Transistor: Cortically Coupled Computer Vision," *Proc. IEEE*, vol. 98, no. 3, 2010, pp. 462–478.

search and triage system using additional information from a BCI that assesses an analyst's "level of interest" in transient images by analyzing the analyst's EEG signals. In essence, C3V labels each image in a dataset with a human-generated cognitive tag and then performs machine learning.

To generate these tags, C3V displays task-relevant images to the analyst in a rapid serial visual presentation (RSVP), wherein hundreds of images are presented in sequence with each image displayed for only 100–200 milliseconds. Because each image is visible very briefly, the analyst can only get the gist of its content. In this brief time, a human analyst can only gain a conceptual understanding of the image—a task to which we are acutely adapted.[3] If this brief conceptual understanding aligns with the analyst's intent or specific interest, then he or she will rapidly orient attention to the image, eliciting a neurophysiological signature that C3V can decode.[1,4,5] Because the orienting event is not all-or-none, but rather a smooth function of the degree of interest, the decoded signature can serve as a level-of-interest score to prioritize the images, much like Google uses PageRank scores to prioritize search results. Interestingly, this attention-orienting signal can be detected through EEG not only for objects in still imagery but also spatiotemporal events in video clips.[6]

Although the attention-orienting paradigm leverages the human visual system's ability to rapidly tag images that contain objects of interest or frames in a video that contain events of interest, the system's throughput is limited to the human rate of operation: 5–10 images per second. C3V leverages CV to either filter potential images of interest to pass on to the analyst or postprocess the level-of-interest scores for a limited set of images to efficiently search a much larger image database. We now describe these two cases in more detail.

## CV followed by human visual processing

Figure 1 shows a prototypical C3V system used to analyze satellite imagery.[2] The system has three components: a CV triage module, an EEG triage module, and a display interface that lets a user visualize the results of the combined triage.

The user first initiates the CV module, which analyzes the entire image—in this case, a large satellite image. The CV processing detects possible sites of interest in the image based on very general models or user-selected constraints (for example, search for manmade objects). The system uses these detections to construct image "chips," or subsections, for presentation to the user via RSVP. The user wears an EEG cap (Figure 1b inset) while the EEG triage module runs a machine-learning–based classifier pretrained to identify attention-orienting events from the user's EEG signals. The system passes on the set of image chips generated by the CV module to the user, decodes the user's EEG signals in real time to generate level-of-interest scores for each image chip, and tags the image chips with these scores to generate a priority list for the regions of interest. The interface then incorporates these priority lists into the satellite imagery, allowing the analyst to pan and zoom, mark objects, and jump to different map locations corresponding to high level-of-interest scores.

The system has been tested in realistic image search environments and found to improve image throughput during search by up to 300 percent.[2]

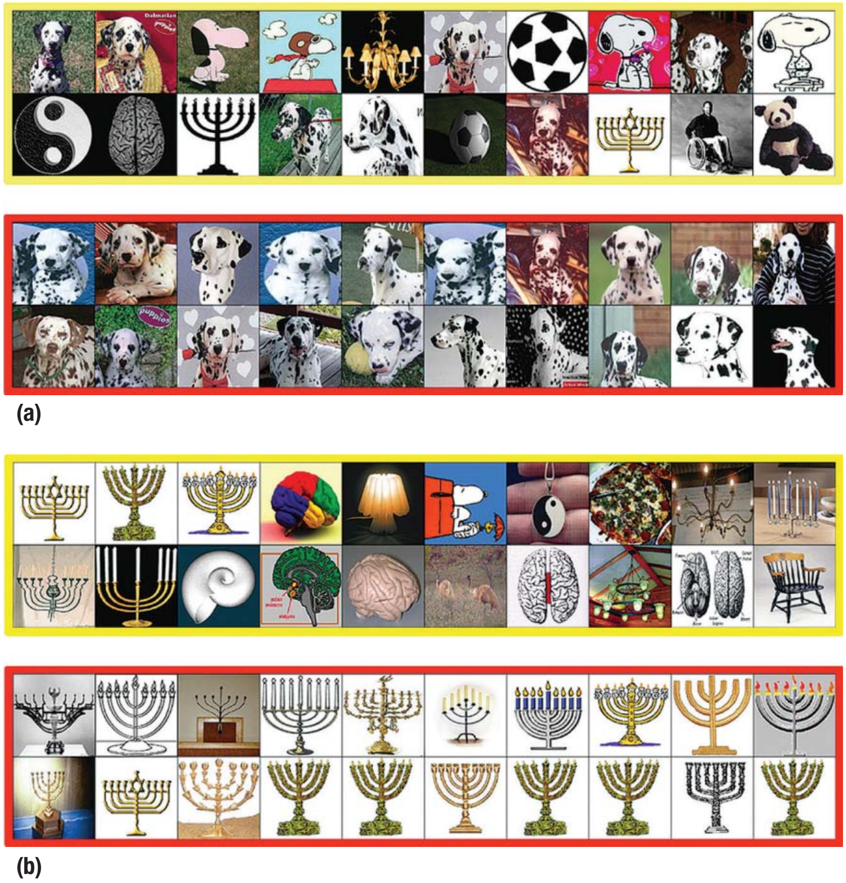


**FIGURE 2.** Top image search results with the subject interested in (a) dalmatians and (b) menorahs from decoded EEG (yellow-bounded boxes) and after subsequent post-processing by semisupervised graph-based CV (red-bounded boxes). Images are rank ordered from left to right and top to bottom in terms of level of interest as inferred from EEG decoding (yellow), with updated rank ordering given EEG score and visual similarity structure in graph-based CV model (red). These results show that integrating decoded EEG and graph-based CV leads to higher precision and recall than either system would achieve alone. Figure adapted from J. Wang et al., "Brain State Decoding for Rapid Image Retrieval," *Proc. 17th ACM Int'l Conf. Multimedia* (MM 09), 2009, pp. 945–954.

### Human visual processing followed by CV

In C3V, the CV module can act not just as a preprocessor for the human-in-the-loop but also as a postprocessor, wherein it uses the cognitive labels of image chips for identifying visually or semantically similar images in a large database. In one example, C3V uses semisupervised graph-based inference models to propagate the level-of-interest scores to unlabeled images in order to assess the attention-grabbing property of unseen images.[7] As Figure 2 shows, CV postprocessing can improve both the precision and recall of a category of searched images in a large database. The performance is further improved when human level-of-attention scores are used as labels for the CV module, which then queues up new images according to perceived interest. When C3V is run in this closed-loop mode, the precision rates significantly increase and then gradually flatten out after iterating the EEG decoding and CV search about 2.5 times.[8]

## EXTENDING C3V TO A MULTIAGENT ENVIRONMENT

Several platforms expand on the C3V paradigm, with different CV models, more ergonomic and useable BCIs, and various communication topologies. For example, the Human-Autonomous Image Labeler (HAIL) developed at the US Army Research Laboratory is a scalable, closed-loop image-labeling system that supports collaboration across a network of heterogeneous human and CV image-labeling agents. HAIL utilizes a machine-learned assignment strategy to achieve superior performance from the joint efforts of all agents than from any subset.[9] HAIL can assign image data to agents either serially, where images are allocated to high-throughput agents first, or in parallel, where all agents analyze subsets of the overall image database simultaneously.

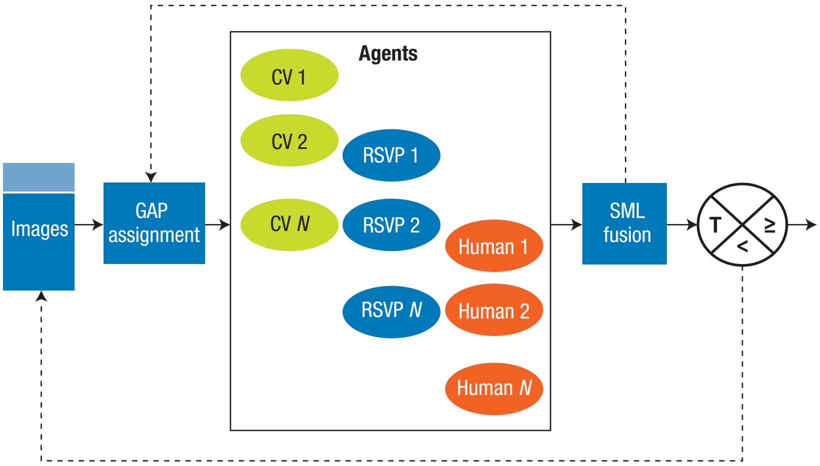


**FIGURE 3.** Parallel image assignment architecture for the Human-Autonomous Image Labeler (HAIL) system. HAIL optimally distributes images from a database to CV as well as system-paced and self-paced human agents in parallel according to the generalized assignment problem (GAP) method. The agents classify the assigned images, and HAIL assigns an overall confidence level to each image at a fusion node using the Spectral Meta-Learner (SML). Based on the calculated image confidence level and inferred agent accuracy, the system deems the images classified or routes them back to the assignment node for further classification.

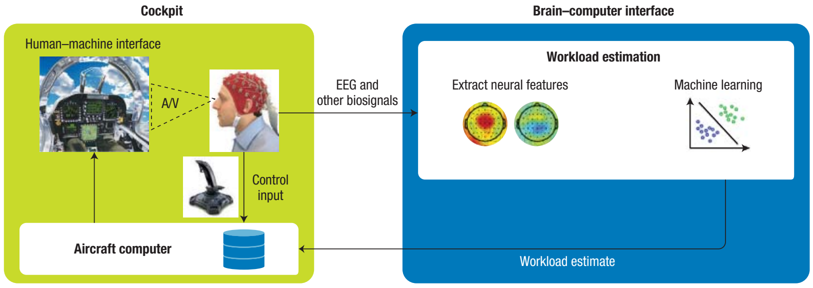


**FIGURE 4.** Brain–computer interface (BCI) that continuously monitors pilot workload and informs the aircraft computer, which can mitigate workload by providing direct feedback to the pilot or manipulating the cockpit interface. The system can tag each human action in the cockpit with the estimated cognitive workload, which can later be used to improve human–machine interaction by assessing which features increase workload.

### Serial image assignment

With serial image assignment, HAIL first allocates images to a deep-learning–based CV module for identification based on a search template. It then forwards the images prioritized by the CV algorithm to a "system-paced" human agent for observation via RSVP. While the human agent is watching the steady flow of images, HAIL decodes his or her EEG signals in real time and can use these to reassess the CV module's decisions. Finally, HAIL displays the remaining images to a self-paced human agent, who provides manual decisions through an intuitive tiled-image display with touchscreen. Although the current system uses static statistical models for both the CV module and EEG decoder, ongoing research is focused on incorporating adaptive classifiers and feedback modalities into the system, which will enable the human and CV agents to adapt and learn from each other.

### Parallel image assignment

HAIL's parallel image assignment policy was inspired by research in optimal crowdsourcing.[10] As Figure 3 shows, the system generates an optimal parallel assignment of images to CV as well as system-paced and self-paced human agents according to the generalized assignment problem (GAP) method. HAIL combines the agents' image classifications with the Spectral Meta-Learner (SML),[11] an unsupervised ensemble fusion classifier, and uses those results along with inferences about the agents' reliability to inform subsequent image assignments. The system iterates through image assignment and joint image classification until reaching a desired level of confidence for all images. This collaborative image processing by heterogeneous agents achieves human-level performance in significantly less time than can be achieved by a single human agent or an ensemble of human agents. Furthermore, HAIL's multiagent adaptive policy optimally combines the efforts of agents whose individual efficacy is unknown a priori and/or changes with time. In this way, the system's computational characteristics evolve with the network topology and node characteristics in real time.

## INFERRING DYNAMIC COGNITIVE WORKLOAD IN COMPLEX HUMAN–MACHINE INTERACTION

The idea that a synergistic human–machine system can smartly adapt to dynamic cognitive workload can be extended to non–vision-oriented human–machine interactions. For instance, Figure 4 shows a closed-loop BCI that continuously monitors pilot workload during a stressful navigation task: boundary avoidance with

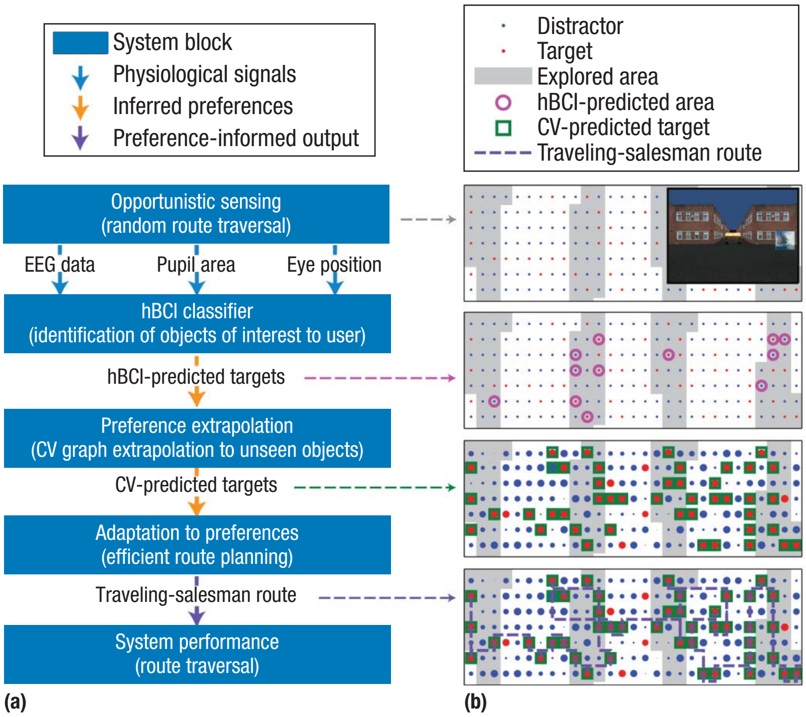


**FIGURE 5.** Example of a BCI that pairs opportunistic sensing with an automatic navigation system. (a) Block diagram. (a) Bird's-eye view (and user's view, inset) of the environment being navigated. Dot size indicates CV score; objects with higher scores are considered more likely to be targets. The hybrid BCI (hBCI) uses physiological signals to label objects that appeared to capture the user's interest. Its CV graph then extrapolates these labels to other, unseen objects in the environment. Finally, the route planner finds an efficient route to visit objects it expects to catch the user's interest. In this way, the navigation system can create a route that is both constrained by the environment and adapted to the user's specific preferences. Figure adapted from D.C. Jangraw et al., "Neurally and Ocularly Informed Graph-Based Models for Searching 3D Environments," *J. Neural Eng.*, vol. 11, no. 4, 2014; doi:10.1088/1741-2560/11/4/046003.

dynamically changing boundary sizes. In such a task, pilot workload depends on various actions—for example, joystick movement, throttling, and interaction with the cockpit interface such as responses to audible and display events—as well as continuous monitoring of information on gauges and digital displays.

Initial experimental results have found distinct neural correlates of high pilot workload that can lead to nonoptimal coupling between pilot and aircraft, resulting in a dangerous phenomenon known as pilot-induced oscillation. However, decoding task-related cognitive workload in real time and providing intervention in the form of feedback to the human or machine could favorably impact human–machine interaction. Additionally, the system can label human actions in the cockpit with cognitive estimates of workload. In post hoc analysis, machine-learning techniques could determine what sequence or combination of actions increase pilot workload. Such information can be used to help build more efficient cockpit interfaces. This conceptual paradigm can be extended to other navigation-related tasks, such as driving a vehicle, in which the interface with car controls and information displays can be optimized to reduce driver distraction and thereby reduce accidents.

## ADAPTING AI TO HUMAN PREFERENCES USING OPPORTUNISTIC SENSING

As their capabilities expand, AI systems are increasingly being tasked to assist and interact with human users—for everything from choosing entertainment to driving cars. Many of these human-facing AI systems would benefit greatly from an ability to sense a user's preferences, which could be used to personalize AI output and build trust between human and machine. But because modern AI systems such as deep learning often rely on massive quantities of training data, communicating these preferences could require prohibitively large effort from the user. Fortunately, recent advances in opportunistic sensing make it possible to evaluate a user's preferences without requiring a dedicated training session or even eliciting them directly—rather, a system can use physiological correlates of subjective interest or expectations that are produced in the normal course of events.[12] These nearly effortlessly produced cognitive labels can be used to tailor an AI system's behavior to match certain preferences of an individual user, thereby increasing human–computer synergy.

Figure 5 outlines a navigation system that uses a hybrid BCI (hBCI) to infer the type of objects that a user prefers and plot an efficient route past them.[13] Experimental subjects were driven through small portions of a large 3D virtual environment full of objects. They were asked to count one "target" category of objects and ignore the other three "distractor" categories. The target category represented objects that the user would prefer to visit, and the overall system's job was to navigate an efficient route past as many of these preferred objects as possible.

In the initial exploration phase, targets elicited an attention-orienting response from the subjects that affected their EEG signals, pupil dilation, and dwell time (time spent gazing at an object). The system collected these physiological signals and, using an bHBI classifier, found a small set of "hBCI-predicted targets" that appeared to capture the subject's attention. It then used CV[7] to extrapolate these target labels to other, unseen objects, and a traveling-salesman problem solver to find an efficient route that visited all of these "CV-predicted targets." Using this route, the median subject was able to see 84 percent of the targets in the environment while traveling only 40 percent of the distance required to see every object. Thus, with very little user effort, the navigation system found a route that was both constrained by the environment and adapted to the user's specific preferences.

**[ CORTICALLY COUPLED COMPUTING IS SUITED TO A WIDE VARIETY OF REAL-WORLD APPLICATIONS. ]**

New AI applications can apply opportunistic sensing to other domains by inferring different types of preferences from physiological signals or tasking the AI with different objectives. For example, developers of a deep learning system tasked with driving an autonomous vehicle might wish to adapt braking speed or turning sharpness to match passenger preferences. The system could use physiological correlates of discomfort or surprise to infer when preferences were not met. This labeling of actions could be extrapolated to a simulated training environment, where the deep learning system could adapt its driving to a multitude of inferred labels without having to wait for additional user input. The newly trained system would then be better equipped to brake and turn the way the user prefers. In this way, the system can sidestep the high cost of evaluating preferences explicitly, while still providing the large number of training labels required for modern AI systems.

Such applications would benefit from hardware and software that are able to reliably sense physiological signals in real-world environments. Thankfully, such hardware and software are advancing rapidly, with consumer EEG headsets and video-based eye-tracking systems increasingly geared toward mobile applications. Additionally, the rising popularity of virtual reality systems might provide an opportunity to evaluate preferences in simulated environments or games in which users can be more easily monitored and conditions more tightly controlled.

Cortically coupled computing is well suited to a variety of real-world applications ranging from large-scale rapid image search and annotation to automation in driving and control in precision tasks such as aircraft navigation. It exploits the concept of opportunistic sensing of the brain state, wherein the system captures the results of a user's implicit or explicit computation and then communicates this information to a traditional computer system via a neural interface, such as a noninvasive scalp EEG. This approach is fundamentally different from traditional BCIs that focus on using brain signals for direct control of computers and robotics.

In the near future, we envision numerous extensions of the technologies described here. For example, it is possible to extend the HAIL and C3V image analysis systems into more general cortically coupled data analysis systems to solve intractably complex analytical and computational problems. Using heterogeneous CV agents networked together with human analysts via a BCI, the systems would incorporate visual analytics techniques to recast complex problems into the human visual domain, implement multiagent systems to distribute the decomposed tasks across the network, and aggregate the results returned into high-confidence solutions. Such an analysis engine could provide new insights and capabilities to many fields by enabling advances in complex data analytics.

Alternatively, cortically coupled computing could be used to integrate human users with physical robots possessing heterogeneous capabilities for sensing and acting. For example, consider a situation where emergency personnel are attempting to rapidly assess a disaster area in conjunction with a swarm of small drones. If the drones could

## ABOUT THE AUTHORS

**SAMEER SAPROO** is a postdoctoral research scientist in the Department of Biomedical Engineering at Columbia University. His research interests include neural computation, neuroimaging, AI, and brain–computer interfaces (BCIs). Saproo received an MS in computer science from the University of California, Irvine, and a PhD in psychology from the University of California, San Diego. Contact him at ssaproo.ucsd@gmail.com.

**JOSEF FALLER** is a postdoctoral researcher in the Laboratory for Intelligent Imaging and Neural Computing at Columbia University. His research interests include BCIs and neuroimaging. Faller received a PhD in computer science from Graz University of Technology. Contact him at josef.faller@columbia.edu.

**VICTOR SHIH** is a graduate student in the Department of Biomedical Engineering at Columbia University. His research interests include machine learning, computational neural modeling, AI, and BCIs. Shih received an MS in biomedical engineering from Columbia University. Contact him at vs2481@columbia.edu.

**PAUL SAJDA** is a professor in the Departments of Biomedical Engineering, Electrical Engineering, and Radiology and director of the Laboratory for Intelligent Imaging and Neural Computing at Columbia University. Much of his current research focuses on using multimodal neuroimaging and behavioral measures to track selective attention and the dynamics of cognitive state during rapid decision making; he also applies these basic scientific findings to engineer neurotechnology systems that improve human–machine interaction. Sajda received a PhD in bioengineering from the University of Pennsylvania. He is a Fellow of IEEE, editor in chief of *IEEE Transactions on Neural Systems and Rehabilitation*, and chair of the IEEE Brain Initiative (see the sidebar). Contact him at psajda@columbia.edu.

**NICHOLAS R. WAYTOWICH** is a joint postdoctoral Research Fellow at the Laboratory for Intelligent Imaging and Neural Computing at Columbia University and the Human Research and Engineering Directorate at the US Army Research Laboratory. His research interests include machine learning, BCIs, human-autonomy integration, and adaptive systems. Waytowich received a PhD in biomedical engineering from Old Dominion University. He is a member of IEEE and the IEEE Systems, Man, and Cybernetics Society. Contact him at nick.waytowich@gmail.com.

**ADDISON BOHANNON** is a research scientist at the US Army Research Laboratory and a PhD candidate in applied mathematics, statistics, and scientific computing at the University of Maryland, College Park. His research interests include human–machine systems, intelligent systems, and scientific computing. Bohannon received an MS in applied mathematics, statistics, and scientific computing from the University of Maryland, College Park. He is a member of IEEE and the IEEE Systems, Man, and Cybernetics Society. Contact him at addisonb@math.umd.edu.

**VERNON J. LAWHERN** is a mathematical statistician in the Human Research and Engineering Directorate at the US Army Research Laboratory. His research interests include machine learning, statistical signal processing, and neurophysiological data mining for BCIs. Lawhern received a PhD in statistics from Florida State University. He is a member of IEEE. Contact him at vernon.j.lawhern.civ@mail.mil.

**BRENT J. LANCE** is a computer scientist in the Human Research and Engineering Directorate at the US Army Research Laboratory. His research interests include brain–machine interaction, human-autonomy integration, and machine learning. Lance received a PhD in computer science from the University of Southern California. He is a Senior Member of IEEE. Contact him at brent.j.lance.civ@mail.mil.

**DAVID JANGRAW** is a postdoctoral Fellow in the Section on Functional Imaging Methods at the National Institute of Mental Health. His research applies machine learning and novel analysis methods to physiological data from naturalistic paradigms, with current studies focusing on sustained attention and visual decision making. Jangraw received a PhD in biomedical engineering from Columbia University. Contact him at david.jangraw@nih.gov.

opportunistically sense what the users identify as important during their search, the drones could speed up the recovery operation by identifying similar regions or objects on a dynamic map. Ultimately, the utility of cortically coupled computing will be proved through systems that demonstrate how human–machine interaction in such a synergistic combination is more computationally powerful than the sum of the parts. ■

## ACKNOWLEDGMENTS

This work was funded by the Defense Advanced Research Projects Agency (DARPA), the Army Research Office (ARO) under grant number W911NF-11-1-0219, the Army Research Laboratory (ARL) under cooperative agreement number W911NF-10-2-0022, the Office of the Secretary of Defense Autonomy Research Pilot Initiative program MIPR DWAM31168, and the UK Economic and Social Research Council under grant number ES/L012995/1.

## REFERENCES

1. A.D. Gerson, L.C. Parra, and P. Sajda, "Cortically Coupled Computer Vision for Rapid Image Search," *IEEE Trans. Neural Systems and Rehabilitation Eng.*, vol. 14, no. 2, 2006, pp. 174–179.
2. P. Sajda et al., "In a Blink of an Eye and a Switch of a Transistor: Cortically Coupled Computer Vision," *Proc. IEEE*, vol. 98, no. 3, 2010, pp. 462–478.
3. A. Oliva and A. Torralba, "Building the Gist of a Scene: The Role of Global Image Features in Recognition," *Progress in Brain Research*, vol. 155, 2006, pp. 23–36.
4. P. Sajda, M.G. Philiastides, and L.C. Parra, "Single-Trial Analysis of Neuroimaging Data: Inferring Neural Networks Underlying Perceptual Decision-Making in the Human Brain," *IEEE Reviews in Biomedical Eng.*, vol. 2, 2009, pp. 97–109.
5. L. Parra et al., "Spatiotemporal Linear Decoding of Brain State," *IEEE Signal Processing Mag.*, vol. 25, no. 1, 2008, pp. 107–115.
6. D. Rosenthal et al., "Evoked Neural Responses to Events in Video," *IEEE J. Selected Topics in Signal Processing*, vol. 8, no. 3, 2014, pp. 358–365.
7. J. Wang et al., "Brain State Decoding for Rapid Image Retrieval," *Proc. 17th ACM Int'l Conf. Multimedia* (MM 09), 2009, pp. 945–954.
8. E.A. Pohlmeyer et al., "Closing the Loop in Cortically-Coupled Computer Vision: A Brain–Computer Interface for Searching Image Databases," *J. Neural Eng.*, vol. 8, no. 3, 2011; doi:10.1088/1741-2560/8/3/036025.
9. A.R. Marathe et al., "Confidence Metrics Improve Human-Autonomy Integration," *Proc. ACM/IEEE Int'l Conf. Human–Robot Interaction* (HRI 14), 2014, pp. 240–241.
10. C.-J. Ho, S. Jabbari, and J.W. Vaughan, "Adaptive Task Assignment for Crowdsourced Classification," *Proc. 30th Int'l Conf. Machine Learning* (ICML 13), 2013, pp. 534–542.
11. F. Parisi et al., "Ranking and Combining Multiple Predictors without Labeled Data," *Proc. PNAS*, vol. 111, no. 4, 2014, pp. 1253–1258.
12. B. Lance et al., "Brain–Computer Interface Technologies in the Coming Decades," *Proc. IEEE*, vol. 100, 2012, pp. 1585–1589.
13. D.C. Jangraw et al., "Neurally and Ocularly Informed Graph-Based Models for Searching 3D Environments," *J. Neural Eng.*, vol. 11, no. 4, 2014; doi:10.1088/1741-2560/11/4/046003.

RESEARCH FEATURE

# Real-Time Computing on Multicore Processors

**Lui Sha, Marco Caccamo, Renato Mancuso, Jung-Eun Kim, and Man-Ki Yoon,** University of Illinois at Urbana–Champaign

**Rodolfo Pellizzoni,** University of Waterloo

**Heechul Yun,** University of Kansas

**Russell B. Kegley and Dennis R. Perlman,** Lockheed Martin

**Greg Arundale and Richard Bradford,** Rockwell Collins

*Architects of multicore chips for avionics must define and bound intercore interference, which requires assuming a constant worst-case execution time for tasks executing on the chip. With the Single Core Equivalent technology package, engineers can treat each core as if it were a single-core chip.*

Although multicore technology has many benefits for real-time systems—among them, decreased weight and power, fewer cooling requirements, and increased CPU bandwidth per processor—multicore chips pose problems that stem from the cores interfering with one another when accessing shared resources. Interference is compounded in real-time systems, which are based on the assumption that worst-case execution time (WCET) is constant; that is, a software task's measured WCET must be the same whether that task executes alone or with other tasks. This assumption holds for single-core chips, but not for multicore chips unless they have isolation mechanisms between cores. Measurements we performed on a commercial multicore platform (Freescale P4080) revealed that a task's WCET can increase by as much as 600 percent when a task on one core runs with logically independent tasks in other cores.

Because of the potential for large and random delay spikes, the US Federal Aviation Administration (FAA), European Aviation Safety Agency (EASA), and Transport Canada specify that only single-core chips can be used unless intercore interference is specifically defined and handled.[1] Indeed, *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*, the primary document by which certification authorities such as the FAA, the EASA, and Transport Canada approve all commercial software–based aerospace systems, was developed for the certification of software in single-core computers.[2] With a single-core chip, architects can assume a constant WCET and can thus schedule tasks and partition resources without unanticipated delays. Hence, the ideal solution is to certifiably bound intercore interference in a multicore chip such that each core can be used as a single-core computer.

As part of studying the feasibility of such a solution, we developed the Single-Core Equivalent (SCE) technology package, which addresses interference problems that arise when cores concurrently access DRAM, the memory bus, shared cache resources, I/O resources, and the on-chip network. With SCE, each core can be used as if it were a single-core chip, allowing the timing analysis and certification of software in a core independently of software in other cores. This has implications for avionics

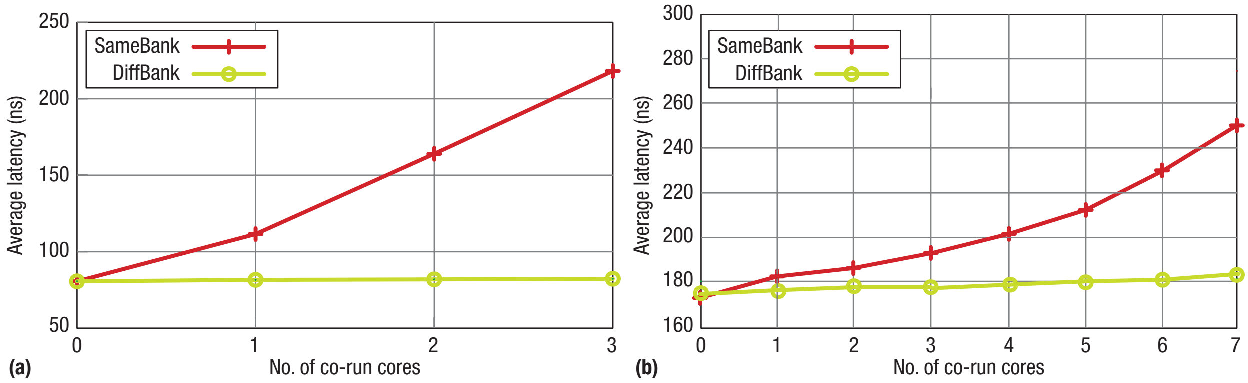


FIGURE 1. Effect of DRAM contention with a synthetic memory benchmark running on (a) the Intel Xeon and (b) the Freescale P4080 multicore chips. In the SameBank case, all cores access the same bank; in DiffBank, each core accesses a different private bank. Both graphs show that memory-access latency increases in the SameBank case as the number of cores running concurrently increases. The results imply that partitioning DRAM banks can reduce contention.

certification, as the D9178/B/C certification process targets avionics software in single-core computers.[2] With SCE, this process could work for multicore computers as well.

Our evaluations with the Freescale P4080, an eight-core chip, show that SCE successfully bounds intercore interference and removes unpredictable delay spikes.

## MEMORY-RELATED INTERFERENCE

Memory-related interference is caused by conflicts in accessing memory and the memory bus. To resolve these sources of interference we created PALLOC, an OS-level memory allocator, and MemGuard, an OS-level memory-bandwidth manager, as part of the SCE package. Together, they increase performance isolation for applications that share DRAM.

### Memory-access conflicts

DRAM is organized into ranks, banks, rows, and columns, with the greatest interference at the bank level. Figure 1 shows the average memory-access latency for a synthetic memory benchmark (linked-list traversal) when varying the number of interfering cores running the same benchmark. In SameBank, all cores access the same bank; in DiffBank, each core accesses a different private bank. In SameBank, memory-access latency increases as a function of the number of concurrently accessing cores. However, in DiffBank, memory-access latency is not affected by other cores' activities. These graphs are evidence that performance isolation improves when each core has its own set of dedicated DRAM banks.[3]

Current OSs do not control how memory pages are mapped onto DRAM banks, which leads to poor performance isolation and unpredictable memory performance. Given a sufficient number of DRAM banks, PALLOC allows applications in different cores to access disjoint sets of specific banks.[3]

### Memory-bus bandwidth

The synthetic memory benchmark associated with Figure 1 does not saturate the shared memory bus, so, when each contending core accesses a different bank, benchmark latency is barely affected. In contrast, multiple cores accessing memory-bus bandwidth can create a considerable bottleneck and thus increase interference. Because low-level memory arbitration policies in commercial hardware platforms are not known, we created MemGuard to manage memory bandwidth through a per-core bandwidth regulator for hard real-time applications.[4]

**Per-core regulator.** The per-core regulator monitors and enforces its corresponding allocation of core-memory bandwidth. Each regulator has a memory-access budget $Q_i$ for every regulation period $P$, which is global across cores. In an $M$-core chip, the sum of $M$ memory-bandwidth reservations is equal to the system's sustainable memory bandwidth. When the given budget is exhausted, the regulator calls the OS scheduler to suspend computation on that core. At the beginning of each $P$, MemGuard replenishes the budget in full and the OS resumes the suspended tasks. $P$ is a processor-wide parameter and shorter than the minimal application task period; currently, it is 1 ms. By restricting each core's maximum use of memory bandwidth, MemGuard effectively partitions memory bandwidth between cores and ensures strong performance isolation.

**Global-bandwidth reclaiming.** Bandwidth reservation alone could significantly waste available memory bandwidth because the core might not use all its reserved bandwidth and the reservable bandwidth that MemGuard can guarantee is much smaller than the hardware's peak bandwidth. To improve bandwidth use, MemGuard provides a bandwidth-reclaiming manager for soft real-time applications. At the beginning of the regulation period, the reclaiming manager estimates the cores' potential surplus bandwidth reservations and then redistributes on demand to the cores that need more bandwidth within the period. Available bandwidth is greater than guaranteed bandwidth, so if the cores collectively exhaust the guaranteed bandwidth before the period ends, MemGuard lets them use the additional available bandwidth.[4]

Figure 2 shows an example of how MemGuard impacts performance. In this experiment, we measured the performance of the 462.libquantum SPEC2006 benchmark, first alone (labeled run alone) and then with a memory hog program (labeled co-run). Without MemGuard, the benchmark's performance dropped more than 50 percent when the benchmark was co-scheduled with memory hog. When MemGuard reserved memory bandwidth (1,000 MBps for libquantum and 200 MBps for the memory hog), performance of the libquantum benchmark decreased but was not affected by the memory hog. When MemGuard enabled reclaiming and then shared the reclaimed bandwidth among cores that needed it, performance improved in both the run-alone and co-run cases.

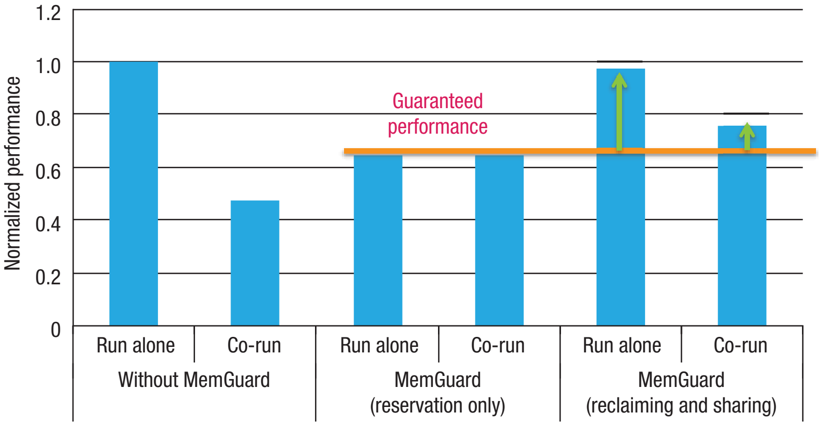


FIGURE 2. Performance impact of MemGuard. The *y*-axis shows the average instructions per cycle (IPC) for the 462.libquantum SPEC2006 benchmark when it runs alone (labeled run alone) and with memory-intensive co-runner (labeled co-run) in three different configurations: without MemGuard, MemGuard with only 1.0-GBps reservation, and MemGuard with both reservation and reclaiming.

## SHARED-CACHE INTERFERENCE

Modern CPUs feature at least one cache level organized as an associate set of a particular cache way. An associative set consists of cache lines with the same index. Depending on the running processes' addressing patterns, the cache controller loads data into the cache and writes data back from it in cache-line blocks. Each block can be loaded in any way and is chosen at fetch time according to the replacement policy. Once the cache way has been selected, the exact position inside the way depends on the value of a subset of the bits that compose the data address (index). Tag bits are used to detect hits, while offset bits are used to address a particular byte in a cache block. SCE's cache-management approach avoids intercore interference through offline profiling and online allocation.[5]

### Offline profiling

The memory-use profiler identifies the most frequently accessed virtual memory pages for a given executable. For each task, it produces a profile offline that ranks memory pages by access frequency. Because the address of a frequently used page is independent of its absolute virtual address, the profile is created only once, and virtual addresses are determined at runtime.

### Online allocation

Online allocation consists of page coloring and lockdown of the last-level cache. Colored lockdown, a process that takes place after page coloring and lockdown, is the result of modifying the Linux kernel's page-management algorithm to make online allocation transparent to the application.[6]

**Page coloring.** Multiple DRAM pages mapped to the same set of shared cache pages have the same color and can be allocated across cache ways. Our OS techniques can reposition task memory pages within the available colors to maximize allocation flexibility.

**Lockdown.** Because real-time applications are dominated by periodic execution flows with tight inner loops, it is possible to optimize use of the last-level cache by locking pages with the highest hit score first. In avionics applications that use the Integrated Modular Avionics (IMA) architecture, such pages can be preloaded in cache at the beginning of each IMA partition. The default configuration should evenly partition the last-level cache across cores.

**Colored lockdown.** Page coloring alone cannot guarantee that frequently accessed pages reside in cache, and, when lockdown is used alone, only a subset of frequently accessed pages can be allocated because, without coloring, there is no way to spread pages across multiple sets. Colored lockdown combines the two by first counting the number of frequently accessed pages with the same color. If, for a given color, the number of pages exceeds the number of available ways, the colored lockdown technique recolors extra pages into available sets and performs a lockdown on all the frequently accessed pages, including the recolored ones. With the combined approach, the total

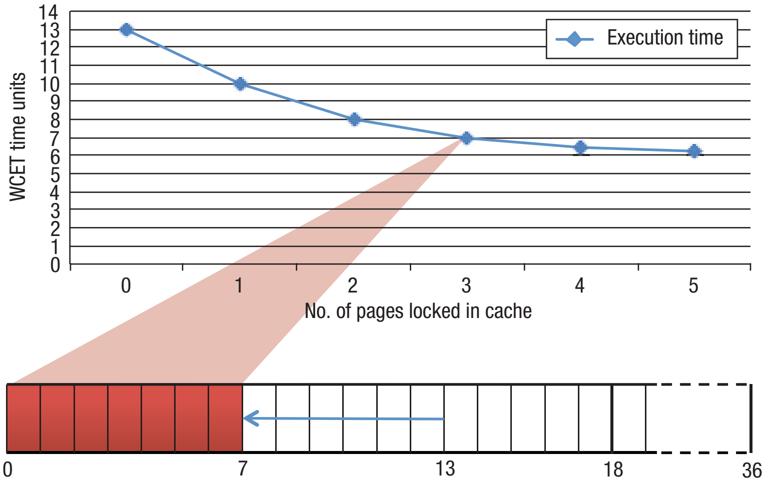


FIGURE 3. A lockdown curve of execution time when cache allocation is even, all but one core is idle, cores access private DRAM banks with PALLOC, and there is no MemGuard regulation. The task under analysis ($\tau_1$) can use up to peak memory bandwidth to retrieve data. When three pages are locked, worst-case execution time (WCET) drops from 13 to 7 time units.

number of pages that can be locked in cache equals the size of the cache allocated to a core.[2]

By exploiting per-task data acquired through profiling, we fit a progressive lockdown PL curve that models a task's WCET as a function of the number of memory pages locked in the last-level cache. Figure 3 shows a sample PL curve.

## INTEGRATED MODULAR AVIONICS

IMA is a common avionics architecture that uses time-division multiplex access (TDMA) to share a single-core computer's CPU cycles among applications, each of which uses a set of IMA time slots (partitions) in the IMA cycle.

SCE provides a solution to migrate applications residing in IMA partitions from single-core computers to fewer multicore computers. In IMA, there is one I/O partition for all application partitions' I/O activities, and IMA cycles in different single-core computers can have different lengths. Simply porting each IMA from a single-core computer to a core in a multicore computer is not the best approach because multiple I/O partitions in different cores could become active simultaneously, causing I/O conflicts in the shared I/O channel.[7] To avoid this problem, SCE uses the I/O core—a dedicated core that consolidates all I/O partitions. With the I/O core, IMA partitions in different cores can have different lengths for their major cycles, which makes I/O scheduling much easier. The I/O core has precedence relationships among physical and device I/O, shown in Figure 4.

As Figure 4 implies, I/O and processing partitions in the I/O core cannot overlap, but physical I/O transactions for different devices can be performed in parallel. To assign I/O and processing partitions, we developed the hierarchically ordered scheduling (HOS) heuristic algorithm.[7] HOS starts by randomly but partially assigning the offsets of all physical I/Os and processing partitions and then finds a complete solution by determining the offsets of all device I/O partitions. Once the physical I/O and processing partition offsets are fixed, the search space for the device I/O partition offsets can be represented as a set of periodic intervals, which reduces problem size considerably. The HOS algorithm quickly finds a solution on average and scales well with problem size.[7]

## SCE APPLICATION

The SCE application has two main stages. The first is to partition globally shared resources to create cores that are the equivalent of single cores. The second is to estimate each task's WCET. Once WCET is obtained, each partitioned task's schedulability analysis is the same as for a single-core chip.[8]

### Create single-core equivalence

Creating cores with single-core equivalence has five steps:

1. *Select the hardware*. The selected multicore chip must provide primitives that support last-level cache locking and the performance counters that MemGuard requires. For our experiments, we selected the P4080 chip.
2. *Ensure that each core has equal resources*. Each core should have private banks (PALLOC), an equal fraction of the memory bandwidth (MemGuard), and an equal amount of the last-level cache.
3. *Partition to allocate cores*. The allocation heuristics are well established for moving from slower single chips to a smaller number of faster single-core chips. We assume that the same heuristics apply in creating single-core equivalence.
4. *Allocate IMA partitions to cores*. Using the HOS heuristic, check if I/O channels can be partitioned temporally and meet all the precedence and capacity constraints. If not, return to the previous step. If there is no I/O solution, more or faster chips are needed. If there is an I/O solution, these four steps should yield a tentative set of virtual single-core chips.
5. *Optimize last-level cache partitioning and schedulability analysis*. Use colored lockdown to

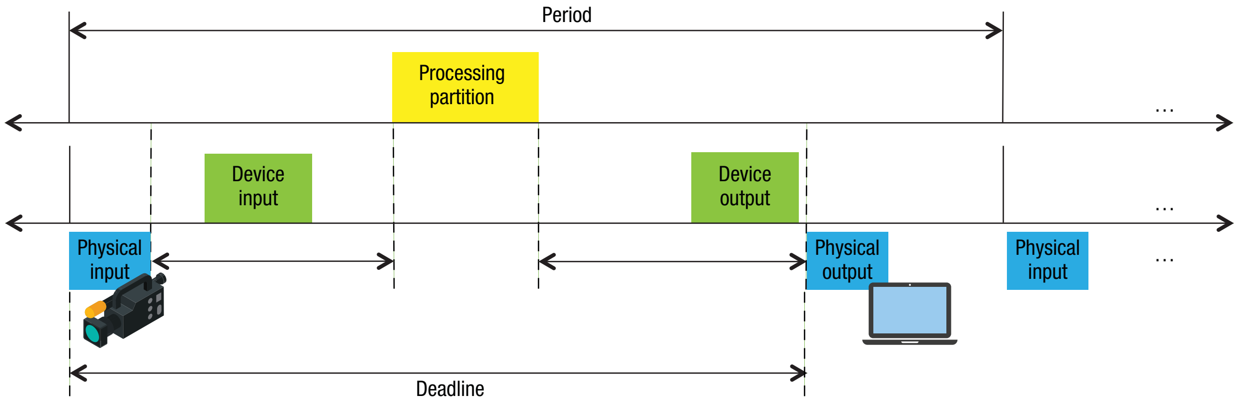


FIGURE 4. Partitions in the I/O core. The I/O core consolidates all I/O partitions by establishing precedence relationships among physical and device I/O. Each transaction is divided into physical and device I/O. Here, a camera taking pictures is the physical input. A device input transfers the buffered images to main memory, where an application within a core processes them. The device output—the processing result—is buffered in main memory and transferred to a physical output device; in this case, a monitor.

optimize the use of last-level cache partitions. For an *M*-core chip, in which $m \leq M$ cores will be used, estimate the WCET(*m*) of each task and check the schedulability of partitions and I/O transactions. If the schedulability check fails, go to step 3. If there is no solution, more and/or faster chips are needed.

## Estimate worst-case execution times

WCET is the foundation for schedulability analysis, which has been the focus of many analytic and experimental methods.[9] SCE can reuse the WCET estimation methods developed for single-core chips and use the estimate to equally partition shared resources. Thus, if only one of eight cores is used, the core will have all the shared resources, and WCET(1) will have the smallest WCET value. If applications never use more than two of the eight cores, it is possible to disable the remaining six cores and divide the total shared resources in half. For the same task, the shared resources then diminish with respect to the number of cores used, and WCET increases accordingly: WCET(1) < WCET(2) … < WCET(8).

However, in a multicore chip, shared resources must be partitioned or this monotonic relationship might not hold because random intercore interference can result in a WCET(8) that is less than a WCET(7). This random interference makes it impossible to estimate WCET(8) with any certainty and complicates certification. Without first partitioning shared resources and then isolating the partitions, software running on one core could interfere with another core's software timing behaviors, making modular core-by-core certification impossible.

SCE addresses this problem by partitioning shared resources according to the cores being used. It first makes all but one core idle and estimates a task's WCET(1) using traditional methods for a single-core chip. WCET(1) is then used to calculate WCET(*m*), where *m* represents the number of cores being used.

**WCET(*m*) and schedulability.** WCET(*M*) represents the maximum intercore interference when all cores are used in an *M*-core chip. The more cores in use, the smaller the share of partitioned resources for each core, and the greater the overhead from partition-management software, which lowers schedulability for each core. If only $m < M$ cores are needed for the application now and in the foreseeable future, the remaining $(M - m)$ cores can be disabled and the shared resources partitioned into *m* chunks.

For that reason, SCE computes WCET(*m*) instead of WCET(*M*). However, if an additional core is needed later on, SCE replaces WCET(*m*) with WCET(*m*+1) in each core's schedulability analysis.

Because WCET(*m*+1) is bigger than WCET(*m*), some applications in a core could become unschedulable and trigger the reallocation of applications to cores. In this case, recertification will likely be required, which is expensive. Hence, system engineers must carefully consider the ramifications of disabling cores in a multicore chip.

**Estimating WCET(*m*).** The first step in estimating WCET(*m*) is to look for last-level cache misses, which generate DRAM transactions. Let $S_{\text{line}}$ be the cache-line size, which is measured as the number of bytes transferred during each transaction. $S_{\text{line}}$ is architecture-specific and is provided in the chip specifications. Let $L_{\max}$ be the maximum delay on a DRAM transaction for the core under analysis. In analyzing worst-case delay, $L_{\max}$ is a key parameter and can be derived either experimentally or through DRAM analysis techniques.[9]

Experimentally, DRAM transfer bandwidth $BW_{\min}$ can be measured when two conditions hold: each memory transaction has a data dependency with the previous one and subsequent requests access different DRAM rows. Hence, $L_{\max}$ can be

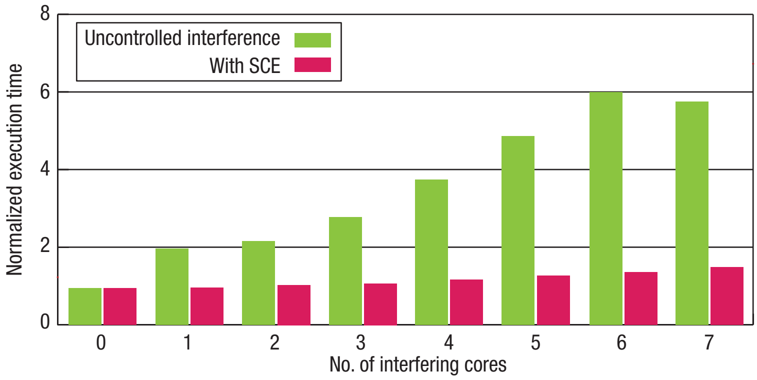


FIGURE 5. Effects of intercore interference for a single task running on the Freescale P4080 eight-core chip. The green bars are the worst-case finishing times (WCFTs) of a core 0 task when one to eight cores are used and intercore interference is uncontrolled. The red bars are the WCFTs of the same core 0 task when one to eight cores are used with the Single Core Equivalent (SCE) and a cache lockdown at 255 pages. WCFTs with SCE increase only slightly when more cores are used, because of increased isolation overhead and smaller size of shared resource partition. But the increase is much less dramatic, because random intercore interference through shared resources is removed. Data is from Lockheed Martin's space systems testbed when porting single-core computers' software to an eight-core Freescale P4080 computer.

derived from[5]

$$BW_{\text{min}} = \frac{S_{\text{line}}}{L_{\text{max}}}. \quad (1)$$

We use Stall to designate the task delay induced by MemGuard regulation and compute it as[5]

$$\text{Stall} = \frac{m\mu S_{\text{line}}}{BW_{\text{min}}}, \quad (2)$$

where $m$ is the number of active cores and $\mu$ is the maximum number of residual cache misses obtained from the PL curve. Each task's WCET($m$) includes the effect of bandwidth partitioning and is computed as Stall + C, where C is a task's WCET(1), for the desired amount of allocated cache.

Equation 2 also provides useful insights into how DRAM bandwidth allocation impacts a task's memory latency. $BW_{\text{min}}$ is clearly proportional to $m$ and $\mu$ as well as to $S_{\text{line}}$. Additionally, stall time is inversely proportional to $BW_{\text{min}}$. We performed a series of experiments that confirm Equation 2 provides a conservative estimate of a task's WCET($m$).[2]

An experimental way to estimate the WCET($m$) of a task in a particular core is to run tasks with worst-case interference in a nonstop loop in all cores.[10] We recommend using both methods and checking if the experimental measure is always smaller than the theoretical estimation, which makes pessimistic assumptions to ensure it will be a valid bound.

## PERFORMANCE EXAMPLES

We ran SCE technology in several experiments to validate its performance in bounding intercore interference. Figure 5 shows the reduction in delays of the same task running on P4080 with and without SCE.

When P4080 is used without SCE to control intercore interference, and seven of the eight cores were used, core 0 task's worst-case finishing time (WCFT) increased 600 percent relative to its WCFT when only one core was used. In addition, random interference caused WCFT to peak when seven of the eight cores were used, not when all eight were used. This counterintuitive pattern makes it hard to determine the worst-case scenario needed to compute WCET($m$) and complete certification.

### Results from sample scenario

Figure 6 shows the system-wide scheduling solution from one of our more comprehensive experiments involving four tasks and three cores. WCET(3) denotes the use of only three cores, core 0 represents the I/O core, and cores 1 and 2 are the application cores. Following our SCE methodology, we first applied PALLOC to ensure that each core had a private set of DRAM banks. We then allocated portions of the last-level shared cache evenly for each core, partitioned memory-bus bandwidth evenly using MemGuard, and applied the HOS heuristic to find an I/O scheduling solution.

Figure 6 shows how different SCE techniques integrate. Colored lockdown is performed at the beginning of each partition instance. MemGuard and PALLOC are statically configured to evenly partition DRAM resources among cores. Finally, I/O operations are globally serialized over the I/O core. Each physical input precedes the corresponding I/O core input operation, and each physical output follows the corresponding I/O core output operation. Similarly, each I/O core input operation precedes the partition instance that consumes the corresponding data, and each I/O core output operation precedes the partition instance that produces the corresponding data.

For example, in core 1, IMA partition 1 has a period of 18 and a reservation of 6 time units, while IMA partition 2 has a period of 36 and a reservation of 12 time units. After colored lockdown, inside partition 1, task 3 has period 18 and WCET(3) 3; task 4 has period 36 and WCET(3) 3. Similarly, tasks 1 and 2 are running inside partition 2 with periods

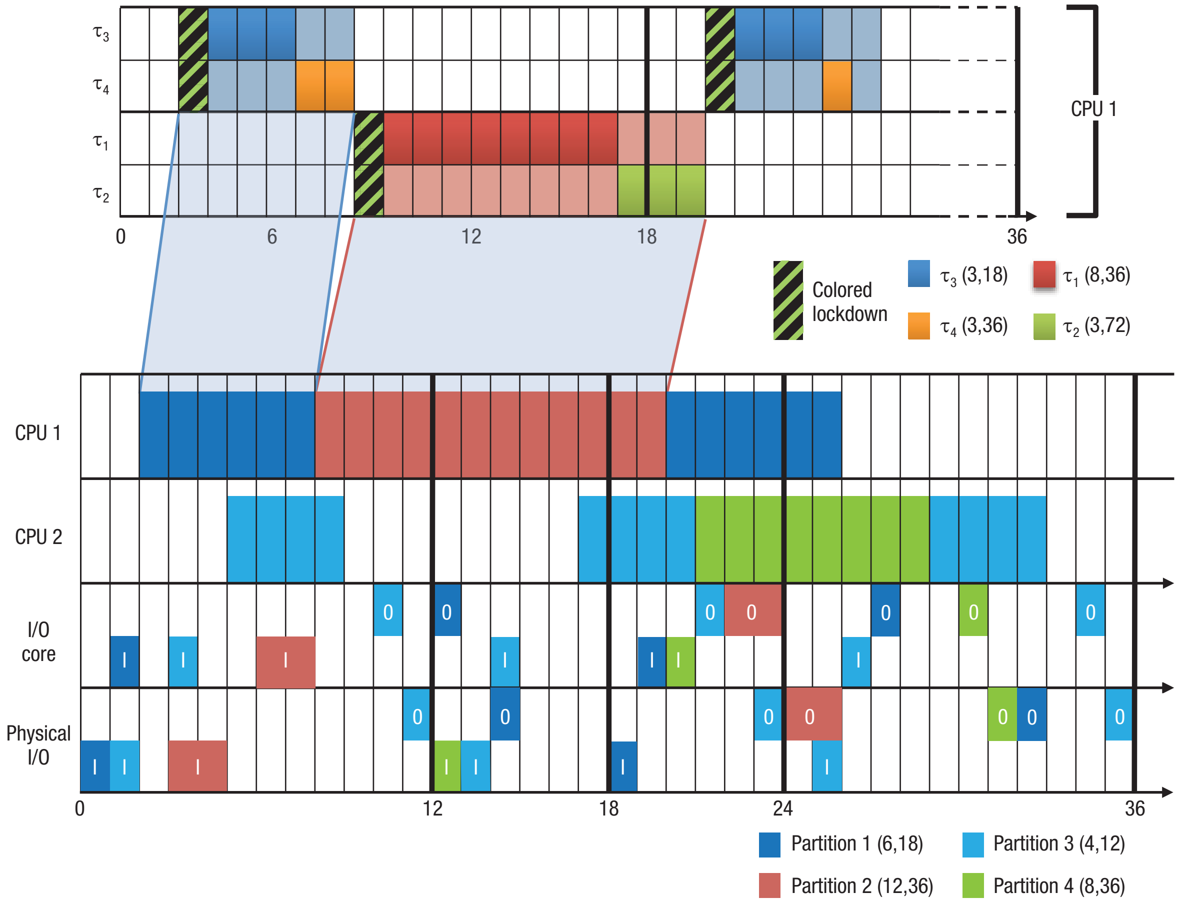


FIGURE 6. SCE solution for a three-core system running four tasks in which all SCE components are integrated.

36 and 72, respectively, and a WCET(3) of 8 and 3.

Making real-time computing efficient and predictable on multicore computers can greatly benefit many real-time applications, such as avionics, automotive, and medical-device control. It also enables avionics certification that has been problematic except when only using one core in a multicore chip. SCE provides a way to use established single-core estimation techniques to handle difficult schedulability analyses with multiple cores. We have already identified many areas for future work, including the certification of SCE itself.[11] An integrated solution to address memory consistency models, real synchronization protocols, and cache coherence protocols would benefit applications that must use multiple cores in parallel. Other topics for additional research are how to incorporate fault-tolerance mechanisms and address backward compatibility with existing software applications. C

## ACKNOWLEDGMENTS

The Single-Core Equivalent (SCE) technology was the result of a team effort. Sha led the architectural development and system integration. Pellizzoni and Zheng Pei Wu led the development of memory configuration and schedulability analysis. Caccamo and Mancuso led the development of last-level cache partitioning and hot code–segment lockdown. Yun and Gang Yao led the development of the mechanism to reserve memory bandwidth. Kim and Yoon led the development of the Integrated Modular Avionics (IMA) partition scheduling with conflict-free I/O for multicore systems. Kegley and Perlman guided the development and transition from the perspective of Lockheed Martin, and Arundale and Bradford provided guidance from the perspective of Rockwell Collins.

This work was sponsored in part by National Science Foundation grants CNS 13-02563 and CNS 12-19064, by Natural Sciences and Engineering Research Council Discovery Grant 402369-2011, by Office of Naval Research grant ONR N00014-12-1-0046, by Lockheed Martin grant 2009-00524, and by Rockwell Collins's grant RPS 645038. Finally, we thank Nancy Talbert for her great assistance in the revision of this article. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors.

## REFERENCES

1. Certification Authorities Software Team, "Position Paper CAST-32, Multicore Processors," rev. 0, 2014; www.faa.gov/aircraft/air_cert/design

_approvals/air_software/cast/cast_papers/media/cast-32.pdf.

2. L. Sha et al., *Single-Core Equivalent Virtual Machines for Hard Real-Time Computing on Multicore Processors*, tech. report, CS Dept., Univ. of Illinois at Urbana–Champaign, 2014; www.ideals.illinois.edu/handle/2142/55672.
3. H. Yun et al., "PALLOC: DRAM Bank-Aware Memory Allocator for Performance Isolation on Multicore Platform," *Proc. IEEE 19th Real-Time Technology and Applications Symp.* (RTAS 14), 2014, pp. 155–166.
4. H. Yun et al., "MemGuard: Memory Bandwidth Reservation System for Efficient Performance Isolation in Multicore Platforms," *Proc. IEEE 18th Real-Time Technology and Applications Symp.* (RTAS 13), 2013, pp. 55–64.
5. R. Mancuso et al., "WCET(*m*) Estimation in Multicore Systems Using Single Core Equivalence," *Proc. 27th Euromicro Conf. Real-Time Systems* (ECRTS 15), 2015, pp. 174–183.
6. R. Mancuso et al., "Real-Time Cache Management Framework for Multicore Architectures," *Proc. IEEE 18th Real-Time Technology and Applications Symp.* (RTAS 13), 2013, pp. 45–54.
7. J.-E. Kim et al., "Integrated Modular Avionics (IMA) Partition Scheduling with Conflict-Free I/O for Multicore Avionics Systems," *Proc. IEEE 38th Computer Software and Applications Conf.* (COMPSAC 14), 2014, pp. 321–331.
8. L. Sha, "Real-Time Virtual Machines for Avionics Software Porting and Development," *Real-Time and Embedded Computing Systems and Applications*, LNCS 2968, J. Chen and S. Hong, eds., Springer, 2004, pp. 123–135.
9. H. Kim et al., "Bounding Memory Interference Delay in COTS-Based Multicore Systems," *Proc. IEEE 19th Real-Time Technology and Applications Symp.* (RTAS 14), 2014, pp. 145–154.
10. R. Wilhelm et al., "The Worst-Case Execution Time Problem—Overview of Methods and Survey of Tools," *ACM Trans. Programming Languages and Systems*, vol. 7, no. 3, 2008, article no. 36.
11. L. Sha et al., "Position Paper on Minimal Multicore Avionics Certification Guidance," 4 Jun. 2016; https://1drv.ms/b/s!AqCnfGZqrIHshuIGAczwiyEUa5ZjTQ.

## ABOUT THE AUTHORS

**LUI SHA** is the Donald B. Gillies Chair professor of computer science at the University of Illinois at Urbana–Champaign (UIUC). His research interests include software certifiability for real-time systems, medical best-practice guidance systems, and aeronautic technology for space programs such as the Mars Pathfinder and International Space Station. Sha received a PhD in computer science from Carnegie Mellon University. He is a Fellow of IEEE and ACM, a member of the NASA Advisory Council's Aeronautics Committee, and a co-recipient of the 2016 IEEE Simon Ramo Medal. Contact him at lrs@illinois.edu.

**MARCO CACCAMO** is a professor in the Department of Computer Science at UIUC. His research interests include real-time OSs, real-time scheduling and resource management, wireless real-time networks, and quality-of-service control in next-generation digital infrastructures. Caccamo received a PhD in computer engineering from Scuola Superiore Sant'Anna. He is a Senior Member of IEEE. Contact him at caccamom@gmail.com.

**RENATO MANCUSO** is a doctoral student in computer science at UIUC's Real-Time and Embedded Systems Laboratory. His research interests include OS-level techniques for embedded systems to enhance predictability, real-time–oriented development of heterogeneous platforms, and the deployment of unmanned aerial vehicles. Mancuso received an MS in computer engineering from the University of Rome Tor Vergata. He is a member of IEEE. Contact him at rntmancuso@gmail.com.

**JUNG-EUN KIM** is a doctoral student in computer science at UIUC. Her research interests include real-time scheduling; safety-critical, real-time multicore architecture; and cyber-physical systems. Kim received an MS in computer science and engineering from Seoul National University. Contact her at jekim314@illinois.edu.

**MAN-KI YOON** is a doctoral student in computer science at UIUC. His research interests include secure embedded systems, multicore architecture, real-time scheduling, and machine learning. Yoon received a BS in computer science and engineering from Seoul National University. Contact him at mkyoon@illinois.edu.

**RODOLFO PELLIZZONI** is an assistant professor in the Department of Electrical and Computer Engineering at the University of Waterloo. His research interests include embedded architectures, real-time OSs, and timing analysis. Pellizzoni received a PhD in computer science from UIUC. He is a member of IEEE. Contact him at rpellizz@uwaterloo.ca.

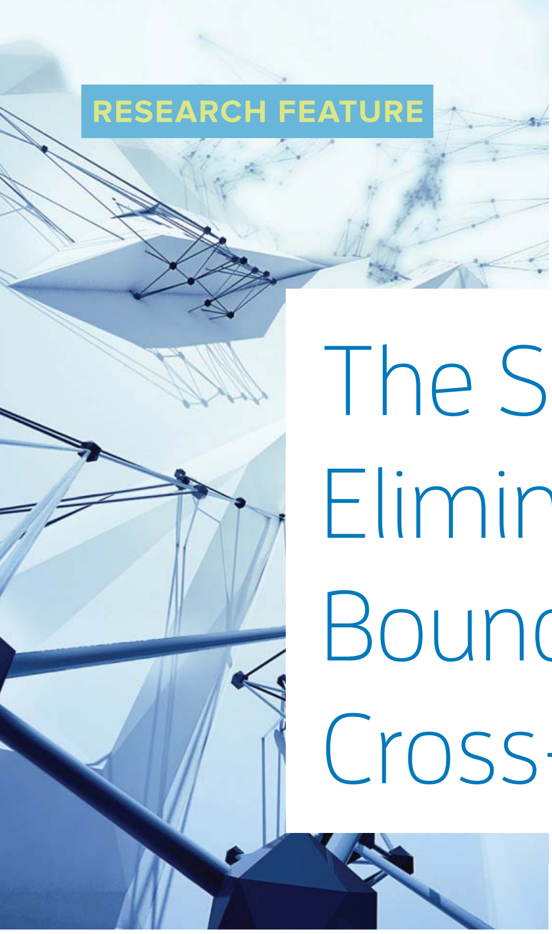
**HEECHUL YUN** is an assistant professor in the Department of Electrical Engineering and Computer Science at the University of Kansas. His research interests include embedded real-time systems, OSs, and computer architecture. Yun received a PhD in computer science from UIUC. He is a member of IEEE. Contact him at heechul.yun@ku.edu.

**RUSSELL B. KEGLEY** is a Lockheed Martin Fellow at Lockheed Martin's Aeronautics Company. His research interests include schedulability analysis, advanced design techniques, middleware architecture, multiprocessor use in avionic systems, and cache-contention modeling and remediation for fighter aircraft programs. Kegley received an MS in computer science from Mississippi State University. He is a member of IEEE and ACM. Contact him at russell.b.kegley@lmco.com.

**DENNIS R. PERLMAN** is a senior research engineer at Lockheed Martin's Space Systems Company. His research interests include the design, development, and implementation of hardware-in-the-loop simulation testbeds for spacecraft programs. Perlman received an MS in applied mathematics from the University of Colorado. Contact him at dennis.r.perlman@lmco.com.

**GREG ARUNDALE** is a principal systems engineer at Rockwell Collins's Advanced Technology Center. His research interests include next-generation avionics architectures. Arundale received a BS in computer engineering from UIUC. Contact him at greg.arundale@rockwellcollins.com.

**RICHARD BRADFORD** is a principal systems engineer at Rockwell Collins's Commercial Systems division. His research interests include real-time scheduling, network modeling and simulation, optimization, and engineering economics. Bradford received a PhD in operations research from Stanford University. Contact him at richard.bradford@rockwellcollins.com.



RESEARCH FEATURE

# The Spreadsheet Space: Eliminating the Boundaries of Data Cross-Referencing

**Massimo Maresca,** University of Genoa

*In the Spreadsheet Space, a virtual space for tabular data sharing, users in diverse administrative domains reference each others' spreadsheets at the cell level as well as share data from a variety of platforms and databases. Spreadsheet Space users enjoy the unlimited scope of cloud collaboration with the security of a more limited environment.*

Spreadsheets are heavily used in accounting, marketing, healthcare, engineering, and many other domains because of their various advantages. They do not require code-development skills, can be modifed to match an organization's culture, and provide a natural environment for model development. Domain experts can use spreadsheets at the level of complexity they need and rapidly obtain useful results.[1]

Spreadsheets support end-user computing, in which people focus on problem solving rather than on technology.[2] Although end-user computing can often be ad hoc and amateurish, spreadsheet development is a form of programming[3] and, as such, must be considered a professional activity subject to engineering rules—design, documentation, debugging, testing, maintenance, quality, and so on.[4]

A recent study estimated that the number of end-user programmers is more than an order of magnitude greater than the number of mainstream programmers.[5] The most widely used spreadsheet tool, Microsoft Excel, has been installed more than a billion times (http://money.cnn.com/2013/11/13/technology/enterprise/microsoft-office-google-docs). Although most users exploit only a fraction of Excel's functionalities, this installation number demonstrates that the spreadsheet phenomenon deserves careful attention and that spreadsheet research could have a strong impact on the way companies do business.

Spreadsheets have evolved from personal office tools that improve productivity to enterprise-level tools that support collaboration and analytics, and provide a simple and flexible framework for collaborative model development.[6] Spreadsheets have become the de facto standard for tabular data exchange over email and, with the advent of cloud computing, the reference tool for tabular data sharing.

To support analytics, spreadsheets are used routinely in the personalized postprocessing and presentation of

0018-9162/16/$33.00 © 2016 IEEE

THE SPREADSHEET SPACE'S VIRTUAL SPACE LETS USERS ACROSS PLATFORMS EXCHANGE CELL LINKS THAT SPREADSHEETS SUPPORT.

data exported by software platforms, as the ubiquitous "export to Excel" command attests. The trend toward big data has created an even stronger need for a smart combination of cloud and end-user computing as well as for effective techniques to move data between the cloud and data scientists' local machines.[7] To improve integration with software platforms, spreadsheets have recently incorporated features to support SQL queries,[8] such as Microsoft Power Query and Google Docs Query, and have been proposed for real-time data access and visualization[9] and data integration and mashup.[10]

However, none of these proposed features leverage the presence of cell references or links. To address that need, we at the Joint Research Center on Computer Platform Engineering (CIPI), established by the University of Genoa and the University of Padua, developed the Spreadsheet Space, a virtual space for tabular data exchange that exploits the typical permanent asymmetric cell links supported by spreadsheets. These links connect spreadsheets stored in different systems belonging to different users, crossing spreadsheet boundaries through the Internet with no limitations. The Spreadsheet Space leverages spreadsheet interconnection and composition technology[11] to open a set of opportunities for the development of new models for spreadsheet use.

To gauge user acceptance, we ran a series of field experiments in a variety of application domains. We then compared the Spreadsheet Space to other methods and established that it provides the power of cloud-based sharing while guaranteeing the security levels of desktop applications.

## SPREADSHEET SYNCHRONIZATION

The idea of connecting spreadsheets over the Internet derives from the observation that spreadsheets are based on cell references, and that these references are suitable pointers to multiple files, possibly owned by different users and located in different administrative domains. In the same way that internal spreadsheet references denote dependencies among spreadsheet cells, cross-spreadsheet references denote dependencies among cells of different spreadsheets. Because of cross-spreadsheet dependency, a cell update in a spreadsheet can trigger cell updates in other spreadsheets. Updating can be either direct, when spreadsheets reference the updated cell directly, or indirect, when spreadsheets reference the updated cell through intermediate spreadsheets.

### Required functionalities

Cross-spreadsheet dependency and direct and indirect updating are characteristics of spreadsheet synchronization. At first glance, enabling this synchronization would simply be a matter of extending the reference syntax, for example, by including a file name or URI to point to other spreadsheets. Unfortunately, the problem is more complex. First, cross-spreadsheet references could point to spreadsheets belonging to different users and hosted in administrative domains that are not accessible to the referencing user. Second, cross-spreadsheet references could point to spreadsheets located in devices that are not always on and thus not permanently connected to a network. Finally, spreadsheet cell updates rely on an event-management system that works only inside a single spreadsheet and not among spreadsheets.

Because of these issues, a platform the supports spreadsheet synchronization must include functionalities that

- allow spreadsheet users to grant read-access rights on their spreadsheets at the cell level to specific individuals they identify,
- allow spreadsheet users to create and maintain persistent copies of the cells made available for external reference, and
- trigger the update of a spreadsheet cell when the cell of another spreadsheet is updated.

### Supporting abstractions

The Spreadsheet Space provides the view and image abstraction to support these functionalities.

**View.** As Figure 1a shows, a *view* is a window on a user-created spreadsheet element. It is implemented as a persistent copy of that element, is constantly synchronized with the element, and is made available to a set of target users for external reference. Elements can be a cell, a cell range (a fixed-size 2D cell array), or a table (a record sequence associated with formulas that automatically adapts to extension or contraction).

The view abstraction directly supports read-access and persistency. A view is controlled by the spreadsheet owner, and is associated with a spreadsheet element rather than an entire spreadsheet. In the Spreadsheet Space, the view concept is different from the accepted idea of view in database theory. In that context, a view is associated with the result of a predefined SQL

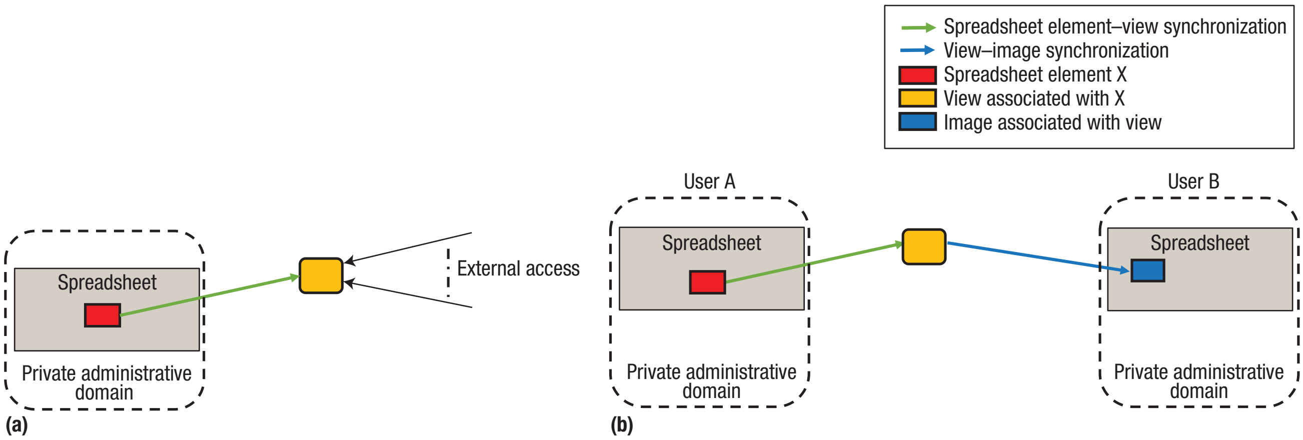

Spreadsheet element–view synchronization
View–image synchronization
Spreadsheet element X
View associated with X
Image associated with view
Spreadsheet
Private administrative domain
External access
(a)
User A
Spreadsheet
Private administrative domain
User B
Spreadsheet
Private administrative domain
(b)

FIGURE 1. Views and images. (a) A view is associated with a spreadsheet element (red rectangle), which can be a cell, cell range, or table. The associated view (yellow rectangle) is constantly synchronized with the element (green arrow) and is available for external access. (b) An image is associated with a view (dark blue box). Spreadsheets are synchronized through a combination of synchronizing spreadsheet elements and views (green arrow) and synchronizing views and images (blue arrow).

query, while in the Spreadsheet Space, a view is a window on a data set that can evolve over time.

**Image.** An *image* is a read-only local copy of an external view of a spreadsheet element. As Figure 1b shows, an image maintains alignment with the corresponding view and, through it, with the spreadsheet element associated with that view. Users create images of remote views in spreadsheets and then reference cells belonging to other spreadsheets through internal references to image cells. Spreadsheet synchronization is the result of combining the synchronization of spreadsheet element and view (aligning views with the associated spreadsheet elements) and the synchronization of view and image (aligning the images with their corresponding views). An external control system orchestrates the combined action of these two synchronizations, and thus satisfies the third required functionality (triggering the update of a cell when a cell linked to it is updated).

A view can be associated with any tabular data, not just from spreadsheets. Consequently, any software platform can create views, and spreadsheets can include images of views created by software platforms and maintain alignment with the software platform's data. Because they essentially become clients of that software platform, spreadsheets of different vendors that use the platform can interoperate.

### Operational primitives

View-image operation is based on a control plane, which is in charge of view–image creation, and on a data plane, which is in charge of view–image synchronization. Control-plane primitives include

- *expose*—through which a user creates a view of a spreadsheet element, which is accessible to everybody or to a set of target users; and
- *join*—through which a user creates a local image of a view.

Data-plane primitives include

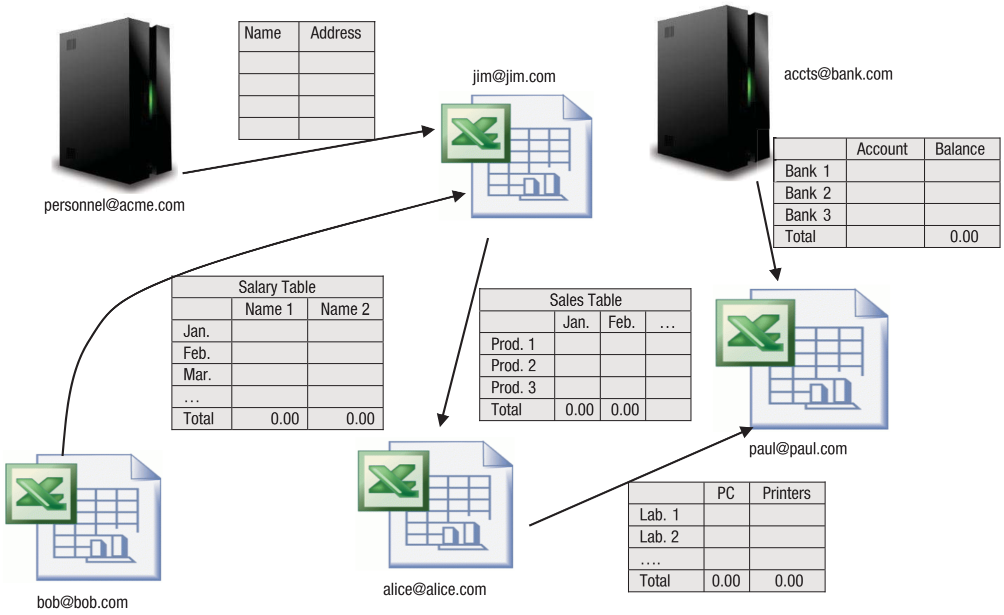
- *update view*—through which a spreadsheet uploads the current content of a spreadsheet element exposed through a view, and
- *update image*—through which a spreadsheet downloads the current content of a view to the corresponding image.

Exposing and joining views always requires explicit user intervention, but synchronization can be configured to be manual or automatic. In manual synchronization, users explicitly control the timing of view and image updates, while in automatic synchronization, updating is immediate. Thus, the manual mode supports data version control, while the automatic mode supports real-time data distribution.

## SPREADSHEET OVERLAY

Through these operational primitives, users can directly activate the Spreadsheet Space functions. However, in workgroup collaboration, spreadsheet users often do not take the initiative to interact with other users or to retrieve data from software platforms unless they are requested to do so by the third-party coordinating collaboration. The Spreadsheet Space supports this reliance on a third party through the *spreadsheet overlay* (SO), which uses the *form* abstraction—a formatted cell range that a spreadsheet user is requested to fill out and expose to other users as a view. Forms are prepared using spreadsheets and submitted to other spreadsheets through the Spreadsheet Space.

An SO, an example of which is shown in Figure 2, is a complete specification of a spreadsheet-based collaboration scheme that a coordinator creates and deploys in the Spreadsheet Space. In abstract terms, it is a directed graph in which nodes are SO participants (user spreadsheets, software platforms, or



| Name | Address |
|---|---|
| | |
| | |
| | |
| | |

| | Account | Balance |
|---|---|---|
| Bank 1 | | |
| Bank 2 | | |
| Bank 3 | | |
| Total | | 0.00 |

| Salary Table | | |
|---|---|---|
| | Name 1 | Name 2 |
| Jan. | | |
| Feb. | | |
| Mar. | | |
| … | | |
| Total | 0.00 | 0.00 |

| Sales Table | | | |
|---|---|---|---|
| | Jan. | Feb. | … |
| Prod. 1 | | | |
| Prod. 2 | | | |
| Prod. 3 | | | |
| Total | 0.00 | 0.00 | |

| | PC | Printers |
|---|---|---|
| Lab. 1 | | |
| Lab. 2 | | |
| …. | | |
| Total | 0.00 | 0.00 |

**FIGURE 2.** Spreadsheet overlay. A set of spreadsheet users—Bob, Alice, Jim, and Paul—are connected to each other as well as to two software platforms, personnel and accts. The arrows denote the relationships among users and between users and spreadsheet platforms. Each arrow is associated with a *form*, which describes the data structure that the source node makes available to the destination node.

both) and edges are forms. An SO's life cycle consists of the creation, deployment, and operational stages.

To create an SO, the collaboration coordinator identifies the participants, prepares the forms that they need to fill out and exchange, and uses a graphical creation environment to enter an SO description similar to the one in Figure 2. To deploy an SO, the software component in charge of deployment sends an invitation to prospective participants, attaching the forms that each participant has been requested to fill out or receive. Once all participants have confirmed their participation and have configured their spreadsheets to participate, the SO enters the operational stage, during which synchronization takes place. The coordinator monitors deployment and operation through a software component that logs all data exchanges and presents the SO's status adding dynamic annotations to the SO source description.

## SPREADSHEET SPACE PLATFORM

The Spreadsheet Space relies on the distributed system shown in Figure 3a, which includes the server and client software components. The server takes care of persistency and synchronization; the clients are associated with spreadsheets and software platforms so that they can participate in the Spreadsheet Space.

Figure 3b shows how the control-plane and the data-plane components interact in a client and server. A possible control-plane interaction is one in which a source user exposes a view. The source client then uploads the view to the view repository, and uses the event manager to issue a "new view" event to target clients, which is essentially an invitation to join the view. (A view could also be exposed to the public instead of to specific users. In that case, update notifications would require a user subscription.) The target clients receive the event notification, download the view from the view repository, and issue "view joined" events to the source client through the event manager.

A possible data-plane interaction is one in which a source client updates an existing view. The source client uploads the update to the view repository, and uses the event manager to issue a "view updated" event to the target clients to invite them to refresh their images. The target clients receive the event notification, download the update, and issue an "image updated" event to the source client through the event manager to signal that they have acquired the update.

## USER-ACCEPTANCE EVALUATION

The Spreadsheet Space is available as a free Internet service (www.spreadsheetspace.net). To test user acceptance, we selected a set of application domains, identified appropriate partner organizations, teamed with the partner organizations' experts to design use cases of interest, and performed several experiments. The experiments' main goal was to detect acceptance problems stemming from a new interaction paradigm.[12]

Table 1 lists the use cases in the field experiments, which together involved nearly 100 Excel users. The experiments led to the identification of three key factors to successful user acceptance: known interaction patterns, trust, and a sense of security.

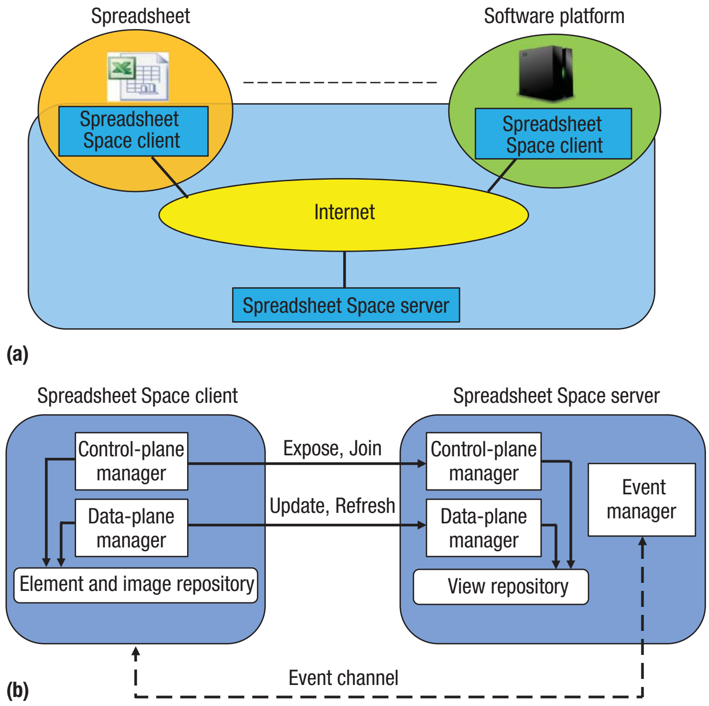


FIGURE 3. The Spreadsheet Space platform. (a) Clients (spreadsheets and software platforms) and servers interact in the Spreadsheet Space through the Internet. (b) Control-plane and data-plane managers create, configure, remove, and update views and images in the element and image repository, and the event manager orchestrates operations between the client and server. The client activates all updates.

### Known interaction patterns

The feedback from participants in the experiments confirmed our user-discomfort hypothesis, but we gained insight into acceptable interaction patterns. Although users were reluctant to accept unstructured spreadsheet interconnection, they were inclined to accept simple and structured interaction patterns that match known user relationships, such as those shown in Figure 4. However, in the Spreadsheet Space such user relationships refer to spreadsheet synchronization rather than to the traditional data-exchange context.

### Trust

Participants wanted a simple and friendly user interface that they could permanently control. The integration of enterprise services makes it easier for users who already trust their enterprise system to extend that trust to the Spreadsheet Space.

### Security

Although the Spreadsheet Space offers end-to-end encryption, most partner organizations required the deployment of a Spreadsheet Space server in their private cloud, rather than as a service in a public cloud. The goal was to enforce data protection, and the feedback strongly implied that the Spreadsheet Space should evolve from a single space to a federation of private Spreadsheet Spaces.

## COMPARISON WITH OTHER MECHANISMS

We were also interested in how the Spreadsheet Space compared with other interconnection mechanisms both among spreadsheets and between spreadsheets and software platforms.

### Spreadsheet interconnection

Table 2 lists the criteria we used to evaluate desktop and cloud spreadsheets against the Spreadsheet Space and the comparison results. We used Excel as the de facto standard for desktop spreadsheets and Google Docs and Microsoft Excel Online as the reference systems for cloud spreadsheets.

Desktop Spreadsheets reference external spreadsheets by path name—for example, `=’[Spreadsheet_Path-Name] Sheet_Name’!Sheet_Element` in Excel. Cloud spreadsheets use a URL—for example, `=importrange(“Spreadsheet_URL”, “Sheet_Name! Sheet_Element”)` in Google Docs. As Table 2 shows, some limitations in desktop and cloud spreadsheets significantly restrict the use of cross-spreadsheet references. The Spreadsheet Space does not have these restrictions.

**Scope.** In desktop spreadsheets, the local path-name nature limits the scope of cross-spreadsheet references to the file system, while in cloud spreadsheets, which are based on URLs, scope is unlimited. In the Spreadsheet Space, scope is also unlimited because of the nature of view as a global reference to a private spreadsheet element.

**Propagation.** Data propagation in spreadsheets relies on a publish/subscribe mechanism through which cells subscribe to other cell updates. In desktop spreadsheets, the application provides this mechanism, which means that cross-spreadsheet data can be propagated only within the same application instance. In cloud spreadsheets, propagation is possible only within the cloud provider’s perimeter, such as within Google. However, in the Spreadsheet Space, the propagation of cross-spreadsheet data has no limitations because the Spreadsheet Space platform provides a central publish/subscribe mechanism.

**TABLE 1.** Use cases employed in Spreadsheet Space evaluation experiments.

| Domain | Partner profile | Use case |
|---|---|---|
| Personal | Stock exchange | Portfolio management: integrate financial information provided by multiple sources (bank, stock exchange, and so on) in the desktop |
| Business | University | Budget consolidation: consolidate research-group budgets in global department budget |
| Marketing | Enterprise | Product-list management: synchronize regional product lists with the company's central product list |
| Public administration | Intermodal hub | Open data: publish and process freight-traffic information |
| Accounting | Accountant | Balance reconciliation: reconcile balances of subsidiary companies that belong to the same holding |
| Big data | Bank | Hadoop spreadsheet integration: interactively analyze Hadoop query results |

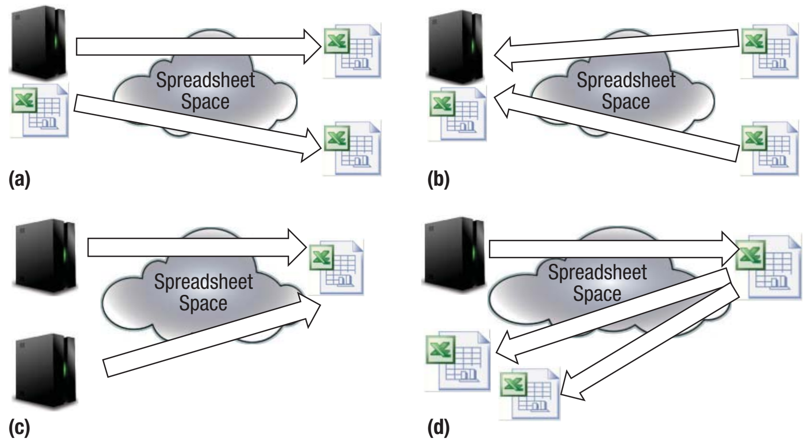


**FIGURE 4.** Regular interaction patterns in the Spreadsheet Space. (a) *Data distribution*, in which spreadsheets maintain synchronization with a data source that exposes data through views. (b) *Data collection*, in which a data collector maintains synchronization with spreadsheets that expose data through views. (c) *Enterprise application integration*, in which spreadsheets synchronize with software platforms and merge data in the desktop. (d) *Data distribution chain*, in which information propagates from a software platform through intermediate spreadsheets to an end user's spreadsheets.

**Availability.** Desktop spreadsheets are unavailable when the machines on which they reside are disconnected. By virtue of their residence in the cloud, cloud spreadsheets are permanently available. The Spreadsheet Space makes desktop data permanently available by maintaining views as persistent copies of spreadsheet elements in a permanently available server.

**Privacy.** Desktop spreadsheets reside in end-user systems, which provide maximum privacy, while cloud spreadsheets reside and are processed in the cloud provider's servers. Although cloud providers must comply with strict information-protection policies, geographically dispersed information is an inherent threat. The Spreadsheet Space implements view–image synchronization over encrypted end-to-end channels, and all processing takes place in end systems. Consequently, privacy is supported to the same degree as with desktop spreadsheets.

**Granularity.** Desktop and cloud spreadsheets manage read-access rights at the spreadsheet level. The Spreadsheet Space manages read-access rights down to the cell level, allowing spreadsheet elements to be exposed (cells, cell ranges, and tables) for external reference while protecting the rest of the spreadsheet.

**Naming.** Cross-spreadsheet references in both desktop and cloud spreadsheets include either coordinates or names defined in the source spreadsheet. Thus, to configure cross-spreadsheet references, target users must be aware of the source spreadsheet's internal structure. In the Spreadsheet Space, in contrast, view creation includes the generation of an external object, which the target users refer to when configuring cross-spreadsheet references.

## Spreadsheet–platform interconnection

At present, interconnection between spreadsheets and software platforms is supported by native spreadsheet mechanisms, such as Microsoft Excel Connections and Google Docs Query, as well as by spreadsheet extensions based on additional software components, such as add-ins (Microsoft) and add-ons (Google). We focus our comparison on native mechanisms because add-ins and add-ons are concerned with specific application requirements, not spreadsheet interconnection in general.

The difference between native spreadsheet mechanisms and the Spreadsheet

**TABLE 2.** Comparison of desktop and cloud spreadsheet interconnection mechanisms with the Spreadsheet Space according to six criteria.

| Criterion | Definition | Desktop | Cloud | Spreadsheet Space |
|---|---|---|---|---|
| Scope | Boundaries of cross-spreadsheet references | Limited to file system | Unlimited | Unlimited |
| Propagation | Boundaries of cross-spreadsheet data propagation | Limited to application instance | Limited to cloud provider | Unlimited |
| Availability | Accessibility of spreadsheets over time | Limited to desktop connected | Permanent | Permanent |
| Privacy | Guarantee that unencrypted data never leaves end-user systems | Full privacy support | Under cloud provider control | Full privacy supported |
| Granularity | Possibility to grant read–access rights to spreadsheet elements | Coarse (spreadsheet level) | Coarse (spreadsheet level) | Fine (down to cell level) |
| Naming | Support of an external namespace | No support | No support | Support provided |

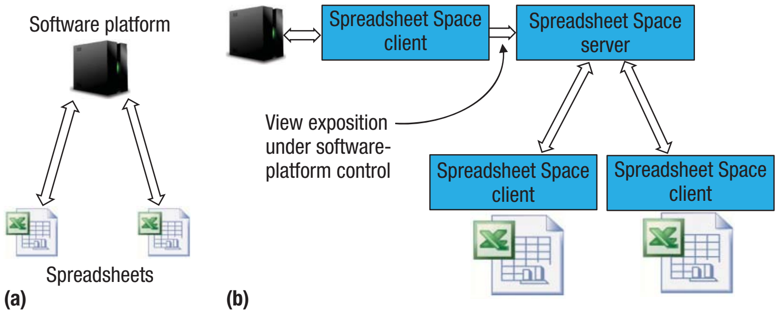


**FIGURE 5.** Decoupling of spreadsheets and software platforms. (a) In a native system, spreadsheets act as platform clients. (b) In the Spreadsheet Space, data is exposed as a view, and spreadsheets access data through the view.

Space is that while native mechanisms implicitly assume that spreadsheets act as clients of software platforms, as shown in Figure 5a, the Spreadsheet Space requires that software platforms expose data as views and that spreadsheets access the exposed data through views, as shown in Figure 5b. The decoupling of spreadsheets and software platforms has scalability, security, and control benefits.

**Scalability.** Decoupling improves scalability by eliminating direct spreadsheet access to software platforms. In the Spreadsheet Space, spreadsheets do not directly access software platforms, so the platform load is a function of the number of views, not the number of clients. The load thus moves from the software platform, where it is critical, to the Spreadsheet Space, where it is not.

**Security.** Security improves because Spreadsheet Space authentication is decoupled from software platform authentication. Because spreadsheets never access software platforms directly, they need only Spreadsheet Space credentials for authentication, not software platform credentials. The improvement is significant because software platform access is typically critical, and, consequently, the widespread distribution of software platform credentials is insecure.

**Control.** Because spreadsheets are not dependent on the software platform, they are not subject to the platform provider's restrictions. Both platform operators and data scientists have the power to configure views, to associate them to client spreadsheets, and to monitor and revoke access.

The Spreadsheet Space began as a way to extend internal spreadsheet references to cross-spreadsheet references over the Internet. Work has since evolved to spreadsheet synchronization—examining the feasibility of synchronizing spreadsheets and software-platform data over encrypted channels regardless of where they are located. The ultimate goal is to provide global reachability and information confidentiality at the same time.

Although compelling, the Spreadsheet Space is nothing but an infrastructure waiting for widespread adoption, which could have implications for data localization, ownership, versioning, and auditability, as well as for business processes and interactivity in general. The challenge is to develop appropriate models and tools able to accompany the deployment of the Spreadsheet Space in organizations and to take advantage of the opportunities it provides. If successful, these models and tools will be the real innovation, while the Spreadsheet Space will be just an enabler. C

## ABOUT THE AUTHOR

**MASSIMO MARESCA** is a professor of computer engineering at the University of Genoa, head of the Scientific Office at the Italian Consulate in San Francisco, and a visiting scholar at both the University of California, Berkeley, and the International Computer Science Institute. His research interests include distributed computing, networking, computing services, and computer-supported cooperation. Maresca received a PhD in computer engineering from the University of Genoa. He is a member of IEEE. Contact him at massimo.maresca@unige.it.

### REFERENCES

1. T. Grossman, "Spreadsheet Engineering: A Research Framework," *Proc. European Spreadsheet Risks Interest Group Symp.*, 2002, pp. 21–34.
2. B.A. Nardi, *A Small Matter of Programming: Perspectives on End User Computing*, MIT Press, 1993.
3. A.J. Ko et al., "The State of the Art in End User Software Engineering," *ACM Computing Surveys*, vol. 43, 2011, pp. 21–44.
4. S. Erwig, "Software Engineering for Spreadsheets," *IEEE Software*, vol. 26, no. 5, 2009, pp. 25–30.
5. C. Scaffidi, M. Shaw, and B. Myers, "Estimating the Number of End Users and End User Programmers," *Proc. IEEE Symp. Visual Languages and Human Centric Computing* (VLHCC 05), 2005, pp. 207–214.
6. H. Waqar and T. Clear, "Spreadsheets as Collaborative Technologies in Global Requirements Change Management," *Proc. IEEE 9th Int'l Conf. Global Software Eng.* (ICGSE 14), 2014, pp. 74–83.
7. D. Fisher et al., "Interactions with Big Data Analytics," *ACM Interactions*, vol. 19, no. 3, 2012, pp. 50–59.
8. J. Cunha et al., "Embedding Model-Driven Spreadsheet Queries in Spreadsheet Systems," *Proc. IEEE Symp. Visual Languages and Human-Centric Computing* (VL/HCC 14), 2014, pp. 151–154.
9. K.S.P. Chang and B. Myers, "A Spreadsheet Model for Handling Streaming Data," *Proc. 33rd ACM Conf. Human Factors in Computing Systems* (CHI 15), 2015, pp. 3399–3402.
10. Z. Obrenovic and D. Gasevic, "End-User Service Computing: Spreadsheets as a Service Composition Tool," *IEEE Trans. Services Computing*, vol. 1, no. 4, 2008, pp. 229–242.
11. S. Mangiante, M. Maresca, and L. Roncarolo, "SpreadComp Platform: A New Paradigm for Distributed Spreadsheet Collaboration and Composition," *Proc. 8th Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing* (CollaborateCom 12), 2012, pp. 297–305.
12. G. Currie and M. Kerrin, "The Limit of a Technological Fix to Knowledge Management: Epistemological, Political, and Cultural Issues in the Case of Intranet Implementation," *Management Learning*, vol. 35, no. 9, 2012, pp. 9–29.

SOCIAL COMPUTING

# Social Computing for Mobile Big Data

**Xing Zhang,** Beijing University of Technology and Beijing University of Posts and Telecommunications

**Zhenglei Yi,** Beijing University of Posts and Telecommunications

**Zhi Yan,** Hunan University

**Geyong Min,** University of Exeter

**Wenbo Wang,** Beijing University of Posts and Telecommunications

**Ahmed Elmokashfi and Sabita Maharjan,** Simula Research Laboratory

**Yan Zhang,** Simula Research Laboratory and University of Oslo

*Mobile big data contains vast statistical features in various dimensions, including spatial, temporal, and the underlying social domain. Understanding and exploiting the features of mobile data from a social network perspective will be extremely beneficial to wireless networks, from planning, operation, and maintenance to optimization and marketing.*

Data services' exponential growth, and the constantly expanding wireless and mobile applications that use them, have ushered in an era of big data. This mobile big data—from applications such as planning, operations and maintenance, optimization, and marketing—poses many new challenges to conventional data analytics because of its large dimensionality, heterogeneity, and complex features. To help address this problem, we provide a classification structure for mobile big data and highlight several

0018-9162/16/$33.00 © 2016 IEEE

EDITOR CHRISTIAN TIMMERER
Alpen-Adria-Universitat Klagenfurt;
christian.timmerer@itec.aau.at

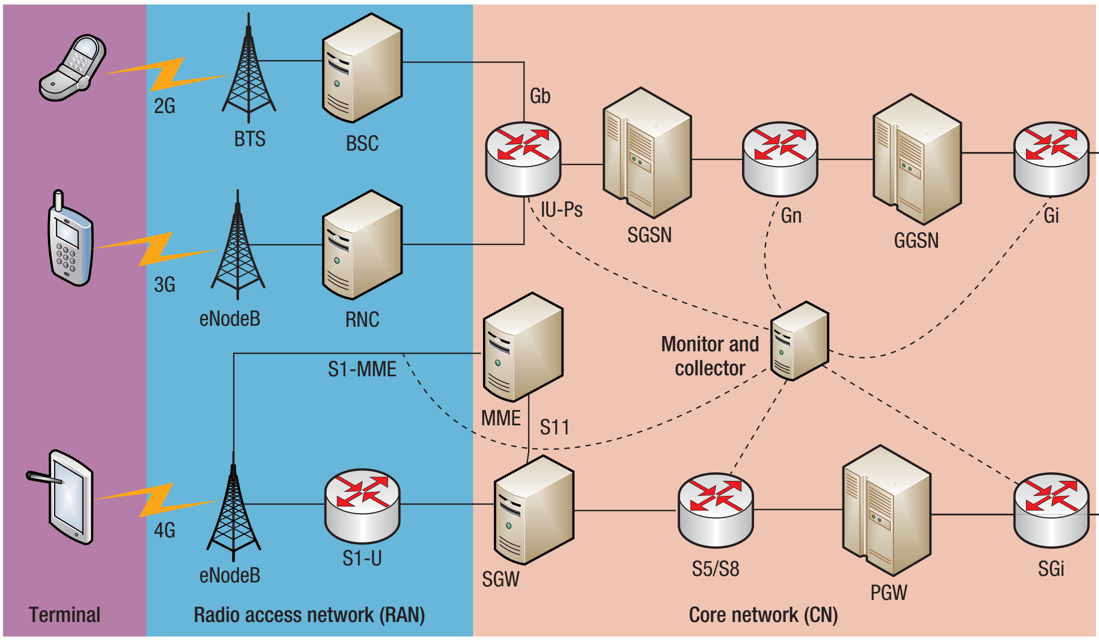


**Figure 1.** Mobile cellular network architecture and data-collecting points. BSC: base station controller; BTS: base transceiver station; eNodeB: Evolved Node B; GGSN: Gateway GPRS Support Node; IU-Ps: Packet Switched Core Network; MME: Mobility Management Entity; PGW: Packet Data Network Gateway; RNC: Radio Network Controller; SGi: Serving Gateway interface; SGSN: Serving GPRS Support Node; SGW: Serving Gateway.

research directions from the perspective of social computing using a significant volume of real data collected from mobile networks.

Conceptually, *big data* refers to not only a large volume of data but also other features that add complexity to it, thus giving it several unique characteristics and differentiating it from huge amounts of data.

## DATA CATEGORIES IN MOBILE CELLULAR NETWORKS

Vast amounts of mobile data are collected and extracted from several key network interfaces, in both radio access networks and core networks, as shown in Figure 1. This data can be roughly classified into four categories: flow record data, network performance data, mobile terminal data, and additional data information.

### Flow record data

A *flow* is a collection of packets between two end nodes defined by specific attributes, such as the well-known TCP/IP five-tuples. The flow record data in cellular networks is typically obtained through deep packet inspection and deep flow inspection, which includes both data records and signaling records, in the form of XDR (call/transaction detail record). These data records contain the main attributes during a data-connection session. The flow record is perhaps the most important data for describing user behavior. For a medium-sized city (1 million subscribers), the total flow record data is about 70 Tbytes (approximately $10^{15}$ records) in one week.

### Network performance data

Network performance data refers to aggregated datasets collected during a certain period (for example, 5 minutes, 15 minutes, or an hour). Network performance data mainly includes the key performance indicator (KPI) data and the measurement report (MR; a statistical data report that contains information about channel quality). KPIs are widely used in mobile cellular networks with the goal of evaluating the network performance and quality of service delivered to users. Network performance data can grow to several Tbytes in one week for a medium-sized city.

### Mobile terminal data

With the development of the Internet of Things (IoT) and 4G cellular networks, a rapidly increasing number of connected devices are generating a large amount of data at terminals. Two typical categories of mobile terminal data are smartphones and the IoT. Terminal data can be collected through a mobile app. The data contains service characteristics, device information, and wireless network parameters such as international mobile subscriber identity, cell ID, signal strength, and download/upload rate.

Vast amounts of mobile data are collected and extracted from several key network interfaces, in both radio access networks and core networks.

### Additional data

Some very important basic data information also exists in cellular networks, which can be summarized as subscriber profiles and geographic information data. The subscriber profile includes a user's billing information and data plan. The geographic information data contains the location of the cellphone or the point of interest information. Basic data information is relatively static compared with other data; however, it's vital for supporting data analysis.

## STATISTICAL CHARACTERISTICS OF MOBILE BIG DATA

Unlike traditional big data in computer networks, mobile big data has its own typical statistical characteristics. In this article, we focus on three main statistical characteristics for mobile data: spatial-temporal distribution, data aggregation properties, and social correlations.

### Spatial-temporal distribution

Mobile network data can be partitioned to different temporal granularities (for example, 5 minutes, 1 hour, or even 1 day). Similarly, in the spatial domain, mobile data can be studied from a whole city, a typical area, one base station (BS), or even one cellphone. Heterogeneity and fluctuation commonly exist in both spatial and temporal distributions. Mobile traffic in different locations such as stadiums, campuses, and dense residential areas exhibits different patterns in different time scales. Full utilization of such characteristics has already been exploited in the deployment of mobile networks.

### Data aggregation properties

Usually, the characteristics of aggregation properties for mobile data are more important for network performance optimization. We focused on group behavior rather than individual behavior.[2] For a given geographical area and a certain time period, group subscribers will likely request the same traffic, resulting in a similar traffic pattern. The aggregation features will be further used in predictive models to improve network performance.

### Social correlations

Because of the social nature of human beings, mobile users in close proximity tend to exhibit similar habits, behavior, and mobility rules.[3] For example, the social correlations of users leads to a larger scale of data-traffic correlations in both temporal and spatial domains, such as the autocorrelation and cross-correlation properties of mobile traffic. Thus, social correlations are a universal phenomenon in mobile data.

## SOCIAL CHARACTERISTICS OF MOBILE BIG DATA

In current literature and applications, mobile big data has been studied and utilized from various perspectives. We investigate the potential performance gain of analyzing mobile big data from the perspective of social network analysis (SNA). SNA is an interdisciplinary field that has been widely used to study the relationship among individuals, groups, or even entire societies.

### User social networks

A user social network (USN) describes the social relationship among users and has recently received a great deal of attention in many fields. As for mobile data, several classical methods have been proposed to construct a USN by using call detail records. In such cases, social graph construction approaches use phone numbers (users) for the nodes and call connection for the graph edges. In recent years, as more people use mobile devices to access the Internet—including online social networks such as Facebook and Twitter—we see a convergence of social and mobile networks. People tend to highly value the content recommended by friends or people with similar interests; thus, the network edges can be easily defined in several ways, such as the content shared between two subscribers, their common interests, and the habits of each individual. Once established, the USN can be used to detect community structures, understand true communication behaviors, and build a user-centric wireless network.

### Base station social networks

In mobile networks, it's difficult to obtain users' accurate positions. However, the location of a BS or cell tower can provide rough spatial information for wireless traffic, which is sufficient from the perspective of a large metropolitan area. Here, we present the concept of a BS social network (BSSN) and construct one with collected traffic data from Hong Kong's LTE network.

Each node in a BSSN is a BS. Unlike a USN, the edges in a BSSN represent relationships rather than real social ties. To construct a BSSN, the relationship between BSs is quantified with the BSs' traffic traces by using the Pearson correlation coefficient. Then, a planar maximally filtered graph is applied to filter the relationship to obtain a BSSN.

Several interesting studies can be performed based on a BSSN, such as link prediction, community detection, and social influence analysis, but we mainly focus on a BSSN's community structure. "Community" refers to a subgraph structure in which nodes have a higher density of edges, whereas vertices between subgraphs have a lower density. Analytical results show that each community in a BSSN corresponds to a typical scenario with a common traffic pattern, which can be used to recognize typical traffic scenarios for mobile networks.

### App social networks

With the popularity of smartphones and mobile applications, unique mobile apps continue to emerge, such as location-based services, games, and shopping applications. Data traces collected from mobile networks reflect use of a large number of applications. Using a similar method to BSSN construction, an app social network (ASN) is established with the real spatio-temporal cellular network traffic data collected from Shenzhen's 3G network, as shown in Figure 2.

In an ASN, each node represents one typical app, and the edges indicate the strength of the relationships among various apps. In Figure 2, the node (app) with a higher degree has a larger size. This indicates that this app tends to be used with various other applications at a specific time; and apps of the same color belong to the same community, such as video, instant messaging, office applications, and others.

### Interaction between social networks

With the vast amounts of data collected from mobile cellular networks, USNs, BSSNs, and ASNs are established from the SNA perspective: USNs mainly reflect on the social ties between users and model their behavior characteristics, BSSNs study the traffic relationship of each BS, and ASNs represent the usage patterns of diverse apps. With the emergence of new mobile applications, we see the convergence of these three networks. For example, *Pokemon Go* (a game in which people use real-time maps to search for *Pokemon* characters) recently became very popular all over the world. Players in the same vicinity (within one BS) share the same live map and can cooperate with each other. Hence, this social network can be exploited to provide a quality experience by utilizing the app's data.

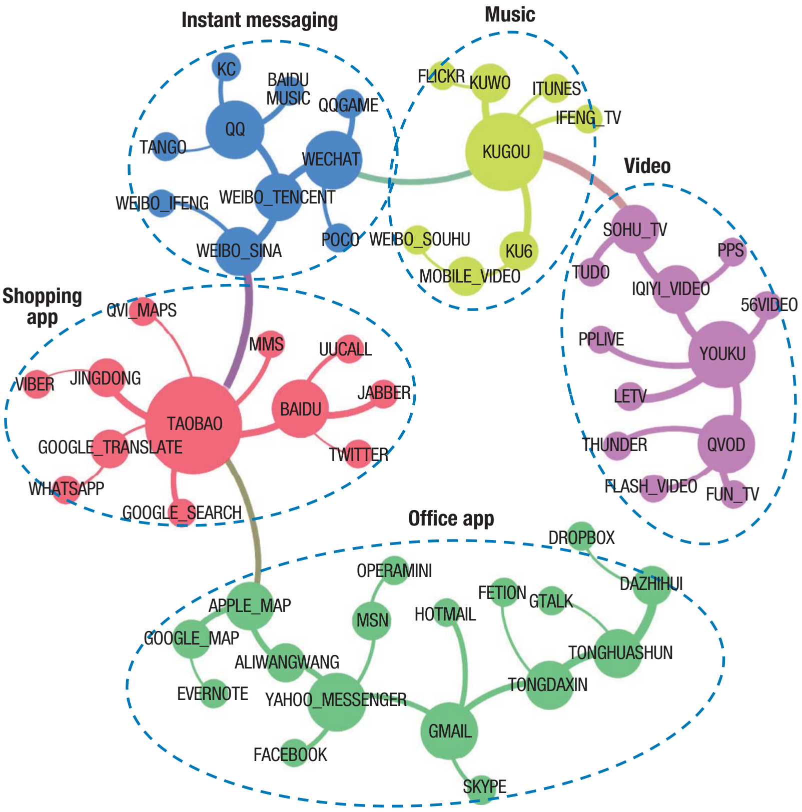


**Figure 2.** App social networks. Each node represents one typical app, and the edges indicate the strength of relationships among various apps. The node (app) with a higher degree has a larger size; apps of the same color belong to the same community, such as video, instant messaging, office applications, and others.

Our results span a wide range of subjects in the field of social computing for mobile big data; however, there are many open questions. First, new social relationships (besides USN, BSSN, and ASN) from mobile big data need to be explored. Second, the dynamics and evolution of mobile big data's social characteristics should be investigated. Third, how do we exploit the social characteristics from mobile big data to design and optimize practical cellular networks?

Along with the evolution of mobile network architecture and other technologies like virtual and augmented reality, social computing will be incorporated in all aspects of cellular networks. Operators and app developers face a great challenge in making full use of the available information. C

## REFERENCES

1. W. Fan et al., “Sensing and Monitoring for Cellular Networks: A Crowdsourcing Platform from Mobile Smartphones,” *Proc. IEEE Int’l Conf. Data Science and Data Intensive Systems* (DSDIS 15), 2015, pp. 472–473.
2. X. Zhang et al., “Energy-Efficient Multimedia Transmissions through Base Station Cooperation over Heterogeneous Cellular Networks Exploiting User Behavior,” *IEEE Wireless Comm.*, vol. 21, no. 4, 2014, pp. 54–61.
3. X. Zhang et al., “Enhancing Spectral-Energy Efficiency for LTE-Advanced Heterogeneous Networks: A User’s Social Pattern Perspective,” *IEEE Wireless Comm.*, vol. 21, no. 2, 2014, pp. 10–17.

**XING ZHANG** is a full professor in the Beijing Advanced Innovation Center for Future Internet Technology at the Beijing University of Technology and the Key Laboratory of Universal Wireless Communication, Ministry of Education, School of Information and Communications Engineering at the Beijing University of Posts and Telecommunications. Contact him at zhangx@ieee.org.

**ZHENGLEI YI** is a master’s student in the School of Information and Communications Engineering at the Beijing University of Posts and Telecommunications. Contact him at zhengleiyi@bupt.edu.cn.

**ZHI YAN** is an assistant professor in the School of Electrical and Information Engineering at Hunan University. Contact him at yanzhi@hnu.edu.cn.

**GEYONG MIN** is chair and director of the High-Performance Computing and Networking Research Group at the University of Exeter. Contact him at g.min@exeter.ac.uk.

**WENBO WANG** is a full professor in the Key Laboratory of Universal Wireless Communication, Ministry of Education, School of Information and Communications Engineering at the Beijing University of Posts and Telecommunications. Contact him at wbwang@bupt.edu.cn.

**AHMED ELMOKASHFI** is a senior research scientist at Simula Research Laboratory. Contact him at ahmed@simula.no.

**SABITA MAHARJAN** is a postdoctoral fellow at Simula Research Laboratory. Contact her at sabita@simula.no.

**YAN ZHANG** is chief scientist at Simula Research Laboratory and a professor in the Department of Informatics at the University of Oslo. Contact him at yanzhang@ieee.org.

COMPUTING EDUCATION

EDITOR ANN E.K. SOBEL
Miami University; sobelae@muohio.edu

# Teaching Cloud Computing

**Scott Campbell,** Miami University

*Just how do you teach cloud computing? It requires substantial hands-on applications and is constantly evolving—thus it presents some unique challenges for educators.*

With the emergence of new computing paradigms, as explored throughout this special issue of *Computer*, it's important that they're incorporated into computing education. Educators in computing are continually challenged by the rate of innovation and change that characterize our field. It's hard to distinguish new technologies that will truly have an important impact from those that, while interesting, don't fundamentally change anything. I'm also frequently amazed at how old technology is reworked into new technology and at how hard it is to incorporate new technological paradigms into current teaching practices.

This holds true for cloud computing, which, when you get right down to it, is really just a new way of doing centralized computing. Yet, it's so profoundly disruptive that it's challenging to properly incorporate teaching about this technology into our curriculum. However, realizing that hardware can now be treated as a software resource—something to be created and managed just as any other object in our programs—is key.

Need to compute a large dataset? Instantiate a set of large, multiple-CPU virtual machines for the computation. Need to handle thousands of mobile requests? Configure a container service and push out a new handler. Treating hardware as a software resource can also impact system updates. We can easily create a new instance of the system hardware, install the new code, run the system tests, and then change a few system parameters to bring the new system online. Treating hardware as a configurable software object is a paradigm that requires us to not only teach computer science as the decomposition of problems into functions, but to also treat the decomposition of problems into resources. Thus, our students need a solid understanding of foundational operations.

## CLASS CLOUDS

Developing systems that take full advantage of cloud computing has been a focus for several years now; thus, programmers are increasingly comfortable with playing the dual roles of programmer and operator. However, this creates a new challenge for educators because operations hasn't traditionally been part of the computer science curriculum. Teaching cloud computing requires the educator to venture into operations, OSs, networking, and other applied areas.

It's easy for students to get bogged down with difficulties in setting up firewalls and network interfaces. Now they'll also need to create a test instance consisting of both software and systems. For this, products such as Vagrant can be used to make this faster while reducing apparent complexity. It's also possible for students to be given direct access to virtual system consoles and asked to configure their own systems. Instructors must deal with these additional dimensions and added complexity while teaching and creating meaningful assignments.

One way of managing this change is to consider teaching DevOps. As a response to new development cycles for Web-based applications, DevOps involves close interactions between operations and programming activities. In blending both the paradigm of treating hardware as a software resource and the use of cloud computing tools, DevOps feeds into this approach of teaching operations as a software tool. It's no longer sufficient to simply build applications; the applications themselves must control the infrastructure. Instructors need to consider teaching a range of solutions from use of simple virtual servers to more complete services—for example, microservices such as Amazon's Lambda or container services such as Docker integrate applications inside standardized virtual computing systems. These services offer rich APIs, allowing software to have total control of the infrastructure. For example, Openstack (as well as all other cloud service providers) offers Representational State Transfer (REST) APIs, allowing software to monitor overall workload as well as to start and stop hardware as required to provide optimal system operation. This adds another dimension to the programmer's realm and is a task that overlaps traditional operations.

## CLASS CLOUD COSTS

Cost is another area of concern for educators. Although cost savings is a key cloud computing tenet, there's still a cost. Typically, students develop programs on their laptops or on shared lab resources provided by their institution. These resources—which are cheaper to install and operate—frequently aren't up to datacenter production standards, as they're sized to handle student projects, not 24-7 production systems. In addition, instructors are rarely responsible for finding funding for such resources. However with larger, multiuser cloud computing systems, instructors often must find funding for them if they want students to fully use them. Vendors have responded with grants and free introductory services for educational institutions. For example, Amazon recently introduced its Amazon Web Services (AWS) Educate Program in which students are given $100 of cloud computing resources per year. These grants aren't typically enough to leave servers running 24-7, but they're sufficient for specific assignments. Although this addresses some factors relating to cost, the instructor still must handle additional management and administration tasks.

In addition, overhead can be further reduced through open source solutions available for educators that want to run their own systems. I've regularly used Openstack running on older servers to successfully introduce students to cloud computing basics. Through Openstack it's possible to have a cost-effective development and test platform that can run 24-7 for student projects. Students can manage their own infrastructure and have access to a rich set of APIs to programmatically create and manage resources. These systems can then be left running long term at little additional cost. Although these tools don't have the rich range of options that other cloud providers offer, they're an inexpensive option for teaching cloud management.

Determining which set of resources to integrate and the right level of detail is a major challenge, especially when designing educational assignments and labs. For years instructors have used techniques to get students involved in soliciting system requirements, which leads to the system design. These techniques should be extended so students can make infrastructure choices as well as programming choices.

This adds a totally new dimension. Using an analogy, in the past we'd ask the student to write a control system for a pickup truck. With these new approaches, we instead ask them to select which vehicle to use to solve the problem. They might pick a helicopter if they need access to remote areas or a freighter if the problem involves moving huge loads. Alternatively, students will find that large processing-based systems will benefit from multiple instances that can be started and stopped when needed, whereas mobile applications could benefit from containerized solutions. As educators, it's up to us to help the students understand the range of options, along with the relative strengths and weaknesses of each.

This added complexity requires educators to perform quite a balancing act while teaching. Students need to debug hardware and configuration errors in addition to logic and data errors. System configuration mistakes can be very difficult to debug. For example, it's very difficult to track down errors when they cause the service to fail during creation. Students now need to learn new operations-related vocabulary, as well as integrate multiple log files to find errors. It's also more difficult to create beginning frameworks or prototypes to give to students.

> Determining which set of resources to integrate and the right level of detail is a major challenge, especially when designing educational assignments and labs.

However, having access to hardware that's managed via software makes teaching DevOps and cloud computing much more practical, although managing enough physical servers and the associated infrastructure for each student to have his or her own individual server is very difficult—thus, most educators don't even attempt it. Tackling the complexity is worthwhile when it gives students the skills to be successful using these new technologies.

New computing paradigms should be welcomed and embraced by users, developers, and educators. Increased flexibility, update speed, cost-effectiveness, efficiency, and so on are some of cloud computing's key benefits. Educators should introduce these new technologies to equip students with the skills to effectively use and expand them.

I believe students with these skills will be highly valued by companies and researchers alike. This greater potential for student success makes the additional work for the instructors worthwhile and beneficial.

**SCOTT CAMPBELL** is the director of technology for the College of Engineering and Computing at Miami University. Contact him at campbest@miamioh.edu.

STUDENT DESIGN SHOWCASE

# Seeing Is Understanding

**Greg Byrd,** North Carolina State University

*To help teach object-oriented programming, students at King Abdulaziz University in Saudi Arabia created a self-paced, interactive program that associates code with visual cues to reinforce the concepts of inheritance and polymorphism.*

Object-oriented languages, like Java, C++, and Python, incorporate features that simplify the construction and maintenance of large, complex software systems. The use of classes, objects, and interfaces allow for code reuse and extensibility. However, students sometimes struggle initially with object-oriented concepts, such as the difference between a class and an instance of that class, or the intricacies of inheritance in a class hierarchy.

A group of senior computer science students at King Abdulaziz University (KAU) in Jeddah, Saudi Arabia, designed OOPVisual, an interactive program that helps students learn object-oriented programming (OOP). OOPVisual uses 3D animated objects to illustrate coding concepts and offers a drag-and-drop interface to avoid typing errors. The visual cues show students how small differences in code can lead to major changes in behavior. More important, they help learners deepen their understanding of the code and of how OOP concepts work in larger systems.

Drawing on their own experiences as beginning programmers, the team focused on *polymorphism*—the ability of an entity or substance to appear in multiple forms. For example, water can be found in liquid, solid, and vapor states. In software, polymorphism refers to the programmer's ability to create a uniform interface to manipulate data of different types. This simplifies the code and makes it easier to extend when new types of data are introduced.

In Java, one form of polymorphism comes from *subtyping*, the ability to derive a subclass from a parent class. A subtype object (instance) can be referenced by a variable of either the subtype or the parent type. Consider, for example, an Employee class that is a subclass of Person. An object that represents an employee can be referred to as either an Employee or a Person. In other words, every object that is an instance of the Employee class is also an instance of the Person class. Thus, we can define functions (methods) that operate on a Person, and we can use that same function on Employee objects.

This is a powerful feature, but one that can confuse novice programmers. A survey of KAU's second semester programming class revealed that 47 percent of its 186 students found understanding polymorphism difficult and that 59 percent had trouble implementing polymorphism in Java.

0018-9162/16/$33.00 © 2016 IEEE

EDITOR GREG BYRD
North Carolina State University; gbyrd@computer.org

## INCORPORATING VISUALIZATION WITH CODE

Having grown up with 3D graphics, animation, and interactive videogames, today's students might view text-based explanations and examples as outdated. With this is mind, the team set out to incorporate familiar graphical user interface (GUI) elements into OOPVisual.

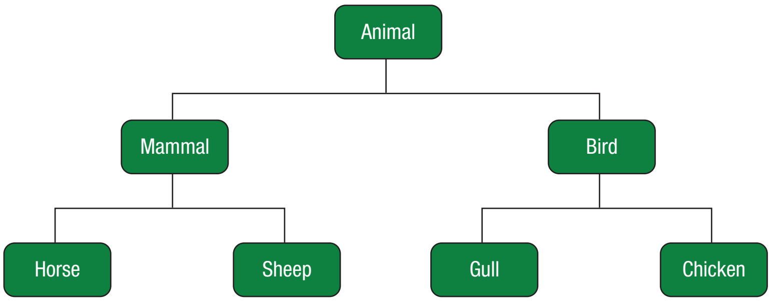
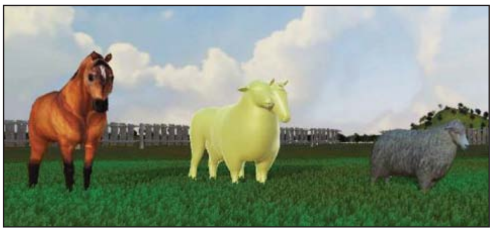


**Figure 1.** OOPVisual class hierarchy. The overall setting for the user experience is a virtual farm populated by different types of animals. This familiar context serves as the analogy for software data types.



```
Horse h =
  new Horse();

Mammal m =
  new Horse();

Sheep s =
  new Sheep();
```

**Figure 2.** Visual representations of Horse, Mammal, and Sheep objects, with corresponding Java code.

## PROJECT DETAILS

- Title: OOPVisual
- School: King Abdulaziz University, Jeddah, Saudi Arabia
- Students: Wejdan Eissa Moussa, Raniyah Mutlaq Almalki, Maryam Abdulrahman Alamoudi
- Faculty mentor: Arwa Allinjawi

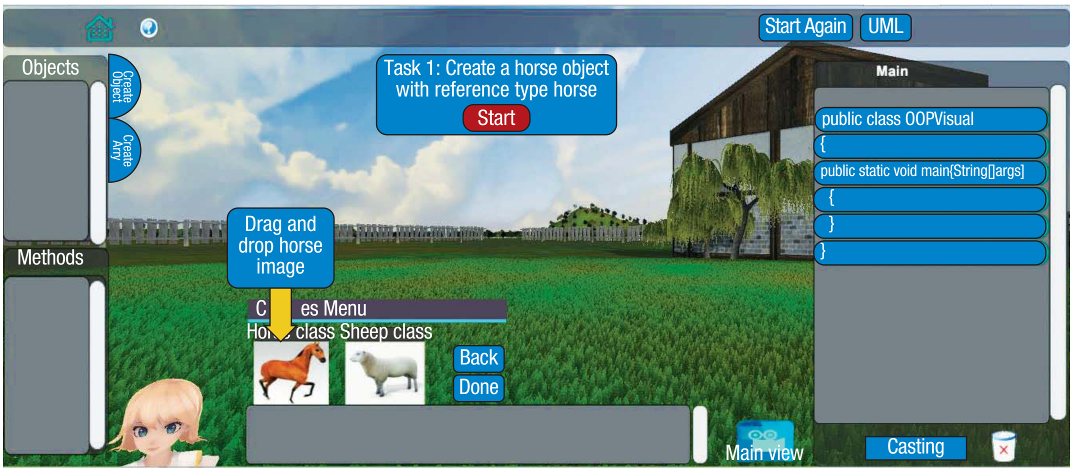
The overall setting for the user experience is a virtual farm populated by different types of animals. This familiar context serves as the analogy for software data types. The students begin with the knowledge that there are different animal types and that although all animals have things in common (they all move, breathe, and eat, for example), there are also differences among types. For instance, some have hair (mammals) while others have feathers (birds).

This familiar concept can be expressed as a class hierarchy, as shown in Figure 1. Animal is the base class, and Mammal and Bird are subclasses of Animal. Further specialization gives us Horse and Sheep as subclasses of Mammal, and Gull and Chicken as subclasses of Bird.

Each type has its own visual representation. These are animated images that move and make sounds, not just static pictures. When students instantiate an object through a series of mouse clicks and a drag-and-drop operation, the corresponding animated figure appears in the farm scene.

As shown in Figure 2, the visual representation of the most specific classes, such as Horse and Sheep, are what you would expect. But how do you represent a Mammal object? For this abstract type, the team created a shape that's a strange but recognizable combination of horse and sheep. This helps students understand that a Mammal can refer to either a Horse or a Sheep, and only operations that are defined for the parent class (Mammal) can be used to manipulate the object. Similar shapes are defined for Bird, combining characteristics of Gull and Chicken; and Animal, which contains recognizable elements of all four animal types.

Note that Java doesn't allow instantiation of abstract types. In other words, programmers can't create an instance of a Mammal object. However, they can create a Horse object and then refer to that object using a Mammal variable, as shown in the Java code in Figure 2. This concept is important for students: the mammal shape on the screen is either a horse or a sheep, but they can't tell which; they can only treat it as a mammal.



```
public class OOPVisual
{
public static void main(String[]args)
{
}
}
```

**Figure 3.** OOPVisual user interface. Objects and their methods are shown on the left, with the corresponding Java code on the right. As students progress through the exercise, pop-up help and instructions appear.

**Figure 4.** The OOPVisual team at the Microsoft PanArab Imagine Cup semifinal in Cairo: Raniya (left) and Wejdan (right). Not pictured: Maryam.

## USER INTERFACE

Figure 3 shows the basic user interface. The farm scene, which students will populate with animal objects, is shown in the middle. Users can zoom in and out, and scroll left and right—they're not limited to the space shown.

On the left are controls related to objects. The top box lists the objects currently in the scene. Attached to the top box are buttons for creating an object, or an array of objects. To create an object, students click through the class hierarchy to find a class that can be instantiated, for example, Animal, then Mammal, and then Horse. They then drag the horse icon to the scene and drop it, at which point a pop-up forces them to name the variable and select its type (Animal, Mammal, or Horse) from a drop-down menu.

Once the object is created in the desired way, the Java code that produces the object appears in the right-hand box. It's critical to connect the visual content with code. After all, the program's goal is to teach students how to write code.

When students click on a visual object, the bottom box on the left displays the operations (methods) that can be applied to that object. For example, the *walk* method causes the animal to walk forward for a specified amount of time. Other methods include *eat*, *jump*, and *makeSound*.

If the object is referenced by an abstract class (for instance, the Mammal object in Figure 2), the methods of the underlying object (for example, Horse) are shown, but only the methods allowed for Mammals can be used. The Horse-specific methods are grayed out and can't be selected. This reinforces the notion that the programmer can only "see" a Mammal and doesn't know which concrete class was actually instantiated.

The tool also illustrates the concept of dynamic binding of a type to the object. When a method is invoked on an abstract object, the underlying concrete object appears briefly to perform the method task. After the method has been executed, the image changes back to the abstract representation.

Students can also create an array of objects. First, they create an array and choose a specific class for elements of the array. They can then instantiate objects for each element. If the array class is abstract, they can use different subtypes to populate the array. For example, an Animal array can contain both a Sheep and a Chicken. But both will be seen as Animals when referenced through the array. This powerful feature of polymorphism is important for students to fully understand.

## PROGRAM MODES

In addition to a collection of help videos, the program allows students to interact in four ways:

- *Concept tutorials* walk students through the steps required to accomplish tasks such as creating a Horse object with a Horse reference, a Horse object with a Mammal reference, or a Mammal array that contains two Horses and three Sheep. The program displays what to do at each step, allowing students to perform only the correct steps. At the end of each tutorial, students can choose to do related exercises for more practice or to move to the next tutorial.
- *Exercises* are also guided activities with specified tasks; however, students must now choose what to do. The program shows a green check for a correct choice and a red X for a wrong one. Hints are available at every step.
- *Quizzes* present the students with tasks but without step-by-step instructions. Final results are checked, but the students aren't led through the choices.
- *Create your own scene* allows the students to interact freely with the virtual farm environment. They can experiment within the bounds of what can be expressed through the interface.

## IMPLEMENTATION AND RESULTS

The OOPVisual team used the waterfall design methodology for requirements gathering, specification and analysis, design, implementation, testing, and maintenance. They chose this methodology because it allowed them to identify all of the requirements up front and to discover design errors and challenges early.

The implementation was completed using the Unity 5 game engine and the C# programming language. Unity provides built-in graphics, such as shaders and effects, as well as physics and collision engines. C# was used to program the animals' behavior and the camera's movements. Microsoft Visual Studio was the implementation environment, with specific tools to support Unity development and testing.

To test OOPVisual's effectiveness, the team gave multiple-choice tests, before and after using the tool, to 63 KAU Faculty of Computing and IT students. The students were at various academic levels, but none had taken the OOP class that year. After using OOPVisual, the number of correct answers increased by 67 percent.

In addition to presenting conference papers in Saudi Arabia and Japan, the OOPVisual team entered their project in Microsoft's 2016 Imagine Cup competition, a global contest that encourages students from various disciplines to create applications, games, and solutions. OOPVisual was the first-place national winner in the Innovation category, among more than 100 entries from Saudi Arabia. The team traveled to Cairo to participate in the PanArab semifinal event.

Future plans for OOPVisual include the addition of more classes, such as fruits, vegetables, and people. The team also intends to enhance the coding environment to orchestrate interactions among the objects, and to provide a "play" button for the scene. Other goals include improving error recovery so that students can undo an action when needed. The team plans to work with universities and textbook publishers to visualize other languages and computer science concepts. C

**GREG BYRD** is associate head of the Department of Electrical and Computer Engineering at North Carolina State University. Contact him at gbyrd@computer.org.

CYBERTRUST

# Entropy as a Service: Unlocking Cryptography's Full Potential

**Apostol Vassilev and Robert Staples,** National Institute of Standards and Technology

*Securing the Internet requires strong cryptography, which depends on good entropy for generating unpredictable keys. Entropy as a service provides entropy from a decentralized root of trust, scaling across diverse geopolitical locales and remaining trustworthy unless much of the collective is compromised.*

Cryptography is fundamentally important for information security, whether the information is data in transit over the Internet or at rest on storage devices. Today, the security of data protected by cryptography depends not on secret algorithms, but primarily on having strong keys and keeping them secret.

Generating strong cryptographic keys is no simple matter, however. Experts recommend using deterministic random bit generators (DRBGs),[1] but the sequence of numbers generated by a DRBG can be traced predictably to the seed (initial value) supplied to the generator. Knowing the seed, one can reconstruct the number sequence produced by a particular DRBG. Thus, DRBGs must be seeded with hard-to-guess random data from a reliable source.

In information theory, such so-called "high-entropy" sources provide true randomness. They're usually based on nondeterministic physical processes such as ring oscillators or some kind of quantum behavior. Most practical computer systems rely on events like mouse movements, keyboard stroke timings, network events, and hard-disk access times to generate hard-to-guess random data for seeding DRBGs. Although sometimes plausible, such sources often provide a limited amount of unpredictability—that is, low entropy—because, as in the case of headless or other embedded devices, they lack these sources of unpredictability.[2]

This problem is exacerbated in cloud computing environments, which often lack the sources of nondeterminism harnessed by traditional computers for harvesting entropy.

0018-9162/16/$33.00 © 2016 IEEE

EDITOR JEFFREY VOAS
National Institute of Standards and Technology; j.voas@ieee.org

Cloud service providers typically use a single reference image of a guest virtual machine (VM)—a "golden" image—in order to create multiple instances of it in response to user demand. Each instance often has very limited ability to harvest randomness.

Another domain with a strong demand for good cryptographic keys is the Internet of Things. IoT devices tend to be small and resource constrained, but their functional capabilities span a wide range. It's reasonable to conclude that some types of IoT devices with network connectivity and modest computational power to perform asymmetric cryptographic operations would benefit from an entropy service architecture.

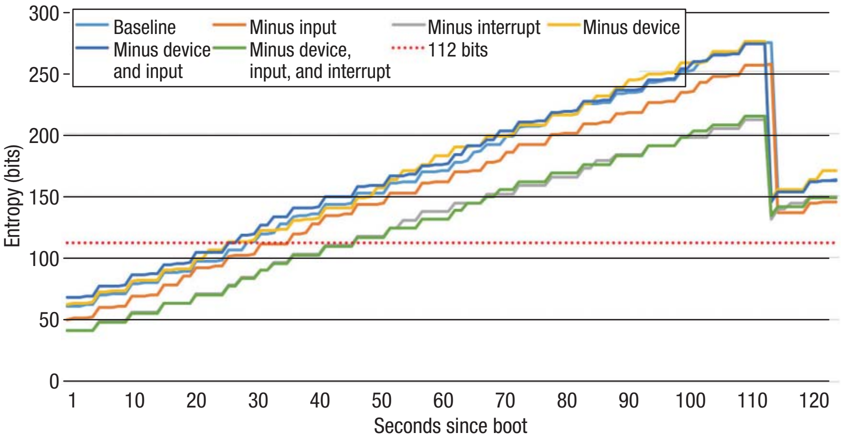


**Figure 1.** Entropy pool size of a Linux kernel in a simulated embedded device with different combinations of entropy input devices disabled (/proc/sys/kernel/random/entropy_avail). Depending on the combination of sources disabled, the entropy count in the pool took 20–45 seconds to generate the bare-minimum threshold of 112 bits.

## RECENT EXPLOITS OF POOR ENTROPY

Concerns about cryptographic keys' potential weakness aren't just theoretical, as evidenced by some recent real-life examples of catastrophic security breaches resulting from poorly constructed or predictable cryptographic keys.

### Boot-time "entropy hole"

A study by researchers at the University of California, San Diego (UCSD) and the University of Michigan (UM) provided one of the most comprehensive Internet-wide searches of Transport Layer Security (TLS) and Secure Shell (SSH) servers to date, checking 12.8 million TLS and 23 million SSH hosts.[2] The study results were alarming: 5 percent of TLS and 10 percent of SSH hosts shared keys because of insufficient entropy from the source used, allowing the researchers to actually calculate the private keys of 0.5 percent of TLS hosts and 1 percent of SSH hosts.

The study showed that almost all vulnerable hosts were embedded network devices like routers or firewalls. Such hosts often run a pared-down Linux kernel and don't have the usual random events from input devices that a desktop computer would have. This creates an "entropy hole," in which the output of /dev/urandom could be constant across multiple boots for a period of time early in the boot process. In one case, the same key was generated in over 25 percent of boots.

### Entropy starvation in embedded devices

We built on the approaches used in the UCSD/UM study to investigate the strength of Linux kernel entropy sources. In particular, we simulated the behavior of an embedded device, without a hard drive or a keyboard/mouse, always starting with an empty entropy pool (no seed). We built a pared-down Linux kernel with different combinations of kernel entropy input devices disabled. Depending on the combination of sources disabled, the entropy count in the pool took anywhere from 20 to 45 seconds to generate the bare-minimum threshold of 112 bits, indicated by the red dotted line in Figure 1. More time was needed to reach the threshold when some contributing entropy sources were turned off, simulating environments with constrained resources.

Our experiment illustrates the potential weakness of Linux kernel entropy sources in embedded Internet deployments—for example, in cloud environments. We observed particularly strong demand for entropy through the nonblocking /dev/urandom interface, with requests as high as 4,096 bits shortly after boot when little random data is accumulated. In fact, this Linux behavior opened the door for the exploits described by the UCSD/UM researchers.

## ENTROPY AS A SERVICE

The security issues resulting from the effects of poor entropy illustrate the fundamental importance of good randomness for security.[3] The existing technological headwinds that hinder the implementation of robust random bit–generation capabilities

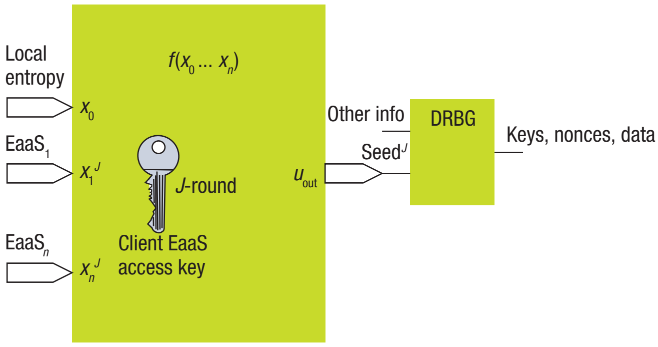


**Figure 2.** Entropy as a service (EaaS) client key management. The initial (0-round) key for accessing EaaS is provisioned out of band (factory, enterprise, and so on). The *J*-round key is obtained from EaaS and determined by KDF (DRBG(Seed$^{J-1}$)), where Seed$^{J-1}$ = $u_{out}$ = $f(x_1, x_2^{J-1}, \ldots x_n^{J-1})$, $1 \le i \le n$, and *f* is a one-way function. KDF denotes key derivation function—an appropriate procedure for asymmetric key generation (E. Barker and A. Roginsky, *Recommendation for Cryptographic Key Generation*, NIST Special Publication 800-133, Nat'l Inst. of Standards and Technology, Dec. 2012; http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133.pdf). DRBG: deterministic random bit generator.

in conventional computing devices make apparent the need for alternative means of providing high-quality entropy to devices that cannot produce their own in sufficient quantity or quality.

The widely available and highly redundant nature of the Internet creates an effective medium by which to provide good random data to clients, in this case using a Representational State Transfer (REST; www.restapitutorial.com) interface, in the form of *entropy as a service* (EaaS). Instead of relying solely on weak pseudorandom data in classical computers, EaaS provides a novel and secure way of delivering high-entropy data to requesting devices. It also leverages existing protocols and technologies, which makes adoption easy. EaaS scales across diverse geopolitical locales, remaining trustworthy unless much of the collective is compromised.

### Secure entropy transfer

EaaS uses HTTP to transfer entropy payloads from the service to clients. To secure this transmission, the server encrypts the data using the client's provided public key and digitally signs the payload with the server's own private key.

The client makes an HTTP GET request to the EaaS server with the number of bytes of random data to return and its own public key, which is used to encrypt the returned payload. The structure of the server's XML response is as follows:

```
<response>
  <entropy>
     encrypted base64-encoded
     random data
  </entropy>
  <timestamp></timestamp>
  <dsig></dsig>
</response>
```

Here, `<entropy>` connotes the payload, encoded in base64 format, and `<dsig>` is the digital signature.

### Resolving the chicken-and-egg conundrum

Clients need a public key to initially access EaaS and request high-entropy data to strengthen their key-generation capabilities. How can a client get a strong public key? It's much easier and cheaper to generate strong keys out of band than to implement robust random bit–generation capabilities in conventional devices. Manufacturers can, and often do, generate strong keys in the factory and provision them on devices. This key-provisioning model is widely used in many industries, including smartcard and Trusted Platform Module (TPM) manufacturing. Upon receiving a shipment of devices to deploy, a customer also receives through independent means the secrets required to change the factory keys on each device and assume ownership of the devices.

To break this well-known provisioning model and take control of a deployed device, an attacker must penetrate the factory security and record every device key issued as well as monitor every interaction between a device and EaaS. Missing just one such interaction would defeat the attack. This sets a high security bar for attackers.

Another mitigation against potential attacks is to always mix the externally obtained random data with locally generated pseudorandom data using suitable cryptographic mechanisms, such as hashing, and to renew the EaaS access key on each round, as Figure 2 shows. Note that the mechanism for updating the client key can provide perfect forward secrecy.

## EaaS ARCHITECTURE

The EaaS architecture, shown in Figure 3, consists of two main parts: the server side and the client side. The critical components are the entropy source, the EaaS server, and a secure device in client systems capable of providing strong isolation and protection of cryptographic keys stored inside the device and offering a set of basic cryptographic services.

The EaaS server is continuously fed random data from the attached quantum source. The data enters a FIFO (first in, first out)-like buffer in the server's RAM, and, when a client request arrives, the server reads the top value from the buffer, signs and encrypts it, and then sends it to the requester. The FIFO buffer shifts after every request and when new data comes from the random source. The EaaS server ensures that the FIFO buffer is erased prior to server shutdown and never paged to disk. Open implementations can help ensure that this actually occurs.

The client system consists of a classic computing device enabled with a dedicated hardware component capable of storing secret cryptographic keys and seeds. A dedicated software application bridges the communication between EaaS and the hardware component. Examples of secure hardware components are the TPM, TrustZone technology in ARM processors, and Identity Protection Technology in Intel processors. If a client system or device doesn't have a secure hardware component, it can still use EaaS. The presence of a hardware component simply provides further guarantees to the system or device user, when present.

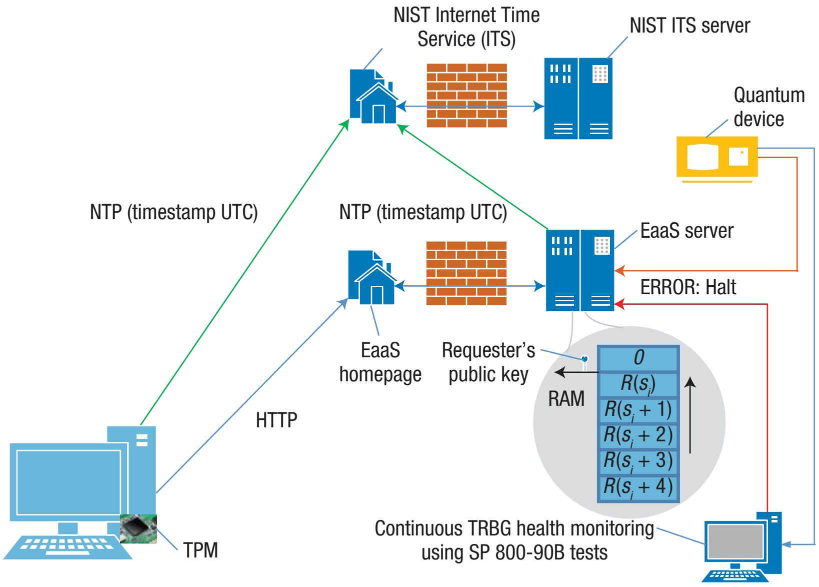


**Figure 3.** EaaS architecture. EaaS transfers entropy to clients in a secure fashion through the use of timestamping and digital signatures. NTP: Network Time Protocol; TPM: Trusted Platform Module; TRBG: true random bit generation; SP 800-90B: M. Turan et al., *Recommendation for the Entropy Sources Used for Random Bit Generation*, NIST Special Publication 800-90B, 2nd draft, Nat'l Inst. of Standards and Technology, Jan. 2016; http://csrc.nist.gov/publications/drafts/800-90/sp800-90b_second_draft.pdf.

## BUILT-IN ATTACK MITIGATIONS

An important feature of EaaS is that it transfers entropy to clients in a secure fashion.

As Figure 3 shows, the server response's timestamp and digital signature allow the client to verify the authenticity of both. Timestamping in particular prevents response-replay attacks. The digital signature protects against both man-in-the-middle attacks—when a malicious actor intercepts messages and serves as a relay—and Domain Name System (DNS) poisoning attacks, in which a malicious actor either intercepts DNS requests or sets up a spoof server near the victim, provided the EaaS public key is provisioned on the client in advance.

Attacks involving dishonest or curious EaaS server instances are mitigated by mixing data from several sources together before use. Thus, even if multiple EaaS instances were somehow colluding against a specific client, if the client can access just one source of noncolluding entropy, including its own weak entropy pool, the efforts of the malicious instances are mitigated, as they have no way of knowing the input from the other, good sources.

## REAL-WORLD USES

One useful application of EaaS is in assessing the security strength of an enterprise system. The strength of cryptographic keys being generated at the system endpoints is of great importance for this task. Endpoints using EaaS can attest the strength of keys generated from data coming from a known good source. Additionally, enterprises could set up their own internal EaaS, and have complete trust over their endpoints' entropy.

Another real-world application is VM orchestration in cloud computing environments. Two VM instances instantiated from a common ("golden") image can demonstrate similar or even identical internal states of the local entropy pool, so gaining insight into one would yield insight into the other. However, this is easily remedied by using EaaS to feed unique random data into the image after cloning, or by requesting some EaaS data upon boot.

Another important use case is that of embedded Internet devices that might be entropy starved.[2] One way to fix this is to use EaaS to obtain entropy on devices upon boot. The devices could also store some entropy across boot cycles. Thus, a device would only be vulnerable for a few seconds after the initial boot, until the EaaS call is made, and simple design decisions could prevent key generation in this small window of time.

Figure 4 shows how seeding Linux with EaaS upon boot greatly improves the behavior of the Linux Kernel Process Scheduler (LKPS), a critical component of the operating system. To implement fair and efficient process scheduling, the LKPS acquires random data through a blocking interface, and

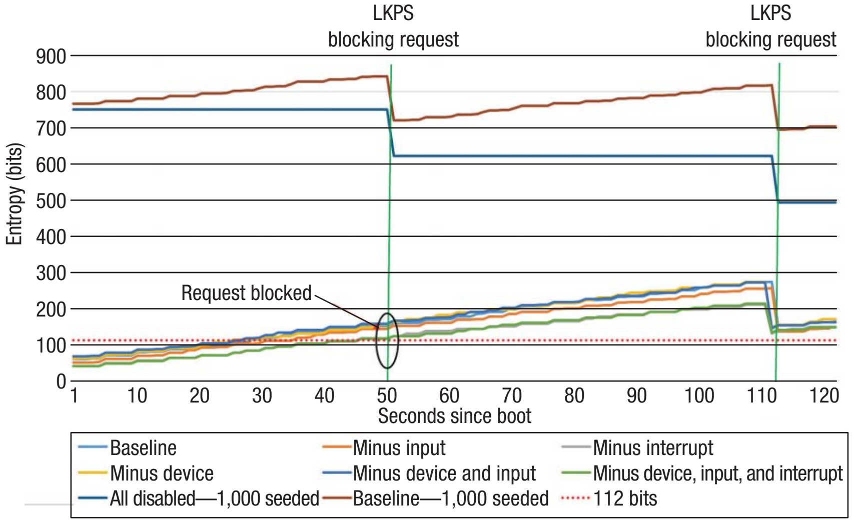


**Figure 4.** Seeding Linux with EaaS upon boot greatly improves the behavior of the Linux Kernel Process Scheduler (LKPS). The black oval indicates missed LKPS seeding due to a lack of sufficient entropy in the kernel.

it can only do so when sufficient entropy has accumulated in the kernel. When Linux is seeded with EaaS, the LKPS acquires its first random seed after about half the time needed in the case of nonseeding. The black oval in the lower part of the figure indicates missed LKPS seeding due to a lack of sufficient entropy in the kernel. In other words, the LKPS reaches its normal operational regime much faster when Linux is seeded from the start, thereby improving the OS's overall stability and performance without any additional design or configuration changes.

The proverbial Achilles' heel of cryptographic security protection is the lack of strength of the keys used to protect critical data. EaaS stands to serve as the basis of a future ecosystem of servers that can provide verifiably high-quality entropy to clients on request, thereby unlocking cryptography's full potential.

To facilitate the creation of such an ecosystem, we plan to share our server implementation, allowing other organizations or entities to review, adopt, and host their own EaaS instances. We also envision the need to develop criteria for establishing trustworthiness of servers participating in the ecosystem. This, in turn, would let EaaS users select and rely on a subset of servers from the ecosystem that satisfies a desired level of trust/risk.

We welcome input and comments regarding EaaS. C

## REFERENCES

1. E. Barker and J. Kelsey, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST Special Publication 800-90A, rev. 1, Nat'l Inst. of Standards and Technology, June 2015; http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf.
2. N. Heninger et al., "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices," *Proc. 21st USENIX Security Symp.* (Security 12), 2012; www.usenix.org/system/files/conference/usenixsecurity12/sec12-final228.pdf.
3. A. Vassilev and T. Hall, "The Importance of Entropy to Information Security," *Computer*, vol. 47, no. 2, 2014, pp. 79–81.

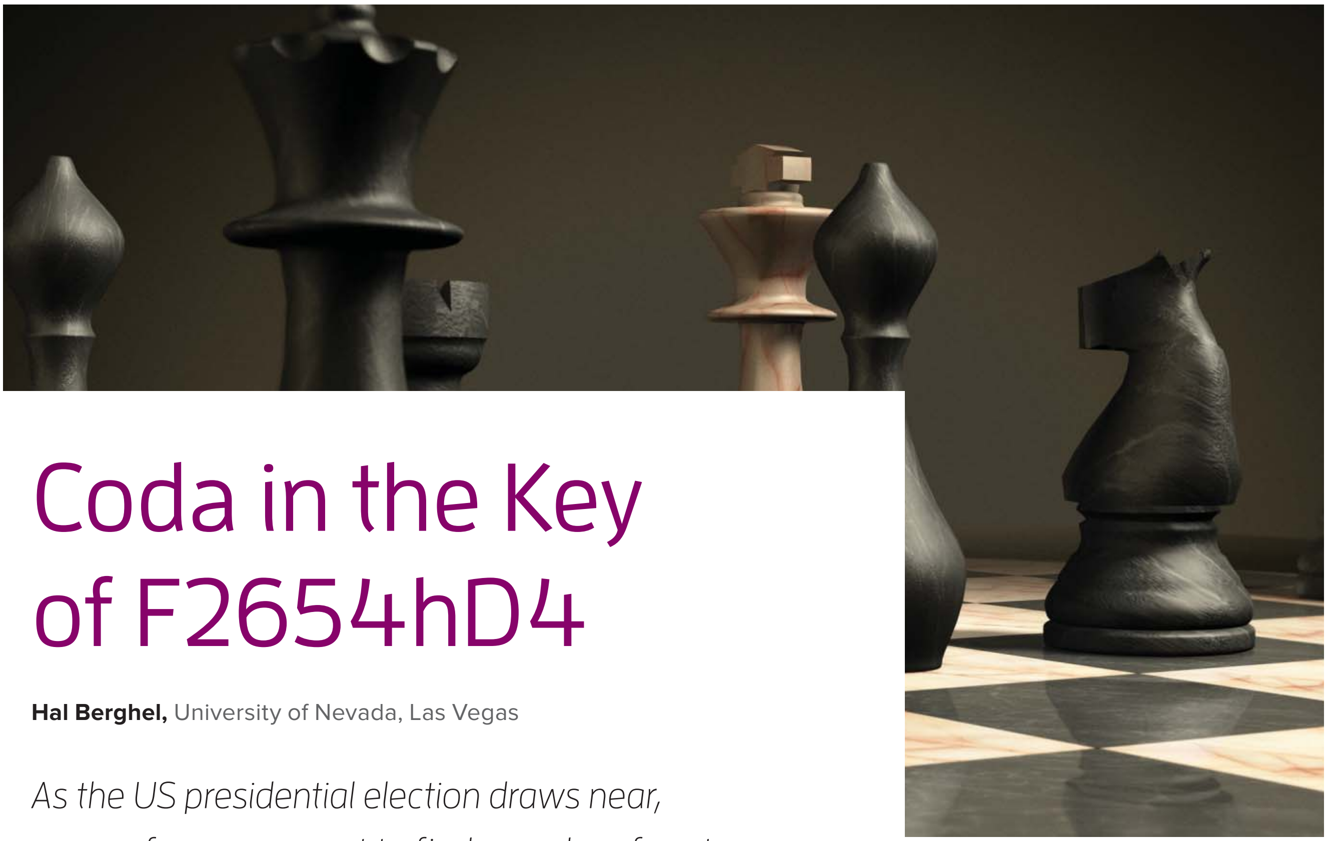
**APOSTOL VASSILEV** is research team lead in the Security Testing, Validation, and Measurement Group, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. He is also chairman of the Industry Working Group dedicated to modernizing the Cryptographic Module Validation Program through the adoption of advanced machine-based testing methodologies and automation. Contact him at apostol.vassilev@nist.gov.

**ROBERT STAPLES** is a student researcher in the Security Testing, Validation, and Measurement Group, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, pursuing an MS degree in cybersecurity at Johns Hopkins University. Contact him at robert.staples@nist.gov.

### DISCLAIMER

The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

OUT OF BAND



# Coda in the Key of F2654hD4

**Hal Berghel,** University of Nevada, Las Vegas

*As the US presidential election draws near, many of us can expect to find ourselves face to face with an electronic voting machine. It's time to re-examine the integrity of these machines.*

Remember that default DES key that was hard coded in the Diebold voting machines for many years? Despite revelations that shocked the voting public, computer scientists concerned with the systems security of Direct-Recording Electronic (DRE) voting machines were central to uncovering the now familiar Diebold debacle of 2005—when activist Bev Harris discovered and then posted the unprotected source code for the Diebold TS and TSx voting machines. She recovered the code from the Diebold website.[1] What unfolded was a fascinating chronicle of corporate irresponsibility, hubris, incompetence, political chicanery, and power politics—all wrapped up in a story befitting a good dime store novel. And the proverbial plot thickened when computer scientists got involved—at that point, things got downright ugly. As painful as it was for the computer scientists involved, the country is far better off for it.

## STANDARDS VACUUMS

The professional computing community is very familiar with the role of standards. Well-known standards, such as ISO 17799 for IT security and the ISO 9000 series for management, establish guidelines and general principles that codify industry best practices. In some cases independent certification bodies are used to assure customers and the public of compliance. Although we all work with standards differently, we can agree on two things: first, standards provide a minimal assurance of integrity and quality, and second, wandering too far afield usually comes at a cost in terms of safety, reliability, performance, profitability, and credibility for the affected organization and its representatives.

There are standards for quality, safety, reliability, and so on, in industries related to food, drugs, military equipment, manufacturing, computer equipment, software, household appliances, floor coverings, and paint, just to name a few. However, one area that's historically been immune to reasonable standards is the manufacturing and use of voting equipment—that which determines our political future.

0018-9162/16/$33.00 © 2016 IEEE

EDITOR HAL BERGHEL
University of Nevada, Las Vegas; hlb@computer.org

The voting franchise has been operating in a standards vacuum for decades, and this vacuum extends well beyond voting equipment.[2] The Diebold story is just the tip of the iceberg.

We know that using the same encryption key for every transaction hasn't been an acceptable practice since the Caesar Cipher was popular in ancient Rome. Furthermore, since the mid-1970s, the DES algorithm was known to be vulnerable to brute-force attacks because of its short key length.[3,4] After DES was deprecated by the National Institute of Standards and Technology (NIST) and replaced by the Advanced Encryption Standard (AES), Diebold went on to hard code it into the source code—a willful circumvention of best practices for the sake of cost savings and expedience. That they then lied about it was an even greater betrayal of the public trust. All of this was possible because of the lack of both industry standards and accountability to the public. Further, Diebold carried on this way for an entire generation of voting machines.

Diebold's story is a shining example of the voting machine industry's heritage of stupidity and arrogance and the public's tolerance of proprietary electronics and software that have never been adequately tested by impartial, legitimate domain experts. Harris's disclosure broke the proprietary veil, and some daring computer scientists read the curious public in on some outrageous security breaches that we never would have known about.

Exposure of the Diebold AccuVote system's weakness is generally credited to Johns Hopkins University computer scientist Aviel Rubin and his colleagues, who in 2003 began analyzing the source code discovered by Harris.[1] It's useful to frame this story in terms of Rubin's analysis of the Diebold source code. Here's what he found:

1. The AccuVote system anonymized the voting order with a linear congruential generator (LCG) that didn't work properly and was inappropriate for this purpose, thereby undercutting the principle of the secret ballot.
2. In parts of the code that required cryptography, either the algorithms were incorrectly applied or not used at all. (Of course, as mentioned, DES was used despite having been deprecated by NIST.)
3. Diebold's approach to key management was juvenile. The same encryption key (see column title) was hardcoded into every voting machine. The vulnerability stemming from a lack of key management was first reported in 1997 by University of Iowa computer scientist Douglas W. Jones without effect (http://homepage.cs.uiowa.edu/~jones/voting/dieboldacm.html).
4. The association between a candidate's record in the ballot definition file and the appearance on the screen of the AccuVote DRE voting machine wasn't cryptographically protected, thus verifying that a voter's intent was accurately recorded was impossible. The fact that a vote appeared in the confirmation screen (front end) was no guarantee that that vote was recorded and tabulated (back end). Diebold's "redundant" storage technique only ensured duplicate copies of the voting transaction, not that any of the transactions were correct.
5. The ballot definition file contained sensitive information like the terminal's ID number, the dial-in numbers for online tally reports, IP addresses for networked computers, and user names and passwords—all in plaintext.
6. The smartcards used by voters to authenticate with the voting machines used no cryptography at all. Therefore, anyone with the ability to create smart cards offsite and get them inserted into an AccuVote DRE station (trivial—see below) could authenticate to the machine and have votes recorded.
7. Election officials' administrative cards all had a default PIN of 1111.

Note that these are professional, technical comments—not parochial or political opinions. Most of us recognize the faults as rookie mistakes that wouldn't withstand scrutiny in a respectable college-level computer science programming class. Rubin claims in his book that this level of sloppiness was characteristic of the entire Accuvote TS code base. For making these deficiencies known, he was vilified by Diebold, sundry election officials, and an occasional politician. Such blowback against technical experts was repeated several times before the Diebold story had played out. We'll pass over the idiocy of making unprotected source code available

The Diebold debacle is fascinating chronicle of corporate irresponsibility, hubris, incompetence, political chicanery, and power politics.

through Diebold's website in silence. The point to remember is that Rubin's was the same low assessment of source code that any of us would give to our students. It was an accurate, fair, and legitimate criticism of sloppy work.

However, Rubin wasn't the Diebold code's only critic. A series of investigations by computer security specialist, Harri Hursti, proved to be even more embarrassing.

## THE HURSTI HACKS

In the mid-2000s, Hursti conducted several evaluations of the Diebold AccuVote system on behalf of Black Box Voting, Harris's election activist organization. The analysis was fairly extensive, pointing to deficiencies in the Diebold boot loader, removable memory, and easy-to-circumvent hardware security. Hursti correctly assessed Diebold's three-layer architecture as insecurity-in-depth, which is one step below security through obscurity.[5] Without question, the most alarming insecurity had to do with Diebold's removable memory cards. In the literature, these are the most prominent of the so-called Hursti hacks. These hacks were so simple and dramatic they were featured in the 2006 HBO documentary, *Hacking Democracy* (http://hackingdemocracy.com).

Hursti showed that the insecurities of the AccuVote TS and AccuVote TSx OSs were so substantive that even elementary changes to the code and/or data fields on the removable memory cards could change the outcome of elections. Despite the fact that these hacks had been demonstrated several times in several different jurisdictions, the initial response from Diebold was to attack the messenger(s). They even demanded that HBO cancel the previously mentioned documentary, but without effect.

Remember that Rubin published his analysis of the source code and Hursti followed up with experimental demonstrations of some result-altering hacks. Diebold's defense and counterclaims began to permanently unravel when University of California and then Princeton University researchers confirmed Hursti's results.[6,7] An earlier independent review by the Science Applications International Corporation (SAIC)—commissioned by the State of Maryland in 2003—had also reported that the "AccuVote-TS voting system is not compliant with the State of Maryland Information Security Policy and Standards ... and is at a high risk of compromise."[8]

A follow-up review by Maryland-based RABA Technologies, LLC, echoed the SAIC report, and added that the back-end management system (GEMS) was also insecure. So by the time *Hacking Democracy* came out, Diebold's proverbial cat was separated from the bag by light years. Diebold's response to these revelations was typical of the power elite: "... voters in the state of Maryland can now rest assured that they will participate in highly secure and accurate elections." Then-governor Robert L. Ehrlich Jr. (R), opined that "Because of this [SAIC] report, Maryland voters will have one of the safest election environments in the nation" (both quotes appear in *Brave New Ballot*,[1] pp. 137–138). There's no way to know whether the better explanation of Diebold's and Ehrlich's spin is cognitive dissonance or outright deceit, but whatever the reason, the known code insecurities went unattended for many years.

Although the hacks themselves are of only marginal historical significance at this point, the complex interplay among Diebold, the election officials who either tried to cover up the insecurities or expose them, the politicians who sought political cover from the exposures, and the computer scientists who uncovered the problems remains critical for little has changed to correct the problems. The capacity of the manufacturers, vendors, and election officials to conceal, cover up, and deceive is as great today as it was 15 years ago. But more is at stake now because DRE voting machines are ubiquitous, and we've since developed a tolerance for chicanery in our elections. There's another player that I haven't mentioned: the Independent Testing Authorities (ITAs) that "validate" these voting systems.

## ITAs AND VOTER "VERIFICATION"

Diebold, Sequoia, and Election Systems and Software (ES&S) came to dominate the digital voting equipment market by the early 2000s. After a few mergers and acquisitions cycles, Diebold and Sequoia became subsidiaries of Dominion Voting Systems. At this point, the competition has been narrowed to a very few players.

Once a voting system is developed, election officials might be deluded into a false sense of security by ITAs (now called Voting System Testing Laboratories) that certify the system's integrity. ITAs work in much the same way as credit ratings services (think Moody's, Standard and Poor's, and the Fitch Group), and they're bound by the same incentives. In all cases, the applicant pays for the service of certification—that is, the beneficiary of the certification provides the revenue stream to the certifier. Thus, if an ITA rejects certification of voting equipment (not likely), other ITAs are enlisted until one approves certification. This arrangement takes conflict of interest to a new plateau.

Further, ITAs/VSTLs only do extensional validation, which is to say they compare results by re-running election records with known outcomes or using canned datasets with well-defined data. That doesn't really contribute much confidence in the system if no one looks "under the hood." Chip design and circuit analysis aren't part of the validation because both are proprietary. No objective source code review is undertaken by skilled computer scientists, unless there's been an accidental leak like the one mentioned earlier. This incestuous relationship among ITAs, manufacturers, and technically ignorant election officials seeking to avoid public scrutiny of their activities is still with us today. This is why the Diebold story is still relevant. Although Diebold Election Systems and its amateurish source code is gone, the structural problems that gave rise to them in the first place are still with us.

If an ITA is to be effective, it must not only provide intentional validation that compares output from canned datasets, but it must also provide a functional analysis of the source code by impartial, skilled computer professionals. This is in effect what Rubin's, David Wagner's, and Ariel Feldman's teams did.[6,7] But at this writing, verification of voting systems' code might amount to nothing more than comparing hash signatures between file versions. Hash signatures are measures of binary identity—not the quality or integrity of code. (Primitive analysis restricted to I/O based on canned datasets is frequently referred to as blackbox analysis, which is the source of the name Harris chose for her elections activist and investigatory group Black Box Voting.)

Further, Diebold apparently didn't bother to run their source code through a commercial-quality source code analyzer. However, Wagner and his colleagues did.[6] In the research they prepared for the California Secretary of State, the Fortify (now HP) static code analyzer identified 16 security vulnerabilities in the AccuVote operating system ranging from array bounds violations and faulty input validation errors to buffer overruns, buffer underruns, and pointer errors. Although the specific details (location of code fragment, and so on) were suppressed from the public report, enough detail was included to convince any computing professional that the code base lacked integrity. Note that these 16 vulnerabilities would have been easily detected with the HP Fortify static source code analyzer had Diebold chosen to use it. I encourage readers to review these referenced reports and confirm for themselves that the Diebold source code used up to and including the 2006 national elections should be on display in the Smithsonian as a primitive artifact. Any thoughtful analysis confirms the computer scientists' claims: the code was amateurish, the security standards were embarrassingly weak, and the systems were fraught with vulnerabilities. According to *Democracy Hacked*, approximately 40 percent of all votes in the US were counted by systems that ran this code at the time of the analysis.

The Diebold source code used up to and including the 2006 national elections should be on display in the Smithsonian as a primitive artifact.

## DON'T WORRY, BE HAPPY

Some have claimed that DRE voting machine security isn't an issue:[9]

> *The conjecture that … we are unable to make such a simple system secure and accurate is contradicted by the facts of our everyday existence. We build secure and accurate computer systems that fly our airliners. We build secure and accurate computer systems that guide our submarines under the ice cap. We build secure and accurate computer systems that guide our astronauts to the moon and bring them safely back to earth. We submit to open heart surgery while a computer monitors our vital signs and controls an artificial heart and lung machine. The list of secure and accurate computer systems that monitor, control, and improve our lives is large and growing daily.*

The appropriate response to this argument is "that's true, but so what?" This is a patently silly position to take for a number of different reasons. For one, threat vectors must be understood in context. Who'd be incentivized to corrupt flight control systems? What might be gained if an airliner was off course? What relationship would the possible perpetrators likely have to the affected airlines? The same applies to moon landings, heart monitors, and the like. In each case, the likely threat would be terrorists or criminals, and external. The fear would be that someone outside the system wants to do harm to others and the incentive might be revenge, anger, hate, jealousy, greed, and so on—all motives that are visceral and personal.

Voting machines provide an entirely different context: the incentive is to subvert the democratic process toward partisan effect, and the likely perpetrator would be internal, or at least very closely linked to a specific political interest. Thus, the perp would likely be a partisan operative either employed by, or closely connected to, a candidate, party, PAC, or particular election officials. Murderers and terrorists tend to not work closely with domain knowledge experts on their weapons of choice. People that steal elections do.

Further, mission-critical systems rely on high-confidence software development paradigms. As shown, the Diebold code certainly wasn't high-confidence. If loss-of-life scenarios require high-confidence methodologies, loss-of-country scenarios make them similarly desirable. The Diebold DRE voting machines under discussion weren't trusted systems, rather, they're twisted systems in which minimal attention was paid to best practices in software development, software security, user privacy, software reliability, and so forth. The letter and spirit of industry standards in effect at the time the equipment was developed were violated, ensuring that expected results would be obtainable only under optimal circumstances in which all involved behaved properly and predictably and without serious corrupting external influences. Elections never offer such controlled environments.

In addition, as long as completely secret ballots are required, there's no way to fully and incontrovertibly audit the DRE voting machines' results. Even when paper-tape backup is used, the voting sequences are scrambled so the individual votes can't be recovered, and there's no voter verification of the vote, but only voter verification of the most recent behavior of the particular voting machine. Although some alternate voting systems have been proposed,[10] no commercial voting machines that I know allow each individual voter to verify that their vote was actually recorded by the digital vote management system and reported to state election officials correctly. For that level of assurance to result with current voting equipment, both the electronic and physical records have to be tallied and publicly reported independently. It's important to understand that the phrase "voter verified" normally refers to the voter's verification of the vote cast at the DRE voting terminal, not verification that it was recorded by the tally management system.

Murderers and terrorists tend to not work closely with domain knowledge experts on their weapons of choice. People that steal elections do.

## WAG THE DOG

In most important respects, little was learned from the Diebold fiasco. To be sure, DRE voting machine manufacturers are more attentive to source code integrity, but the degree is a matter of conjecture because the code is still proprietary and not available for inspection—not even after the election concludes. Lack of trustworthy code remains a real and present danger to election integrity and our democracy. Vilification of the computer scientists wasn't due to their scientific results—no reasonable person challenged their facts—but because they cast doubt on the accuracy and honesty of the election results. In other words, they were castigated for pointing out the obvious: no insecure computing systems are trustworthy, DRE voting machines included!

In addition, the few electronic voting machine manufacturers that remain are more circumspect in how they represent their product to their customers—the jurisdictions and the public. Although Diebold's arrogance and hubris waned as the company headed toward collapse, there's still no mechanism through which the public can establish confidence in these manufacturers' products and practices—too much is hidden from view.

Thus, we can't know the degree to which obsolete, insecure code and bad practices are in active use. It's also unknown whether, or to what extent modern source code is built around a valid security model. In 2016, vendors and manufacturers can still manipulate election officials who lack technical knowledge and skills. Voting machine procurement and approval processes face partisan brinksmanship, and any election official demanding independent testing faces threat of litigation from manufacturers and perhaps even election officials. The overwhelming majority of jurisdictions still fail to perform complete audits of election results, and what's more, there's evidence to suggest that those who scrutinize the fairness of elections might be subject to government surveillance.[11] These aren't good signs.

At its inception, digital voting technology promised to promote universal suffrage—enabling underprivileged, disadvantaged, and immobile voters to come to the polls, as well as related benefits such as the minimization of pressure from partisans and some mitigation against the historical vote-suppression techniques (such as long wait times, confusing ballots, miscounts, undercounts, discarded ballots, corrupted results, and so on).[12] However, DRE voting machines haven't delivered on the promise to help secure the election franchise for all citizens. The Diebold scandal revealed that there's far too much slop in the digital voting process at too many different levels. For that understanding, we're indebted to the computer scientists mentioned in the studies I've referenced here. They're the true heroes of this story, and the country owes them a great debt.

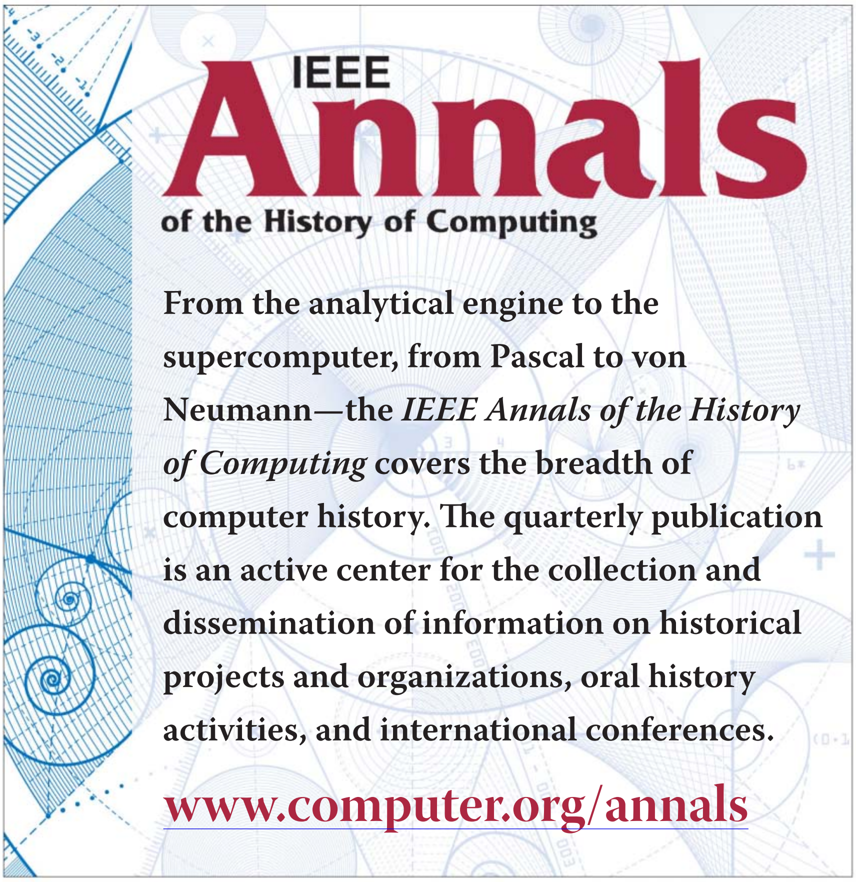
All in all, the way electronic voting is administered in the US still falls far short of reasonable expectations in terms of the ability to verify election outcomes;[13] to technically validate the equipment's source code; to achieve a public understanding of the systems' vulnerabilities; and to completely disclose possible conflicts of interest between vendors and their agents, public officials, and those engaged in vetting the integrity and certification of voting systems. In these areas, we're no farther along than we were 20 years ago. Our best hope at addressing these problems rests with computing professionals. Indeed, our goal should be to demand that these experts be centrally involved in vetting all future voting systems. After all, our first line of defense is the community of computing professionals who are willing to take risks and speak out.

For those interested in this topic, I recommend the seminal book, *Broken Ballots: Will Your Vote Count?*[20] C

## REFERENCES

1. A. Rubin, *Brave New Ballot*, Morgan Road Books, 2006.
2. H. Berghel, "Digital Politics 2016," *Computer*, vol. 49, no. 1, 2016, pp. 75–79.
3. L. Hoffmann, "Q&A: Finding New Directions in Cryptography" [interview with Whitman Diffie and Martin Hellman], *Comm. ACM*, vol. 59, no. 6, 2016, pp. 110–112.
4. J. Gilmore, "DES (Data Encryption Standard) Review at Stanford University," Toad Hall, 20 Sept. 2005 (with subsequent updates through 2015); www.toad.com/des-stanford-meeting.html.
5. H. Hursti, "Diebold TSx Evaluation—Critical Security Issues with Diebold TSx (unredacted)," security alert, Black Box Voting, 11 May 2006; www.blackboxvoting.org/BBVreportIIunredacted.pdf and Supplement www.blackboxvoting.org/BBVreportII-supplement-unredacted.pdf.
6. D. Wagner et al., "Security Analysis of the Diebold AccuBasic Interpreter," Voting Systems Technology Assessment Advisory Board report, Presidential Commission on Election Administration, 14 Feb. 2006; https://web.archive.org/web/20070611092341.
7. A. Feldman, J.A. Halderman, and E. Felten, "Security Analysis of the Diebold AccuVote-TS Voting Machine," *Proc. USENIX/ACCURATE Electronic Voting Technology Workshop* (EVT 07), 2007; www.usenix.org/legacy/events/evt07/tech/full_papers/feldman/feldman_html/index.html.
8. "Risk Assessment Report Diebold AccuVote-TS Voting System and Processes," report commissioned by the Maryland Dept. of Budget and Management, Science Applications Int'l Corporation (SAIC), 2 Sept. 2003 [redacted at the request of the State of Maryland]; www.ballot-integrity.org/docs/SAIC_Report.pdf.
9. B. Williams, "Presentation to the US Election Assistance Commission," presentation, 5 May 2004; www.eac.gov/assets/1/AssetManager/testimony%20brit%20williams%20kennesaw%20university%20public%20meeting%20may%205%202004.pdf.
10. D. Chaum, P. Ryan, and S. Schneider, "A Practical, Voter-Verifiable Election Scheme," *Proc. 10th European Symp. Research in Computer Security* (ESORICS 05), 2005, pp. 118–139; www.pretavoter.com/publications/esorics05.pdf.
11. G. Howland Jr., "www.bigbrother.com," *Seattle Weekly*, 9 Oct. 2006; www.seattleweekly.com/2004-05-19/news/www-bigbrother-gov.
12. T. Campbell, *Deliver the Vote: A History of Election Fraud, an American Political Tradition—1742–2004*, Carroll & Graf, 2004.
13. B. Schneier, "The Problem with Electronic Voting Machines," *Schneier on Security*, 10 Nov. 2004; www.schneier.com/blog/archives/2004/11/the_problem_wit.html.
14. D.W. Jones and B. Simon, *Broken Ballots: Will Your Vote Count?*, Center for the Study of Language and Information, 2012.

**HAL BERGHEL** is a professor of computer science at the University of Nevada, Las Vegas and is an IEEE and ACM Fellow. Contact him at hlb@computer.org.



CLOUD COVER

# Security and Risk Assessment in the Cloud

**Sanjay K. Madria,** Missouri University of Science and Technology

*Fears about the data security that cloud service providers offer is a primary reason many organizations haven't fully adopted cloud services. Key issues include data storage, data replication, integrity verification, access control, risk assessment, and secure query processing.*

Cloud computing offers consumers the ability to purchase scalable and cost-effective technology services over the Internet as needed, without having to buy or maintain the hardware and software that provide them.

The three primary cloud computing service models are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). The availability of robust network connectivity and high Internet bandwidth have made it feasible for organizations to move huge amounts of data to third-party cloud storage, leading to a new model: data as a service (DaaS).

The cloud paradigm has generated significant marketplace and academic interest, resulting in numerous cloud computing services from vendors such as Amazon, Google, Microsoft, Salesforce, and Yahoo. Also, vendors such as IBM and Oracle are adding cloud support to their databases.

In this article, we highlight issues involving data security and access control, as well as risk assessment of applications, all of which are critical to wider cloud adoption.

## DATA-STORAGE SECURITY CONCERNS

Hosting an application in the cloud removes it from an organization's secure private network. There is a lack of user control and transparency when a third party controls user data. In addition, hosted cloud applications are complex and not well understood.

Thus, security is a primary reason many organizations haven't fully adopted cloud services. A main issue is how cloud service providers (CSPs) secure and disseminate the data they host. This entails legal concerns, such as the different data-security and data-handling laws in the countries where CSPs store information.

Users are also worried about whether the cloud can maintain data integrity and whether they can recover information when there is a server failure or data loss. Such losses could occur if CSPs, to reduce storage costs, discard some seldom-accessed data.

Additionally, users want to be able to periodically verify whether CSP storage servers maintain data integrity and store information in conformance with service level

0018-9162/16/$33.00 © 2016 IEEE

EDITOR SAN MURUGESAN
BRITE Professional Services;
san@computer.org

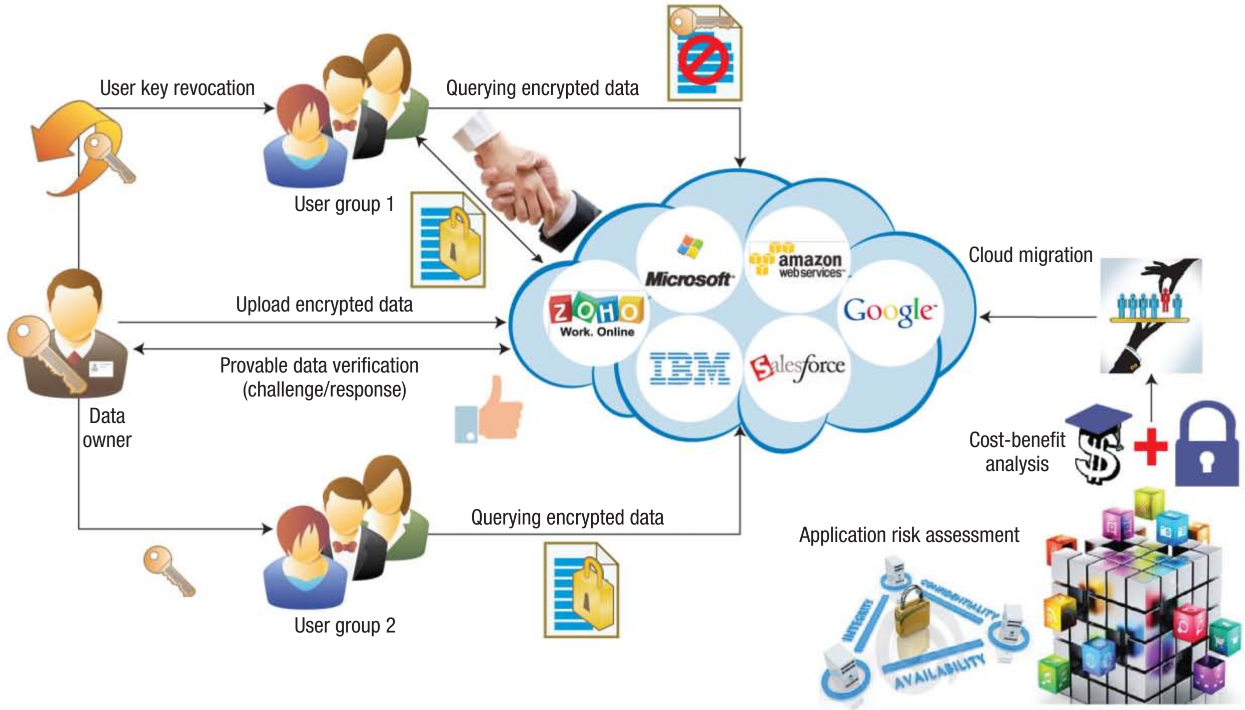


**Figure 1.** Provable data possession and risk assessment model. The data owner uploads encrypted information to the cloud and provides keys to users who can query the data and access attributes based on their profiles. The owners can periodically verify the integrity of their data in the cloud, as well as perform a risk assessment and cost-benefit analysis to determine which applications are safe to migrate to the cloud.

agreements (SLAs). They are concerned that providers might conceal losses due to poor data management, hardware failures, or cyberattacks.

## PROVABLE DATA POSSESSION

The provable data possession (PDP) technique audits and validates the integrity of data stored on cloud servers. In a typical PDP model, which Figure 1 shows, the data owner generates metadata and tags to be used for verifying the integrity of encrypted data files. The data owner sends the encrypted file and tags to the cloud, and deletes the local file copy. To verify the data's integrity, the owner generates a challenge vector and sends it to the cloud. The cloud-based data-protection application replies by computing a response that proves integrity.

Verification schemes, such as the Markle hash tree (MHT), use various cryptographic approaches, but most deal only with static data files and verify only a single copy of data in one cloud. With these approaches, remotely stored data in a single cloud can be accessed by authorized users but can't be updated or scaled by the owner. Recently a few schemes such as ID-DPDP (identity-based distributed PDP)[1,2] have enabled limited dynamic data scalability in multiple clouds without the need for certification management. The ID-PUIC (identity-based proxy-oriented data uploading and remote data integrity checking in public cloud) protocol[2] works with fixed and mobile platforms.

## DATA REPLICATION AND INTEGRITY VERIFICATION

Geo replication—which improves data distribution and access across geographically distributed networks—increases data availability in cloud computing. The cloud system replicates data and stores it on multiple servers at various locations. This helps owners, who might not have a copy stored locally or who might want multiple copies available in case one is deleted, corrupted, or compromised.

A CSP could easily cheat a data owner who expects multiple copies to be stored by keeping only one copy, thereby conserving storage capacity and receiving revenue without having to provide a service. Thus, an owner might like to regularly verify whether the cloud indeed possesses multiple copies of the data as claimed in the SLA, using schemes such as Paillier encryption[3] to create multiple copies of a file and Boneh-Lynn-Shacham to create data tags that include signatures that help with data-integrity verification.

Concerns about security are one of the major reasons why many organizations haven't fully adopted cloud-based services.

In some integrity-verification schemes,[3,4] data owners share a decryption key with authorized users and thus don't need CSPs to give users access to files. These schemes' homomorphic properties[5] make updating files more efficient. The data owner just has to encrypt the new information that is updating the old file and send it to the cloud, which then updates all copies via homomorphic addition. Authenticated data structures such as MHTs and skip lists can be used with this scheme to ensure that the cloud uses the correct file blocks for data-integrity verification. However, homomorphic approaches execute slowly and, therefore, more efficient schemes are needed to reduce latency.

## SECURE QUERY PROCESSING AND ACCESS CONTROL

Secure query processing (SQP) of encrypted data—designed to reveal results only to authorized users making requests—has attracted significant attention because of database-service outsourcing's growing popularity. Academic and industry researchers are working on cloud-related SQP issues[6] that would provide secure operations[7], including efficient SQP protocols for use with encrypted data.

DaaS makes information in the cloud available to customers authorized to see it, based on rights the data owner has granted. Different customers could have different access rights. Enforcing such fine-grained access control (FAC) in the cloud is thus of primary importance, not only in DaaS but also in service models such as SaaS. General approaches rely heavily upon encryption. However, this entails concerns not only about querying encrypted data but also about

- keeping users away from data attributes they're not authorized to access,[8]
- efficiently revoking users' privileges without having to re-encrypt huge amounts of data and redistributing new keys to authorized users,
- preventing collusion between users and CSPs, and
- changing a user's access privileges.

Coupling access control with re-encryption to prevent unauthorized access can provide FAC. The considerable work in this area[6,8,9] has been based on combining attribute-based encryption and proxy re-encryption. To provide better security, some proposed schemes use additive homomorphic encryption and proxy re-encryption.[6]

Most cloud computing security research assumes that the cloud will silently observe data and query execution results but will not disturb the information and also that there is no collusion between the CSP and a revoked data consumer, between users, or between CSPs. However, this might not always be the case, so data owners want ways to prevent data leakage in case of collusion.[10]

They also want ways to

- efficiently and securely handle access rights, revoke those rights, and issue either the same or different rights to a returning user;
- cope with clouds that secretly collect stored information;
- avoid collusion among users;
- extend existing query processing of encrypted data to the federated cloud; and
- develop cloud services that use multiple vendors.

Trusted computing and applied cryptographic techniques might offer new tools to solve these problems. Nonetheless, more research is needed on designing better algorithms to alleviate fear of cloud-related security problems.

Data owners often don't receive or have sufficient information about a CSP's data privacy and security mechanisms to determine how well their information is protected and thus have been reluctant to fully adopt cloud technology. Some concerns include whether the CSP has done its due diligence regarding security; system auditability; whether a provider can meet SLA-related obligations; espionage by the CSP; data lock-in; and data-control issues in multiprovider federated clouds. Again, approaches such as trusted computing and applied cryptography might help data owners ensure their information's security. However, more research needs to be done to alleviate much of today's concerns about cloud security.

Organizations could hire third-party evaluators or follow some security assessment guidelines although doing so depends on the expertise of the third-party evaluators. Organizations need an offline risk-assessment framework[11] to help them select suitable CSPs. The framework should evaluate CSP security and suggest trustworthy providers based on threats it identifies to an enterprise application. Organizations could then select a CSP via a cost–benefit analysis.

Finding a CSP that will provide a secure cloud infrastructure that facilitates secure application migration, data protection, FAC, and safe information retrieval is key for organizations. Further research and collaboration among CSPs, researchers, users, and regulatory agencies are required to achieve this, address cloud security concerns, and drive cloud adoption in multiple domains. 🅲

## REFERENCES

1. H. Wang, "Identity-Based Distributed Provable Data Possession in Multicloud Storage," *IEEE Trans. Services Computing*, vol. 8, no. 2, 2015, pp. 328–340.
2. H. Wang, D. He, and S. Tang, "Identity-Based Proxy-Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud," *IEEE Trans. Information Forensics and Security*, vol. 11, no. 6, 2015, pp. 1165–1176.
3. R. Mukundan, S. Madria, and M. Linderman, "Efficient Integrity Verification of Replicated Data in Cloud Using Homomorphic Encryption," *Distributed and Parallel Databases*, vol. 32, no. 4, 2014, pp. 507–534.
4. R. Mukundan, S. Madria, and M. Linderman, "Replicated Data Integrity Verification in Cloud," *Bull. IEEE Computer Society Technical Committee Data Eng.*, vol. 35, no. 4, 2012, pp. 55–64.
5. C. Gentry, A. Sahai, and B. Waters, "Homomorphic Encryption from Learning with Errors: Conceptually Simpler, Asymptotically Faster, Attribute Based," *Proc. 33rd Int'l Cryptology Conf.* (Crypto 13), 2013, pp. 75–92.
6. B.K. Samanthula et al., "A Secure Data Sharing and Query Processing Framework via Federation of Cloud Computing," *Information System J.*, vol. 48, 2015, pp. 196–212.
7. V. Khadilkar et al., "Secure Data Processing over Hybrid Clouds," *Bull. IEEE Computer Society Technical Committee Data Eng.*, vol. 35, no. 4, 2012, pp. 46–54.
8. T. Jung et al., "Control Cloud Data Access Privilege and Anonymity with Fully Anonymous Attribute-Based Encryption," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 1, 2015, pp. 190–199.
9. S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 2, 2014, pp. 384–394.
10. B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," *IEEE Trans. Services Computing*, vol.8, no.1, 2015, pp. 92–106.
11. A. Sen and S. Madria, "Offline Risk Assessment of Cloud Service Provider," *IEEE Cloud Computing*, vol. 2, no. 3, 2015, pp. 50–57.

**SANJAY K. MADRIA** is a full professor in the Missouri University of Science and Technology's Department of Computer Science. Contact him at madrias@mst.edu.

# Dragging Government Legacy Systems Out of the Shadows

**Robert N. Charette,** ITABHI Corporation

*Several decades' worth of internal and external audits of failed government IT modernization efforts suggest that a legislative solution is wishful thinking. It's time to examine the organizational dynamics and behavioral motivations underlying the problem.*

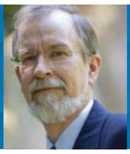
What does the US Department of the Treasury's Individual Master File (IMF) software—the authoritative data source on American taxpayers—have in common with Apple's CEO? Like Tim Cook, the IMF will be celebrating its 56th birthday this year, according to a May 2016 US Government Accountability Office (GAO) report on legacy IT investments.[1] The report identified the IMF and its same-aged companion, the Business Master File (which contains the tax information related to individual business-income taxpayers), as the government's two oldest operating software systems. Indeed, nearly 75 percent of today's Americans weren't yet born when the systems went operational. Both are written in assembly code and execute on IBM mainframes—and there are no immediate plans to replace either of them.

The Treasury Department's elderly software systems are by no means unique. Coming in at 53 years of age are the US Air Force's Strategic Automated Command and Control System (SACCS), which coordinates the operational functions of US nuclear forces, and the US Department of Veterans Affairs Personnel and Accounting Integrated Data (PAID) system for tracking VA personnel time and attendance. The Air Force system runs on a 1970s-era IBM Series/1 minicomputer and depends on 8-inch floppy disks, while the VA system is written in COBOL and executes on an IBM mainframe. SACCS is slated for an upgrade in 2017, whereas PAID is scheduled to be replaced next year.

The GAO report lists 10 systems that are 39 years old or older (6 are over 50 years old), with half not currently scheduled for either replacement or modernization. In addition, the GAO found that operations and maintenance

0018-9162/16/$33.00 © 2016 IEEE

EDITORS
**HAL BERGHEL** University of Nevada, Las Vegas; hlb@computer.org
**ROBERT N. CHARETTE** ITABHI Corp.; rncharette@ieee.org
**JOHN L. KING** University of Michigan; jlking@umich.edu



(O&M) are absorbing the entire budget of 5,233 of the 7,000 or so IT investments by federal agencies. In total, at least $61.2 billion of the US government's current $79.4 billion unclassified IT budget is going to O&M. More concerning, the percentage of O&M expenditures has been steadily climbing (due in part to funding constraints) from around 68 percent in fiscal year 2010 to 76 percent in fiscal year 2017, leaving increasingly less funding for IT system modernization or replacement.

The GAO report warns that archaic IT systems pose "a giant problem" in performing the job of government cost-effectively and efficiently, not to mention the threats of operational failure and unauthorized intrusion. For example, in 2015, outdated financial IT systems contributed to at least $136.7 billion in improper federal payments.[2] The actual cost is likely higher, but without the ability to do a complete financial audit, how much higher is uncertain. With federal IT spending expected to remain flat for the next five years, the risks posed by outmoded IT systems will only grow.[3]

## MODERNIZATION OR BUST: COMPETING APPROACHES

In response to the ticking time bomb of obsolescent IT systems, especially in regard to their cybersecurity vulnerabilities, the Obama administration has proposed creating a $3.1 billion Information Technology Modernization Fund (ITMF)—basically a line of credit—managed by the US General Services Administration (GSA) and provided to government agencies to modernize their IT infrastructure.[4] Upon completing the modernization, the agencies would be required to pay back the funding over time, based upon the garnered savings in O&M expenditures. The administration claims that the ITMF would fund $12–15 billion in modernization projects over the next decade alone.

Initial agencies to receive ITMF funding would be identified by an appointed board of technical and financial experts as those having IT systems most in need of replacement. Agency CIOs could also propose modernization efforts to the board. Another ITMF funding priority would be replacing multiple legacy systems with fewer common platforms that could also be used to facilitate cross-government transition to those same platforms. Furthermore, to encourage use of agile development techniques, ITMF funding could be tied to work products rather than be an annual appropriation.

The ITMF proposal additionally calls for IT experts in acquisition and development to help select and execute modernization efforts, with a publicly accessible online IT dashboard showing the status and progress of each such effort.

While acknowledging possible pushback to elements of the proposal, especially the notion of "borrowing" funding, the administration warns that the status quo is unsustainable—government IT systems can't be kept on indefinite life support.[5]

As an alternative to the ITMF, a bipartisan group of senators and congresspersons led by Representative Will Hurd (R-Texas) proposed the Modernizing Outdated and Vulnerable Equipment and Information Technology (MOVE IT) Act of 2016.[6] This bill wouldn't create a central funding pool for legacy IT system modernization but instead would allow each government agency to set up its own working capital fund by reprogramming current O&M funding along with discretionary appropriations. The fund could be used to replace legacy IT systems, transition to cloud computing and innovative platforms or technologies, address security vulnerabilities, or modernize/enhance agency IT activities. Agencies would need to spend the money within three years, and could use any realized savings to replenish their working capital.

Proponents argue that MOVE IT will get more buy-in from agencies, who will retain full control of their IT modernization efforts without GSA involvement. Agencies will also have far more flexibility in funding projects, as they won't have to wait on yearly congressional appropriations. Moreover, it avoids the tricky issue of what happens if an agency fails to repay an ITMF loan.

Which approach is likely to prevail is unclear at this time. A combination of elements of the two is probable. Whatever approach is adopted, modernizing legacy IT systems is clearly going to be a high governmental priority in the near future.

## THE LEGACY OF LEGACY SYSTEMS: THE MORE THINGS CHANGE ...

Current calls to modernize government IT systems aren't new. In 1980, a GAO report warned that obsolete computer systems were significantly increasing the cost and lowering the productivity of government operations.[7] It stated that most of the government's large and medium-size computer systems were no longer fit for purpose, yet few agencies planned to replace them. Even then, less than 5 percent of these systems were under 5 years old, whereas 60 percent were 15 years old or more. Further, agency IT staff spent nearly two-thirds of their time maintaining existing software rather than developing new software. A companion report by then US Comptroller General Elmer Staats succinctly described the situation: "Software maintenance in the Government is now largely undefined, unquantified, and undermanaged."[8]

In light of these alarming reports, the newly elected Reagan administration adopted an aggressive IT modernization strategy that continued under the George H.W. Bush and Clinton administrations. However, after more than a dozen years the problem of obsolete federal IT still hadn't been resolved and, in many ways, was worse. Efforts to modernize aging computer systems by the Federal Aviation Administration (FAA), Department of Defense (DoD), National Weather Service, and other agencies were in deep trouble or had resulted in costly failures. For example, in 1994, after spending $2.6 billion, the FAA terminated its nearly 15-year attempt to bring the nation's air-traffic control system out of the vacuum-tube era.[9]

An obvious question is whether the current legislative efforts to modernize federal government IT and reign in O&M costs will succeed when all previous exertions have failed.

That same year, Senator William Cohen (R-ME) authored a widely circulated critique, *Computer Chaos: Billions Wasted Buying Federal Computer Systems*, in which he stated, "The Federal government continues to operate old, obsolete computer systems while it has wasted billions of dollars in failed computer modernization efforts. … [T]he system is indeed broken and it is time to fix it."[10] He described in depth the government's ineffectiveness, if not outright ineptitude, when acquiring, overseeing, and maintaining its IT systems. The report contained eight major recommendations, including emphasizing early oversight and planning; only approving projects of manageable scope, cost, and schedule; and halting existing procurements until the government's IT acquisition process was improved.

Senator Cohen's report, along with other ongoing IT project failures like the slow-motion train wreck of the Internal Revenue Service's $4 billion modernization effort, spurred Congress in 1996 to pass the Clinger–Cohen Act, which sought to reform government IT acquisition and management once and for all.[11,12] The Act had several ambitious goals, such as directing agencies each year for the next five years to achieve at least a 5 percent decrease in O&M costs (in constant fiscal year 1996 dollars) along with a 5 percent increase in operational efficiency through improvements in management. It also elevated CIOs to senior management and directed them to report directly to their top agency administrator in order to give them more input into IT project selection, funding, and deployment.

Unfortunately, but not unexpectedly, the Clinger–Cohen Act's implementation left something to be desired. For example, overall IT spending including O&M went up, not down, by 9 percent annually by 2006.[13] In addition, by 2011, 15 years after the Act's passage, only 17 of 30 CIOs were reporting to their agency's top administrator.[14] Further, cybersecurity threats—something not even highlighted in the Cohen report—became an increasingly troubling O&M concern for agencies.

The Act failed to meet another key objective—reducing large government IT failures. Billions of dollars were wasted on the Census Bureau's plan to outfit field workers with handheld computers ($2 billion, canceled in 2008), the Army's software-dependent Future Combat System ($18 billion, canceled in 2009), and the Department of Homeland Security's SBINet "virtual fence" border control system ($1 billion, canceled in 2011), among other projects.[15–17] In 2013, the disastrous and costly rollout of the HealthCare.gov website, coupled with a number of major cybersecurity incidents including the theft of classified information on the F-35 Joint Strike Fighter and several other weapon systems, accentuated the urgent need to gain control of federal IT systems development, operations, and maintenance.[18,19]

In 2014, Congress enacted the Federal Information Technology Acquisition Reform Act (FITARA) in yet another stab at wasteful government IT spending.[20] FITARA aims to enhance (again) the authority and decision-making power of agency CIOs not only in budgeting and acquisition, but in hiring as well. It also attempts to improve the risk management, cost efficiency, and effectiveness of federal IT investments, especially in the areas of datacenter consolidation and hardware, software, and services procurement. To avoid meeting the same fate as the Clinger–Cohen Act, FITARA mandates that agencies rate their progress in meeting the legislation's requirements.

Although FITARA's main goal is to improve IT governance, it's also hoped that CIOs can convince senior decision makers to recognize IT's central importance to accomplishing their agency's mission. However, FITARA doesn't provide any funding to address IT modernization issues, especially in regard to improving cybersecurity or moving to a cloud computing platform—hence the dual ITMF and MOVE IT proposals to upgrade legacy IT systems.

## MODERNIZATION MEPHISTOPHELES

An obvious question is whether the current legislative efforts to modernize federal government IT and reign in O&M costs will succeed when all previous exertions have failed. Even a cursory review of the numerous *IEEE Spectrum* articles detailing failed modernization efforts of the past decade reveals a distressing recurrent pattern that should temper any optimism.[21]

For example, Federal CIO Tony Scott claims that agency CIOs will now be held accountable for how well they manage and modernize their agency's IT, but that seems unlikely given that the average tenure of government CIOs is only two years (versus six in industry).[22,23]

Many agency executives aren't too bothered by failure anyway. For example, the Air Force canceled the Expeditionary Combat Support System (ECSS) in 2012 after spending $1 billion and some eight years to deliver a critical logistics system that, the Air Force admitted, didn't yield "any significant military capability."[24] Asked whether anyone would be demoted or fired, the last general in charge of the project said no one believed that it was necessary.[25] Apologies for the wanton waste also don't seem necessary. An internal Air Force review even went so far as to proclaim that ECCS shouldn't be seen as a failure but as "the first step to understanding" what needed to be done.[26]

This "What, me worry?" perspective is reinforced by the pervasiveness of what NASA Inspector General Paul K. Martin terms "Hubble Psychology,"[27] in which project personnel (government workers and contractors alike) expect that "projects that fail to meet cost and schedule goals will receive additional funding and that subsequent scientific and technological success will overshadow any budgetary and schedule problems."[27] A corollary to this optimism bias is that if the project does get canceled, program managers are confident that it'll be resurrected again sometime in the future. The ECSS project, for example, was the Air Force's third major failed attempt to deliver a modernized logistics system since the 1980s. One can surmise that learning from past mistakes isn't a requirement for government IT modernization, claims to the contrary notwithstanding.

Even public scrutiny isn't much of a deterrent. Since 2009, the US government has made a concerted effort to make the progress of modernization projects more transparent by publishing progress data on an IT dashboard. When the dashboard was rolled out, then Federal CIO Vivek Kundra said that it would enable "better decision-making, giving us the ability to turn around poorly performing projects and to divest from those which no longer make sense," because agency CIOs would be rating the projects.[28]

However, as the GAO reported in June of this year, federal agency CIOs are an optimistic bunch who are more likely to believe that their projects are less risky than they really are.[29] For instance, the GAO examined 95 IT investments across 15 different agencies and found that 60 of them are at greater risk of failure than was indicated by the CIOs, including 10 rated as low risk that were in fact high risk. On the positive side, 13 projects were rated as being higher risk than the GAO said they should be.

In addition, the GAO found that many agencies didn't provide timely data for display on the dashboard, making it difficult to track projects' status. IT modernization efforts can easily be identified as being at medium or even low risk of failure all the way up until the day they're canceled or, more likely, restarted. For example, the US Social Security Administration (SSA)'s new Disability Case Processing System (DCPS) was officially listed as a low-risk project even though for 5 consecutive years Release 1.0 was projected to be 24–32 months away from completion.[30] Not surprisingly, when a working release of DCPS was finally deployed, it fared poorly. After repeated attempts to fix it, the SSA decided in 2015 to write off nearly the entire initial investment of $311 million and start the DCPS project again from scratch at an additional cost of at least $131 million.[31] The dashboard also listed ECSS as being a medium-risk project though it was widely acknowledged to be spiraling out of control.

Another recurring and predictable issue involves trust—or, better put, mistrust. An Institute for Defense Analyses (IDA) report on repeated DoD enterprise resource planning (ERP) modernization project failures found that "Program managers are unable to deliver a completely factual version of their status to leadership if it contains any element that could be considered significantly negative. … Program managers fear that an honest delivery of program status will result in cancellation. As a result of this, leadership is unable to be effective in removing obstacles to program success."[32] In fact, the IDA noted, program personnel needed "courage" to deliver bad news to senior agency leadership, as doing so might be career threatening. This phenomenon isn't confined to the DoD by any means.

Finally, there is a generally negative perception of IT O&M within government. In the words of British technology historian David Edgerton, O&M exists in a "twilight world" even though the machinery of government is fully dependent upon it.[33] A 1981 *New York Times* article on federal IT modernization efforts stated that O&M had all "the appeal of technological janitorial work,"[34] and little has changed since then. A recent Partnership for Public Service report found that many federal IT support staff feel "disconnected" from their agency and not regarded as being integral to its success.[35]

Several decades' worth of internal and external audits of abortive government IT modernization efforts suggest that trying to legislate away the problem is wishful thinking. As safety expert Sidney Dekker would contend, it's time to examine the organizational dynamics and behavioral motivations underlying such projects' "drift into failure."[36] With more than $200 billion to be spent on IT system development, modernization, and enhancement at the federal level and another $150 billion at the state and local levels over the next decade, not to mention hundreds of billions more for O&M on obsolete systems, waiting will be costly. Otherwise, by around 2025 we can expect another round of disheartening reports and proposed legislative solutions to the same old IT legacy problems.

## REFERENCES

1. D.A. Powner, *Information Technology: Federal Agencies Need to Address Aging Legacy Systems*, Testimony before the US House Committee on Oversight and Government Reform, GAO-16-696T, US Govt. Accountability Office (GAO), 25 May 2016; www.gao.gov/products/GAO-16-696T.
2. J. Davidson, "U.S. Made $136.7 Billion in Improper Payments in 2015, a Growing Problem," *The Washington Post*, 14 July 2016; www.washingtonpost.com/news/powerpost/wp/2016/07/14/u-s-made-136-billion-in-improper-payments-in-2015-a-growing-problem.
3. F. Konkel, "Deltek: Federal IT Spending to Remain Flat over Next 5 Years," *Nextgov*, 13 July 2016; www.nextgov.com/cio-briefing/2016/07/deltek-federal-it-spending-remain-flat-over-next-5-years/129890.
4. T. Scott, "Improving and Modernizing Federal Cybersecurity," blog, The White House, 8 Apr. 2016; www.whitehouse.gov/blog/2016/04/08/improving-and-modernizing-federal-cybersecurity.
5. M. Rockwell, "Scott: ITMF Paybacks Won't Be an Issue," *Federal Computer Week*, 29 June 2016; https://fcw.com/articles/2016/06/29/itmf-scott-payback.aspx.
6. "Reps. Hurd, Connolly & Sens. Moran, Udall Introduce the MOVE IT Act," press release, Office of Congressman Will Hurd, 14 July 2016; https://hurd.house.gov/media-center/press-releases/reps-hurd-connolly-sens-moran-udall-introduce-move-it-act.
7. *Continued Use of Costly, Outmoded Computers in Federal Agencies Can Be Avoided*, Report of the US Comptroller General to Congress, AFMD-81-9, US GAO, 15 Dec. 1980; http://gao.gov/products/AFMD-81-9.
8. *Federal Agencies' Maintenance of Computer Programs: Expensive and Undermanaged*, Report of the US Comptroller General to Congress, AFMD-81-25, 26 Feb. 1981; www.gao.gov/assets/140/132087.pdf.
9. R.N. Charette, "Large-Scale Project Management Is Risk Management," *IEEE Software*, vol. 13, no. 4, 1996, pp. 111–117.
10. W.S. Cohen, *Computer Chaos: Billions Wasted Buying Federal Computer Systems*, US Senate Subcommittee on Oversight of Govt. Management, 12 Oct. 1994.
11. D.C. Johnston, "Computers Clogged, I.R.S. Seeks to Hire Outside Processors," *The New York Times*, 31 Jan. 1997; www.nytimes.com/1997/01/31/us/computers-clogged-irs-seeks-to-hire-outside-processors.html.
12. US Code: Title 40—Public Buildings, Property, and Works; Subtitle III—Information Technology Management, pp. 179–191; www.gpo.gov/fdsys/pkg/USCODE-2011-title40/pdf/USCODE-2011-title40-subtitleIII.pdf.
13. W. Andrues, "The Clinger-Cohen Act, 10 Years Later: The Five Percent Solution," part 1, *Government Executive*, 11 July 2006; www.govexec.com/technology/2006/07/the-clinger-cohen-act-10-years-later-the-five-percent-solution/22226.
14. *Opportunities Exist to Improve Role in Information Technology Management*, Report to the US Senate Committee on Homeland Security and Governmental Affairs, GAO-11-634, US GAO, Sept. 2011; www.gao.gov/assets/590/585305.pdf.
15. R.N. Charette, "Census: Going Back to Paper Due to 'Lack of Communication,'" *IEEE Spectrum*, 4 Apr. 2008; http://spectrum.ieee.org/riskfactor/computing/it/census_going_back_to_paper_due.
16. R.N. Charette, "US Army's Future Combat Systems Program Formally Terminated," *IEEE Spectrum*, 24 June 2009; http://spectrum.ieee.org/riskfactor/computing/it/us-army-future-combat-systems-program-formally-terminated.
17. R.N. Charette, "Napolitano Cancels the US $1 Billion SBInet Virtual Fence Project," *IEEE Spectrum*, 28 Feb. 2011; http://spectrum.ieee.org/telecom/security/napolitano-cancels-the- us-1-billion-sbinet-virtual-fence-project.
18. W.D. Jones, "The Obamacare Rollout: What Really Happened?," *IEEE Spectrum*, 4 Nov. 2013; http://spectrum.ieee.org/riskfactor/computing/it/the-obamacare-rollout-what-really-happened.
19. E. Nalashima, "Confidential Report Lists U.S. Weapons System Designs Compromised by Chinese Cyberspies," *The Washington Post*, 27 May 2013; www.washingtonpost.com/world/national-security/confidential-report-lists-us-weapons-system-designs-compromised-by-chinese-cyberspies/2013/05/27/a42c3e1c-c2dd-11e2-8c3b-0b5e9247e8ca_story.html.
20. S. Donovan, "Management and Oversight of Federal Information Technology," memo M-15-14, US Office of Management and Budget, 10 June 2015; www.whitehouse.gov/sites/default/files/omb/memoranda/2015/m-15-14.pdf.
21. R.N. Charette and J. Romero, "Lessons from a Decade of IT Failures," *IEEE Spectrum*, 16 Oct. 2015; http://spectrum.ieee.org/static/lessons-from-a-decade-of-it-failures.
22. M. Rockwell, "Scott: FITARA Marks Sea Change for IT," *Federal Computer Weekly*, 11 Aug. 2015; https://fcw.com/articles/2015/08/11/fitara-tony-scott.aspx.
23. J. Moore, "Are CIOs to Blame for the Legacy IT Crisis?," *Nextgov*, 1 Apr. 2016; www.nextgov.com/cio-briefing/2016/04/are-cios-blame-legacy-it-crisis/127171.
24. R.N. Charette, "U.S. Air Force Blows $1 Billion on Failed ERP Project," *IEEE Spectrum*, 15 Nov. 2012; http://spectrum.ieee.org/riskfactor/aerospace/military/us-air-force-blows-1-billion-on-failed-erp-project.
25. S. Reilly, "How the Air Force Blew $1B on a Dud System," *Air Force Times*, 4 Apr. 2013.
26. R.N. Charette, "The U.S. Air Force Explains Its $1 Billion ECSS Bonfire," *IEEE Spectrum*, 6 Dec. 2013; http://spectrum.ieee.org/riskfactor/aero

space/military/the-us-air-force-explains-its-billion-ecss-bonfire.

27. P.K. Martin, *NASA's Challenges to Meeting Cost, Schedule, and Performance Goals*, report no. IG-12-021, NASA Office of the Inspector General, 27 Sept. 2012; https://oig .nasa.gov/audits/reports/FY12/IG-12 -021.pdf.
28. T. O'Reilly, "Radical Transparency: The New Federal IT Dashboard," *Radar*, 30 June 2009; http://radar.oreilly.com/2009/06/radical-transparency-federal-it-dashboard.html.
29. *IT Dashboard: Agencies Need to Fully Consider Risks When Rating Their Major Investments*, Report to Congressional Requesters, GAO-16-494, US GAO, 2 June 2016; www.gao.gov/products /GAO-16-494.
30. "Social Security Spent $300M on 'IT Boondoggle,'" Associated Press, 24 July 2014; http://bigstory.ap.org/article/social-security-spent-300m-it-boondoggle.
31. K. Byrd, *Modernizing Social Security's Information Technology Infrastructure*, Testimony of the US Social Security Administration Office of the Inspector General before the US House Committee on Ways and Means, 14 July 2016; http://waysandmeans.house.gov/wp-content/uploads/2016/07/20160714SS-Testimony-Byrd.pdf.
32. P.K. Ketrick et al., *Assessment of DoD Enterprise Resource Planning Business Systems*, IDA Paper P-4691, Inst. for Defense Analyses, Feb. 2011; www.acq.osd.mil/parca/docs/2011-ida-erp-business-systems-p-4691.pdf.
33. D. Edgerton, *The Shock of the Old: Technology and Global History since 1900*, Profile Books, 2006.
34. A. Pollack, "Old Programs, New Problems," *The New York Times*, 24 Nov. 1981; www.nytimes.com/1981/11/24/business/old-programs-new-problems.html?pagewanted=all.
35. R.W. Walker, "'Back Office' Professionals in Government Feel Disconnected from Agency Missions—Report," *FedScoop*, 18 July 2016; http://fedscoop.com/back-office-professionals-in-government-feel-disconnected-from-agency-missions-report.
36. S. Dekker, *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems*, Ashgate Publishing, 2011.

**ROBERT N. CHARETTE**, coeditor of the Aftershock column and founder of ITABHI Corporation, is an internationally acknowledged authority and pioneer in risk management, systems engineering, and the lean development and management of large-scale software-intensive systems. Contact him at rncharette@ieee.com.

# CS CONNECTION

## IEEE COMPUTER SOCIETY LAUNCHES E-LEARNING HACKING COURSE

The IEEE Computer Society (CS) and NotSoSecure, a leading UK-based penetration testing and hacking training company, join forces to offer The Art of Hacking, the industry's first combined e-learning, skills-based hacking course.

The course contains training in both Web and infrastructure hacking, and teaches students to think like the enemy to effectively stay ahead of threats to their defense perimeter. Containing five introductory/intermediate-level modules—three on infrastructure hacking and two on Web hacking—the course is structured to have 75 percent hands-on "HackLab" experience, and 25 percent assessment. By completing the "Capture the Flag" examination at the course's conclusion, students can earn one of two certification levels: Art of Hacking Ninja or Art of Hacking Master.

Newest cases, threats, hacks, and intrusions are continuously included in the HackLab to ensure that students can protect themselves against the latest threats.

For more details, visit www.computer.org/artofhacking.

## TECHNICAL COMMITTEE ON DATA ENGINEERING ANNOUNCES 2016 AWARDEES

Since 2014, the IEEE Computer Society's Technical Committee on Data Engineering (TCDE) has honored innovations and long-term contributions to the data engineering field through several TCDE-sponsored awards.

TCDE's Executive Committee recently announced the 2016 awardees for the TCDE Early Career Award; the TCDE Computer Science, Engineering, and Education (CSEE) Impact Award; and the TCDE Service Award.

Arnab Nandi, of The Ohio State University, received the TCDE Early Career Award for his contributions to user-focused data interaction technologies—namely, building data analysis, exploration, and querying systems that allow highly interactive experiences for end users. Nandi's work on gesture-based database querying allows users to interact with databases without the need for a keyboard. Such work has far-reaching impact for consumer devices, vehicles, manufacturing, and healthcare.

Mike Carey, of the University of California, Irvine, received the TCDE CSEE Impact Award for leadership and research excellence in building impactful data-management systems, engineering tools, products, and practices. Carey's career in building systems that combined research with solid engineering produced a string of systems that have consistently been in the forefront of the technical state of the art: Wisconsin Storage Subsystem, Exodus, Shore, BEA, Asterix, and many others. He also helped establish the University of Wisconsin as a unique place for university-centered work on database systems. His work there resulted in a series of landmark papers that laid the foundations for current and future database systems.

For being a driving force behind the creation of the IEEE International Conference on Data Engineering (ICDE), Gio Wiederhold of Stanford University was awarded the TCDE Service Award. In the early 1980s, bringing practical research results to leading database research community conferences was a particular challenge. Recognizing that the narrow focus on "databases" should be expanded to "data management," and that the "engineering" (namely, system building and application) of the technology is just as important, Wiederhold spearheaded ICDE's creation, leading to the first meeting in Los Angeles in 1984.

The TCDE Award Committee selects winners, who are recognized at TCDE's flagship conference, ICDE, and publicized on the TCDE website. For more information about TCDE awards and the nomination process, visit http://tab.computer.org/tcde/tcdeawards.html.

## NOMINEES SOUGHT FOR CS AWARDS

Honor your profession and the leaders in your industry by submitting nominations for IEEE Computer Society awards. Award categories span achievement, service, and education. The nominations deadline is **15 October 2016**.

For complete details, visit https://awards.computer.org/ana/award/viewAwards.action.

0018-9162/16/$33.00 © 2016 IEEE

# CELEBRATING THE LIFE OF T. MICHAEL ELLIOTT (1942–2016)

The IEEE Computer Society's (CS's) first executive director and then chief executive officer, T. Michael Elliott, passed away on 24 May 2016. Hired in 1982, Michael led the CS through an era of unprecedented growth and modernization. Described by past CS presidents Bruce Shriver (1991) and Duncan Lawrie (1992) as the "conscience" and "soul" of the CS, Michael was highly valued by members, volunteers, and staff alike for his mentorship and contributions to the CS's health and success.

"He was the heart of the Computer Society for many years," said Thomas W. Williams, former CS Board of Governors member and IEEE division director. "He was also a visionary … [the] digital library was his doing long before any other part of IEEE even had a thought of it. He dedicated the majority of his professional career to the Computer Society [and] … was a very compassionate, personable man who took care of his employees, helped them grow, and took care of the volunteers and helped them grow as well. We will miss him."

Throughout Michael's 18 years of leadership, the CS achieved many firsts within IEEE, including establishing offices in Europe and Asia. Membership also grew by more than 50 percent to over 100,000.

"[Michael] recognized the need for changes in the business model for organizations to provide and share information," said CS past president Ron Hoelzeman (1995). "He introduced a standalone email system for Society members well before it became established practice. … He codified the planning process and provided seminars on planning for both volunteers and staff. … He recognized [non-US members] as … an important source of new members [as well as] an opportunity for the Society to reach those who were underserved with up-to-date information."

When True Seaborn, the CS's long-time publisher, retired in 1996, he reflected on Michael's commitment to innovation: "I credit him for his leadership in establishing a Society-wide culture that continually stays close to the technical envelope." At a time when computer-aided publishing was in its infancy, the CS took the lead in acquiring "essentially all feature content electronically, handling editing and layout on-screen, and then transmitting our files via FTP to the printer."

Under Michael's guidance, the CS's periodical publishing program expanded from 6 to 21 titles, sponsored conferences grew from 40 to 150, and book production increased from fewer than 50 to more than 170 volumes per year. CS past president Helen Wood (1990) remembers that "Michael was a consummate professional with a passion for excellence in his every endeavor."

"Michael continued to play a pivotal role in nearly every critical undertaking, from strategic planning to the most mundane matters, whether they involved our staff or volunteers or membership," recalled CS past president Edward A. Parrish (1988). "Thanks to his tireless efforts, the Society blossomed and became the largest and most successful of all the groups within IEEE. I considered him not only a colleague, but a personal friend and will sorely miss him."

Michael received an EdD in higher education from Indiana University Bloomington. He served as special assistant to the president and later became assistant provost at Purdue University, executive director of the National Commission on United Methodist Higher Education, deputy commissioner for the Missouri Department of Higher Education, and director of the Arkansas State Board of Higher Education—serving in the cabinets of former governors Bill Clinton and Frank White.

Michael was honored with the CS's Meritorious Service Award as well as the Harry Hayman Award for Distinguished Staff Achievement, the highest staff honor.

Michael's family asks that those wishing to remember him with a donation make a gift in his name to the American Kidney Fund (www.kidneyfund.org).

My best counsel to my friends and colleagues who remain is to remember the words of Gandhi: "You must be the change you wish to see." *—Michael's parting advice on retiring in 2000*

# CALL AND CALENDAR

## CALLS FOR ARTICLES FOR *COMPUTER*

*Computer* plans a June 2017 special issue on VLSI for the Internet of Things (IoT).

The IoT promises to revolutionize a wide range of applications. But capitalizing on IoT will require a new generation of VLSI systems. IoT sensing and actuation devices must be extremely low cost while still delivering a complete system on chip (SoC)—capable of performing computation, security, and wireless communications—at extremely low power levels. Other types of edge devices can help close the gap in capabilities, cost, and energy usage between the traditional TCP/IP-based cloud and sensing/actuation nodes.

This new generation of IoT devices requires a very different approach to VLSI design at all levels of abstraction: technology, circuits, logic, architecture, and systems. While the microprocessor market is driven by complex functionality, resulting in large chips that reach clocking and thermal limits, IoT systems require low-cost, low-power SoCs.

Abstracts are due **1 October 2016** (mail to **co-0617@computer.org**), and articles are due **1 November 2016**. Visit www.computer.org/computer/cfp6 to view the complete call for papers.

## CALLS FOR ARTICLES FOR OTHER IEEE CS PUBLICATIONS

*IEEE Internet Computing* plans a July/August 2017 special issue on energy-efficient datacenters.

In the last decade, datacenters have become the core of modern business environments as computation has moved to the cloud. Datacenters are among the US's fastest-growing electricity users. This costs businesses money, consumes limited energy resources, and contributes to environmental pollution.

Because of this, there's a demand for more energy-efficient datacenters. This special issue calls for research on various approaches that can enable their creation.

Brief article descriptions are due **28 September 2016** (mail to **ic4-2017@computer.org**), and articles are due **28 October 2016**. Visit www.computer.org/web/computingnow/iccfp4 to view the complete call for papers.

*IEEE Transactions on Nanotechnology* and *IEEE Transactions on Emerging Topics in Computing* plan a September 2017 joint special section on VLSI and nanotechnology design trends for computing innovations.

Topics of interest include VLSI design; low-power and power-aware design; testing, reliability, and fault tolerance; VLSI circuits; computer-aided design; emerging technologies; and post-CMOS VLSI.

Articles are due **30 September 2016**. Visit www.computer.org/cms/Computer.org/transactions/cfps/cfp_tetcsi_vlsi_ndtci.pdf to view the complete call for papers.

*IT Professional* plans a May/June 2017 special issue on mobile data analytics.

Mobile data analytics (MDA) is increasingly important to IT professionals, entrepreneurs, and academics. MDA deals largely with big data analytics on resource-constrained mobile devices. This has become increasingly important with the proliferation of mobile commerce, mobile advertising, and data collection by mobile app vendors.

However, MDA faces challenges as it demands heavy processing and lots of memory and storage, normally unavailable in mobile devices.

This special issue seeks to present readers with MDA trends, issues, novel solutions, and applications. We are soliciting articles from industry, business, academia, and government.

Articles are due **1 October 2016**. Visit www.computer.org/itpro/cfp4 to view the complete call for papers.

*IEEE Computer Graphics and Applications* plans a July/August 2017 special issue on animation of natural virtual characters.

Virtual characters are used in a range of applications—from interfaces to games—in which they must effectively employ human nonverbal communication.

## SUBMISSION INSTRUCTIONS

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting.

Visit www.computer.org/conferences for instructions on how to submit conference or call listings as well as a more complete listing of upcoming computing-related conferences.

0018-9162/16/$33.00 © 2016 IEEE

Building systems that can support these interactions involves both the automatic specification and generation of appropriate character motion, as well as coordination across communication modes and among multiple characters.

For this special issue, we solicit papers describing innovative character-animation techniques and models.

Articles are due **1 November 2016**. Visit www.computer.org/web/computingnow/cgacfp4 to view the complete call for papers.

*Computing in Science & Engineering* plans a July/August 2017 special issue on computational advances in solar system studies.

In recent decades, advances in computing power, algorithm development, and data management and analysis have greatly expanded our knowledge of and ability to explore the solar system. At the dawn of the space age, no one anticipated the enormous impact that computers and computational science would have.

In this special issue, we invite articles about the computational tools that have produced surprising results as we explore our solar system. We encourage submissions from the computer science, computer engineering, and planetary sciences communities.

Articles are due **1 November 2016**. Visit www.computer.org/web/computingnow/cscfp4 to view the complete call for papers.

*IEEE Security & Privacy* plans a September/October 2017 special issue on genome privacy and security.

Over the past several decades, genome-sequencing technologies have evolved from slow, expensive systems accessible only to a few scientists and forensic investigators to high-throughput, relatively low-cost tools available to consumers. A consequence is that genomics has become one of the next major privacy and security challenges.

As genomics becomes increasingly integrated into healthcare and direct-to-consumer services such as ancestry testing, DNA data leakage is a serious risk for both individuals and their relatives.

This special issue will present the views of computer scientists, medical doctors, legal scholars, and ethicists on this important topic.

Abstracts are due **1 November 2016**. Articles are due **1 January 2017**. Visit www.computer.org/web/computingnow/spcfp5 to view the complete call for papers.

## SEEKING PAPERS ON COMPUTATIONAL SOCIAL SYSTEMS

*IEEE Transactions on Computational Social Systems* welcomes submissions on topics such as modeling, simulation, analysis, and the understanding of social systems from the quantitative and/or computational perspective. Learn more at www.ieeesmc.org/publications/transactions-on-computational-social-systems/call-for-papers-and-special-issues.

## SEEKING PAPERS ON SOFTWARE ENGINEERING

*Computing in Science & Engineering* seeks submissions on scientific-software engineering. The magazine seeks to provide a venue for the publication of significant work in the field, recognizing that the development of scientific software differs significantly from that of other software. Learn more at www.computer.org/cms/Computer.org/ComputingNow/docs/2016-software-engineering-track.pdf.

*IEEE Software* plans a July/August 2017 special issue on reliability engineering for software.

As the number of smart, interconnected devices in our cars and homes grows, engineers must increasingly consider software reliability in the connectivity tools and platforms they create. Reliability engineering emphasizes dependability, whether at a critical moment or throughout the software's life cycle.

This theme issue focuses on reliability challenges and successes in software engineering.

The guest editors seek articles reporting case studies, experience reports, practices, approaches, techniques, and guidelines, all involving practical software results.

Articles are due **1 December 2016**. Visit www.computer.org/software/cfp4 to view the complete call for papers.

*IT Professional* plans a July/August 2017 special issue on cognitive computing.

## EVENTS IN 2016 AND 2017

### OCTOBER 2016

| | |
|---|---|
| 3–5 | ICCD 2016 |
| 4–7 | ICHI 2016 |
| 17–19 | WiMob 2016 |
| 23–27 | eScience 2016 |
| 31 October–3 November | CIC 2016 |

### NOVEMBER 2016

| | |
|---|---|
| 4–6 | ICEBE 2016 |
| 13–18 | SC 2016 |
| 29 November–2 December | AICCSA 2016 |

### DECEMBER 2016

| | |
|---|---|
| 2–4 | T4E 2016 |
| 8–10 | CIT 2016 |
| 12–14 | WF-IOT 2016 |
| 19–22 | HiPC 2016 |

### JANUARY 2017

| | |
|---|---|
| 22–25 | PRDC 2017 |
| 30 January–1 February | ICSC 2017 |

Cognitive computing refers to smart systems that learn at scale, reason with purpose, and interact with humans and other smart systems. Rather than being explicitly programmed, such systems learn and reason from their interactions with us and from experiences with their environment.

This special issue seeks to provide readers with an overview of current cognitive-computing issues and practices, as well as a look into the future.

Articles are due **1 December 2016**. Visit www.computer.org/itpro/cfp4 to view the complete call for papers.

*IEEE Transactions on Emerging Topics in Computing* (TETC) plans the following special section for its December 2017 issue, with articles due **1 December 2016**:

Emerging topics for disaster management: Visit www.computer.org/cms/Computer.org/transactions/cfps/cfp_tetcsi_etdm.pdf for the complete call for papers.

Cyber social computing and cyber-enabled applications: Visit www.computer.org/cms/Computer.org/transactions/cfps/cfp_tetcsi_csccea.pdf for the complete call for papers.

*IEEE Transactions on Emerging Topics in Computing* (TETC) and *IEEE Transactions on Learning Technologies* (TLT) plan a joint special section for their October–December 2017 issues, with articles due **1 December 2016**:

Innovation in technologies for educational computing: Visit www.computer.org/cms/Computer.org/transactions/cfps/cfp_tetcsi_itec.pdf for the complete call for papers.

*IEEE Internet Computing* plans a September/October 2017 special issue on 5G.

The past two decades have witnessed phenomenal progress in wireless access to the Internet and telecommunication services.

Cellular network operators in several countries have deployed fourth-generation long-term evolution (4G LTE) radio technologies to keep up with the demand. Fifth generation—5G—cellular network technologies are slated for 2020.

This special issue will provide a comprehensive update of 5G radio and network technologies. The guest editors seek papers from industry and academia. Of particular interest are review and tutorial articles that summarize the subject area, and provide guidance to researchers and planners so that they can anticipate 5G's impact on networks and envision new mobile services.

Brief article descriptions are due **12 December 2016** (mail to **ic5-2017@computer.org**), and articles are due **12 January 2017**. Visit www.computer.org/web/computingnow/iccfp5 to view the complete call for papers.

*IEEE Security & Privacy* plans a November/December 2017 special issue on digital forensics.

Modern societies are becoming increasingly dependent on open networks over which commercial activities, business transactions, and government services are delivered. However, these networks can attract cybercriminals.

In these cases, cybercrime detection and evidence collection can be difficult because clues are often buried in large data volumes. In addition, investigations of cybercriminal activities often span international borders and are subject to multiple jurisdictions and legal systems. These challenges require the use of digital forensics, which is becoming increasingly important.

This special issue aims to collect information on the most relevant digital-forensics research.

Articles are due **1 March 2017**. Visit www.computer.org/web/computingnow/spcfp6 to view the complete call for papers.

*IEEE Transactions on Emerging Topics in Computing* (TETC) plans the following special section for its March 2018 issue, with articles due **1 March 2017**:

Reliability-aware design and analysis methods for digital systems, from gate to system level: Visit www.computer.org/cms/Computer.org/transactions/cfps/cfp_tetcsi_rdamds.pdf to view the complete call for papers.

*IEEE Transactions on Emerging Topics in Computing* (TETC) plans the following special section for its June 2018 issue, with articles due **1 June 2017**:

Cybersecurity threats and defense advances: Visit www.computer.org/cms/Computer.org/transactions/cfps/cfp_tetcsi_cstda.pdf to view the complete call for papers.

## ICSC 2017

The 11th IEEE International Conference on Semantic Computing (ICSC) is sponsored by the IEEE Computer Society and IEEE.

Semantic computing is used in areas such as analytics, interfaces, the Internet of Things, cloud computing, software-defined networking, wearable computing, context awareness, mobile computing, search engines, big data, and multimedia.

The technical program will address important semantic-computing-related topics such as natural deep learning, search engines, robotics, Web services, security, privacy, and applications in areas such as healthcare, manufacturing, education, finance, business, biomedicine, and various other scientific fields.

The conference will take place 30 January–1 February 2017 in San Diego. Visit http://icsc.eecs.uci.edu/2017/index.html for complete conference information.

### OCTOBER 2016

**3–5 October: ICCD 2016, 34th IEEE Int'l Conf. Computer Design**, Phoenix; www.iccd-conf.com/Home.html

**4–7 October: ICHI 2016, IEEE Int'l Conf. Healthcare Informatics**, Chicago; www.ieee-ichi.org

**17–19 October: WiMob 2016, 12th IEEE Int'l Conf. Wireless and Mobile Computing, Networking and Communications**, New York; http://conferences.computer.org/wimob2016

**23–27 October: eScience 2016, 12th IEEE Int'l Conf. eScience**, Baltimore; http://escience-2016.idies.jhu.edu

**31 October–3 November: CIC 2016, 2nd IEEE Int'l Conf. Collaboration and Internet Computing**, Pittsburgh; www.sis.pitt.edu/cic2016/index.html

### NOVEMBER 2016

**4–6 November: ICEBE 2016, 13th IEEE Int'l Conf. e-Business Eng.**, Macau, China; http://conferences.computer.org/icebe/2016/index.htm

**13–18 November: SC 2016, Int'l Conf. High-Performance Computing, Networking, Storage, and Analysis,** Salt Lake City; http://sc16.supercomputing.org

**29 November–2 December: AICCSA 2016, 13th IEEE/ACS Int'l Conf. Computer Systems and Applications**, Agadir, Morocco; www.aiccsa.net/AICCSA2016/home

### DECEMBER 2016

**2–4 December: T4E 2016, 8th IEEE Int'l Conf. Tech. for Education**, Mumbai, India; www.ask4research.info/t4e/2016

**8–10 December: CIT 2016, 16th IEEE Int'l Conf. Computer and Info. Tech.**, Nadi, Fiji; http://nsclab.org/cit2016

**12–14 December: WF-IOT 2016, IEEE 3rd World Forum on Internet of Things**, Reston, Virginia; http://wfiot2016.ieee-wf-iot.org

**19–22 December: HiPC 2016, 23rd IEEE Int'l Conf. High Performance Computing, Data, and Analytics**, Hyderabad, India; www.hipc.org/hipc2016/index.php

### JANUARY 2017

**22–25 January: PRDC 2017, 22nd IEEE Pacific Rim Int'l Symp. Dependable Computing**, Christchurch, New Zealand; http://prdc.dependability.org/PRDC2017

**30 January–1 February: ICSC 2017, 11th IEEE Int'l Conf. Semantic Computing**, San Diego; http://icsc.eecs.uci.edu/2017/index.html

Selected CS articles and columns are also available for free at **http://ComputingNow.computer.org.**

# CAREER OPPORTUNITIES

**PROGRAMMER ANALYST** - dsgn, dvlp, maintain, test & implmt applic s/w d/bases utilizing knowl of MS BizTalk Server, C#, C.HTTP, WCF, Oracle, SOAP, SQL Server and MS Visual Studio; Reqs MS Comp Sci, Engg or rel. Mail resumes to Strategic Resources International, Inc, 777 Washington Rd, Ste 2, Parlin , NJ 08859.

**SYSTEMS ANALYST III.** Design and support large complex campus local area networks (LAN) and wide area networks (WAN) solutions utilizing Cisco Nexus Switches, firewalls, load balancers, and Infoblox DNS devices. Participate in complex network transformations and provide solutions. Resolve routing issues using the Enhanced Interior Gateway Routing Protocol (EIGRP) and Border Gateway Protocol (BGP). Apply to: Gerald O'Mara, #82115, AHS Hospital Corp, 100 Madison Avenue, Morristown, NJ 07960.

**CLOUDERA, INC.** is recruiting for our Austin, TX office: Software Engineer: maintain a thorough understanding of Cloudera security components related to key mgmt.. &amp; file system encryption. Mail resume w/job code #36229 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

**CLOUDERA, INC.** is recruiting for our San Francisco, CA office: Sr. Product Mgr.: work closely with engineering to define technical product roadmaps for specific products &amp; features in Cloudera product suite. Mail resume w/job code #36921 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

**TEXAS STATE UNIVERSITY.** Department of Computer Science. Applications are invited for multiple tenure-track Assistant Professor positions in the Department of Computer Science to start the fall 2017 semester. Consult the department's faculty employment page at www.cs.txstate.edu/employment/faculty/ for job duties, qualifications, application procedure, and information about the department and the university. Texas State University, to the extent not in conflict with federal or state law, prohibits discrimination or harassment on the basis of race, color, national origin, age, sex, religion, disability, veteran's status, sexual orientation, gender identity or expression. Texas State University is a member of The Texas State University System. Texas State University is an EOE.

**DIRECTORS,** Continuous Improvement, Jacksonville, FL: Develop annual strategic plans for four business units. Use Hoshin Kanri methodology to deploy strategic plans. Utilize Six Sigma Methodologies to improve product quality. Plan and deploy training for all four business units in Green Belt, Strategy Deployment and Value Stream Mapping. Send res to Xorail, Inc. 5011 Gate Pkwy, Bldg 100, Ste 400, Jacksonville, FL 32256

**SYS. ANALYST** open'g @ G4S Tech. SW Solutns., LLC, Burlington, MA. Do data structure + algorithm + stored procedure creatn; code + bus. req't analysis; code refactor'g; database develpmt; script'g + test'g. Need exper. w/C#, XML, HTML, CSS, SQL Serv. + SharePoint. Must live in Grtr. Boston/Lowell/Worcester, Central MA, MA MetroWest, MA No./So. Shore, or So. NH. Cannot telecommute. Resumes to employer @ amy.dechant@usa.g4s.com. Ref. Job SR in subj. ln.

## CALIFORNIA STATE UNIVERSITY, EAST BAY (Hayward, CA)

Department of Computer Science
**Faculty Position in Computer Science**

2 POSITIONS (OAA Position No. 16-17 COMPUTER SCIENCE-TT) The Department invites applications for 2 tenure-track appointments as **Assistant Professor in Computer Science**(considering all areas of computer science, capable of teaching in emerging areas) starting Fall 2017.
For details see http://www20.csueastbay.edu/about/career-opportunities/.
For questions, email: CSSearch@mcs.CSUEastBay.edu.

SOFTWARE

## Oracle America, Inc.

has openings for

## SOFTWARE DEVELOPMENT MANAGER

positions in **Westminster, CO**.

Job duties include: Manage team that designs, develops, troubleshoots and debugs software programs for databases, applications, tools, networks, etc.

Apply by e-mailing resume to jeremy.meade@oracle.com, referencing 385.20312.

Oracle supports workforce diversity.

TECHNICAL

## Oracle America, Inc.

has openings for

## TECHNICAL ANALYST

positions in **Lehi, UT**.

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure.

Apply by e-mailing resume to paul.merideth@oracle.com, referencing 385.19519.

Oracle supports workforce diversity.

**WWW.COMPUTER.ORG /COMPUTER**

## ENGINEERING/TECHNOLOGY

**CyOptics, Inc.**, an optical networking developer, has openings in Breinigsville, PA for R&D Engineer Fab Process (FPE03): Perform process development and manufacturing support for the following processes: Reactive-Ion Etch (RIE) Grating Etch; Grating Holography; AFM (Atomic Force Microscopy) and Period Measurement; and Oxide Etch.; R&D Engineer Hardware (HW03): Provide high speed flexible printed circuit (FPCB) and printed circuit board (PCB) design and layout. Position may require 10% domestic and/or international travel.; R&D Engineer Hardware (HW04): Provide high speed TO-can design and process development. Position may requires 15% domestic and/or international travel.; Optoelectronics Design Engineer (ODE01): Design high speed InP optoelectronic devices, including lasers, modulators and detectors. Ref job code and mail resume to CyOptics, Attn: HR (GS), 1320 Ridder Park Dr., San Jose, CA 95131.

JUNIPER NETWORKS

**Juniper Networks is recruiting for our Sunnyvale, CA office:**

**Functional Systems Analyst Staff #37662**: Provide support to the MTS business function. Work with Blueprint & Realization-related activities, such as design, configuration, customization, implementation and production support.

**Software Staff Engineer #6362**: Design, modify and execute test cases on Company security platforms to ensure quality.

**Software Engineer #30992**: Analyze, design, write, debug and modify programming code, either from scratch or by using existing open-source libraries, to develop the web-UI component of Software Defined Networking (SDN) product.

**Mail single-sided resume with**
**job code # to**
**Juniper Networks**
**Attn**: MS A.8.429A
**1133 Innovation Way**
**Sunnyvale, CA 94089**

IEEE computer society IEEE

## 2017 B. Ramakrishna Rau Award
## Call for Nominations

*Honoring contributions to the computer microarchitecture field*

**New Deadline: 1 May 2017**

Established in memory of Dr. B. (Bob) Ramakrishna Rau, the award recognizes his distinguished career in promoting and expanding the use of innovative computer microarchitecture techniques, including his innovation in complier technology, his leadership in academic and industrial computer architecture, and his extremely high personal and ethical standards.

**WHO IS ELIGIBLE?** The candidate will have made an outstanding innovative contribution or contributions to microarchitecture, use of novel microarchitectural techniques or compiler/architecture interfacing. It is hoped, but not required, that the winner will have also contributed to the computer microarchitecture community through teaching, mentoring, or community service.

**AWARD:** Certificate and a $2,000 honorarium.

**PRESENTATION:** Annually presented at the ACM/IEEE International Symposium on Microarchitecture

**NOMINATION SUBMISSION:** This award requires 3 endorsements. Nominations are being accepted electronically: www.computer.org/web/awards/rau

**CONTACT US:** Send any award-related questions to awards@computer.org

**www.computer.org/awards**

# San Diego State University

## Department of Computer Science
## Chair of Computer Science

Department of Computer Science at SDSU seeks candidates for the Chair position with a PhD in Computer Science or a related discipline, and a sustained record of supported research. The Department is a dynamic and growing unit looking for a visionary Chair to lead its future development.

We strive to build and sustain a welcoming environment for all. SDSU is seeking applicants with commitment to working effectively with individuals from diverse backgrounds and members of underrepresented groups.

For more details and application procedures, please apply via https://apply.interfolio.com/36248.

SDSU is a Title IX, equal opportunity employer. A full version of this ad can be found at: http://cs.sdsu.edu/

## CAREER OPPORTUNITIES

**SPLUNK INC.** has the following job opportunities in **San Francisco, CA: Software Engineer (Req#A3GSEN)** Design & dev sw for large-scale machine data search engine. **Security Applications Engineer (Req#9V4RJ9)** Create, design & maintain ent-grade co apps that leverage Big Data. **Human Factors Engineer (Req#9VESMH)** Drive the process of user-ctr'd design & lean UX for user research throughout prod dev process. **Splunk Inc.** has the following job opportunities in **Cupertino, CA: Software Engineer (Req#9W9T44)** Design & dev scalable, performant sw that functions across multiple operating sys and envir. **Senior Software Engineer (Req#A3FQG5)** Design & dev scalable, performant sw that functions across multiple operating sys and envir. Refer to Req# & mail resume to Splunk Inc., ATTN: J. Aldax, 250 Brannan Street, San Francisco CA 94107. Individuals seeking employment at Splunk are considered without regards to race, religion, color, national origin, ancestry, sex, gender, gender identity, gender expression, sexual orientation, marital status, age, physical or mental disability or medical condition (except where physical fitness is a valid occupational qualification), genetic information, veteran status, or any other consideration made unlawful by federal, state or local laws. To review US DOL's EEO is The Law notice please visit: https://careers.jobvite.com/Splunk/EEO_poster.pdf. To review Splunk's EEO Policy Statement please visit: http://careers.jobvite.com/Careers/Splunk/EEO-Policy-Statement.pdf. Pursuant to the San Francisco Fair Chance Ordinance, we will consider for employment qualified applicants with arrest and conviction records.

**SR. SOFTWARE ENGINEERS.** Des & dvlp self-checkout & POS applns using techs like Java/J2ee, C, C++, w/4690. Reqs BS in CS, CE or rltd field,& 5 yrs rltd exp.
**SR. SAP CONSULTANTS.** Sys study, gap analysis, mapping, dvlp/configure SD modules, perform UAT. Reqs BS in Bus Admin, IT or rltd field, & 5 yrs rltd exp. Exp in SAP R/3, 6.0 ECC Version on SD modules & test tool HPQC QTP.

Educ may be equiv/frgn equiv. Comb of degrees accepted. Mltpl opngs in Annapolis, MD & other client sites. No travel. May reqr reloc. Apply:Vedasoft, Inc, 801 Compass Way, Suite 218, Annapolis, MD 21401.EOE.

**FIELD TECHNICAL ELECTRONICS AND COMMUNICATIONS SERVICE ENGINEER,** Long Beach, Los Angeles County, CA – Management of field technicians, inspect project sites, and direct & coordinate maintenance, support, documentation and testing activities to ensure compliance with specification codes and customer requirements. Must have B.S. + 2 yrs exp. Send CV to WM Wireless, Inc. Attn: Rose Sajo – 6723 N. Paramount Blvd., Long Beach, CA 90805.

**SENIOR JAVA DEVELOPER-APPLICATIONS:** Analyze software requirements to determine feasibility. Design, develop and modify existing software to correct errors, adapt to new hardware or improve performance using scientific analysis and mathematical models. Develop and direct software system testing and validation procedures. Coordinate software

**TECHNOLOGY** Help build the next generation of systems behind Facebook's products.

# Facebook, Inc.

currently has the following openings in **Menlo Park, CA (multiple openings/various levels)**:

**Systems Engineer (3396J)** Integration of new hardware products into Facebook software & datacenter infrastructure. **UX Research Manager (3432J)** Be an expert user experience researcher with a proven track record of doing research that impacts a complex & diverse product. Position requires occasional travel to unanticipated locations. **Data Engineer (DE616J)** Design & build data reporting & visualization needs for a product or a group of products. **Systems Developer (1308J)** Build, test, debug, & make code changes to tools & systems. **UX Researcher (5085J)** Responsible for the design of studies that address both user behavior & attitudes. **Developer Operations Manager (5732J)** Serve as the voice of the user, working directly with product, engineering, global policy & other partners responsible for solving these challenges, playing an integral role in improving the user experience on Facebook. **Product Manager (411J)** Plan business objectives, develop product strategies & establish responsibilities across product area. **Software Engineer (5828J)** Help build the next generation of systems behind Facebook's products, create web &/or mobile applications that reach over one billion people, & build high volume servers to support our content. **Partner Engineer (2572J)** Handle technical integrations with partners to optimize Facebook user experience. Position requires international business travel to unanticipated sites. **Production Engineer (PE616J)** Participate in the design, implementation & ongoing management of major site applications & subsystems. **Data Center Construction Cost Estimator (6874J)** Manage cost estimating efforts for new & retrofit Data Center projects. Prepare conceptual cost estimates with planning, engineering, sourcing, energy & site selection team during early concept stage. Position requires 20% national & international travel to unanticipated worksites. **Product Designer (7345J)** Design, prototype, & build new features for Facebook's website or mobile applications. **UX Researcher (6450J)** Oversee & design the user experience component to generate actionable insights.

Mail resume to: Facebook, Inc. Attn: SB-GIM, 1 Hacker Way, Menlo Park, CA 94025. Must reference job title & job# shown above, when applying.

installation. Must have min. of Bachelor's Deg. in Computer Sciences or related + 7 years of exp. in Java development., including Javascript, AJAX, HTML, CSS & XML. Must pass a coding/computer programming test. Mail resume to: MotionPoint, H.R. Director, 4661 Johnson Rd., Suite 14, Coconut Creek, FL 33073.

**DATA WAREHOUSE AND ANALYTICS DEVELOPER** (ETL/Informatica). Ascension Health-IS, Inc. is seeking a Data Warehouse and Analytics Developer (ETL/Informatica) in St. Louis, Missouri to code design and development on the data warehouse/analytics Extract Transform Load (ETL) toolset, Informatica PowerCenter; support Informatica toolset; participate in testing (e.g. user acceptance testing, unit, system, regression, integration testing); troubleshoot applications and datasets; monitor and maintain installed systems. Contact Jenna Mihm, Vice President Legal Services & Associate General Counsel, Ascension Health, 4600 Edmundson Road, St. Louis, MO 63134, 314-733-8692, Jenna.Mihm@ascensionhealth.org To apply for this position, please reference Job Number 09.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

# PROGRAMMER ANALYST

positions in **Austin, TX.**

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure.

Apply by e-mailing resume to pulkit.khurana@oracle.com, referencing 385.18935.

Oracle supports workforce diversity.

**Microsoft®**

Microsoft Corporation currently has the following openings (job opportunities available at all levels, including Principal, Senior and Lead levels):

### REDMOND, WA

**Research Software Development Engineers (all levels):** Responsible for conducting applied research into new products and services through software engineering techniques. http://bit.ly/MSJobs_Research_Software_Engineer

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. Requires domestic and international travel up to 25%. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### MOUNTAIN VIEW, PALO ALTO, SUNNYVALE, CA

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. Requires domestic and international travel up to 25%. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### ALISO VIEJO, CA

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### SAN FRANCISCO, CA

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### CAMBRIDGE, MA

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. Requires domestic and international travel up to 25%. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### DURHAM, NC

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### FARGO, ND

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### NEW YORK, NY

**Research Software Development Engineers (all levels):** Responsible for conducting applied research into new products and services through software engineering techniques. http://bit.ly/MSJobs_Research_Software_Engineer

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

### RESTON, VA

**Software Engineers and Software Development Engineers in Test (all levels, including Leads and Managers):** Responsible for developing or testing computer software applications, systems or services. (http://bit.ly/MSJobs_SDE) (http://bit.ly/MSJobs_IT_SDE)

**Multiple job openings are available for each of these categories. To view detailed job descriptions and minimum requirements, and to apply, visit the website address listed. EOE.**

# CAREER OPPORTUNITIES

**CLOUDERA, INC.** is recruiting for our Palo Alto, CA office: Software Engineer: Design & implement engineering systems that scale well – to petabytes of data & 1000s of nodes. Mail resume w/job code #37924 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

**MANAGER.** Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Resp. for advising on & implementing strategic solutions for finance & Enterprise Perform. Mgmt. (EPM) processes incl. planning, budgeting, forecasting, reporting, fin. consolidation, shared services, global bus. services & bus. process outsourcing. Advise clients on best practices & help scope, plan & drive day-to-day execution of client engagements related to finance/EPM incl. strategy: blueprint, org. design, people develop., strategy / KPI articulation; transform.: process & data optimization, service delivery model redesign, cost mgmt..; & Tech: EPM/BI info archi., planning & budgeting solutions, reporting & analytics solutions. Leverage func. data modeling tools incl. Excel & Tableau & tech. content knowl. of Hyperion (DRM, ODI, OBIEE, OBIA, FDM, Fin.. Mgmt., Planning, Essbase) & other tech. solutions to achieve success. Requires: M.S. in Finance, Eng. or related field & 2 yrs. exp. in job offered or 2 yrs. exp. as a Consult. or Fin. Analyst. Concurrent exp. must incl.: 2 yrs. exp. using Hyperion solutions incl. DRM, ODI, OBIEE, Planning & Essbase & 2 yrs. exp. with data modeling tools incl. Excel & Tableau. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131.

**APPLICATIONS** for two full-time, academic year, tenure-track Software Engineering faculty positions, at a rank and salary commensurate with the applicant's background and experience are being accepted. For full details, qualifications and application instructions (online faculty application required), visit WWW.CALPOLYJOBS.ORG and refer to Requisition #104135. Review will begin January 2, 2017 and will continue until the positions are filled. EEO.

**APPLICATIONS** for a full-time, academic year, tenure-track Computer Science faculty appointment in the area of computer graphics and animation, at a rank and salary commensurate with the applicant's background and experience are being accepted. Particular areas of interest include: computer graphics, interactive media, animation, and data visualization. Anticipated start date 9/7/17. For full details and to apply visit:http://www.calpolyjobs.org/applicants/Central?quickFind=165455.

**SOFTWARE DEVELOPER AND SENIOR SOFTWARE DEVELOPER.** Valassis Communications Inc. has openings for the positions of Software Developer and Senior Software Developer in Windsor, CT to analyze, write, test, & deploy software applications. To apply mail resume to Valassis Communications Inc., Attn: Patty [16-CT-11307.3] 19975 Victor Parkway, Livonia, MI 48152.

**ENGINEERING. ERICSSON, INC.** has openings for the following positions: **TECHNICAL SUPPORT ENGINEER _**

INFORMATION TECHNOLOGY

**Oracle America, Inc.**

has openings for

**IT SENIOR MANAGER**

positions in **Redwood Shores, CA**.

Job duties include: Manage development, design and optimization of automated testing initiatives and processes.

Apply by e-mailing resume to steve.dadds@oracle.com, referencing 385.14346.

Oracle supports workforce diversity.

TECHNOLOGY

**Oracle America, Inc.**

has openings for

**APPLICATIONS DEVELOPER**

positions in **Irvine, CA**.

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications. Write code, complete programming and perform testing and debugging of applications.

Apply by e-mailing resume to huyvu.nguyen@oracle.com, referencing 385.17072.

Oracle supports workforce diversity.

TECHNOLOGY

**Oracle America, Inc.**

has openings for

**DIRECTOR OF PRODUCT MANAGEMENT**

positions in **Redwood Shores, CA**.

Job duties include: Participate in all software and/or hardware product development life cycle activities. Move software products through the software product development cycle from design and development to implementation, testing and/or marketing. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to navin.thadani@oracle.com, referencing 385.20854.

Oracle supports workforce diversity.

**in ATLANTA, GA** to perform day-to-day telecom order management; analyze production issues; & provide resolution of database, syst, or network issues. **JOB ID: 16-GA-3789. PRODUCT MARKETING ENGINEER _ in PLANO, TX** to develop technical solutions in strategic growth areas; present & implement solutions with customers; define & drive network & IT transformation growth strategies. Up to 10% domestic/int'l travel required. JOB ID: 16-TX-3725. **To apply mail resume to Ericsson Inc.** 6300 Legacy Dr, R1-C12 Plano, TX 75024 indicating appropriate Job ID.

**SOFTWARE ENGINEER – ATG.** Harland Clarke Corp. has an opening for the position of Software Engineer – ATG in San Antonio, TX to contribute to all steps within the systems development life cycle including functional/non-functional requirements gathering/validation, analysis, design, construction, testing, & implementation. To apply mail resume to Harland Clarke Corp, Attn: Diana [16-TX6174.66], 15955 La Cantera Parkway, San Antonio, TX 78256.

**IT PROFESSIONALS:** Established IT company has multiple openings for the following positions: Software Engineer (Lead), Database Administrator, & Programmer Analyst (BS deg. or the equiv. in CS, Engg. (any) or related & 24 mo. of relevant industry exp. Will consider candidates w/combination of education &/or work exp. considered to be equiv. to a bachelor's deg. in CS, Engg. (any) or related & 24 months' of relevant industry exp). Solutions Architect (BS deg. In CS, Engg. (any), BUS or related & 24 mo. of relevant industry exp. Will consider candidates w/combination of education &/or work exp. considered to be equiv. to a bachelor's deg. in CS, Engg. (any), BUS or related & 24 months' of relevant industry exp). Quality Assurance Analyst (BS deg. or equiv. in CS, Engg. (any) or related & 24 mo. of relevant industry exp. Will consider candidates w/ combination of education &/or work exp. considered to be equiv. to a bachelor's deg. in CS, Engg. (any) or related & 24 months' of relevant industry exp). Lead Software Developers & IT Project Manager (MS deg. or equiv. in CS, Engg. or related & 12 mo. of relevant industry exp. We will consider applicants with a relevant bachelor's deg. & at least 5 yrs. of relevant industry exp. for these positions). Sr. Solutions Architect (MS deg. or equiv. in CS, Engg., BUS or related & 12 mo. of relevant industry exp. We will consider applicants with a relevant bachelor's deg. & at least 5 yrs. of relevant industry exp. for these positions). Business Consultant (IT) (Master's deg. Or equiv. in Bus. Admin. or related & 12 mos. of relevant industry exp. We will consider applicants with a bachelor's deg. or equiv. in Bus. Admin. or related & at least 5 yrs. of relevant industry exp. for these positions). Positions based out of company's US headquarters in Morristown, NJ & subject to relocation to various office & client sites throughout the US. Send resumes to Karen Fernandes, HR Dept., Majesco, 412 Mt. Kemble Ave., Ste. 110C, Morristown, NJ 07960.

TECHNICAL

**Oracle America, Inc.**

has openings for

**CONSULTING TECHNICAL MANAGER**

positions in **Redwood Shores, CA**.

Job duties include: Analyze requirements and deliver functional and technical solutions. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to
Swarnali.bag@oracle.com,
referencing 385.19233.

Oracle supports workforce diversity.

SOFTWARE

**Oracle America, Inc.**

has openings for

**SOFTWARE DEVELOPER**

positions in **New York, NY**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to
ari.kermaier@oracle.com,
referencing 385.16476.

Oracle supports workforce diversity.

CONSULTANT

**Oracle America, Inc.**

has openings for

**CONSULTANT**

positions in **Frisco, TX**.

Job duties include: Analyze requirements and deliver functional and technical solutions. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to
zein.shivji@oracle.com,
referencing 385.16095.

Oracle supports workforce diversity.

# CAREER OPPORTUNITIES

Microsoft®

Microsoft Corporation currently has the following openings (job opportunities available at all levels, including Principal, Senior and Lead levels): Redmond, WA

**Applied Scientist:** Utilize knowledge in applied statistics and mathematics to handle large amounts of data using various tools. http://bit.ly/MSJobs_Data_Applied_Science

**Artists and Art Directors (all levels, including Leads and Managers):** Responsible for designing and creating art assets that meet or exceed industry standards for quality while supporting Microsoft Game Studio (MGS) business goals. http://bit.ly/MSJobs_Art

**Business Managers and Business Development Managers/Business Development and Strategy Analyst Manager:** Develop business opportunities for sales of software and services. http://bit.ly/MSJobs_Business_Development

**Business Process Manager:** Responsible for the design, implementation, and release of programs or projects. http://bit.ly/MSJobs_Fin_Plan_Analy_Contr

**Consultants:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Requires domestic and international travel up to 25%. http://bit.ly/MSJobs_Technical_Delivery

**Consultants:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Roving Employee—requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. http://bit.ly/MSJobs_Technical_Delivery

**Content Developer/Engineer (All levels, including Leads and Managers):** Responsible for the design, development, deployment, vision, and business strategy for content creation, acquisition, production, editorial, and publishing activities at Microsoft. http://bit.ly/MSJobs_Content_Publishing

**Data Scientist:** Manipulate large volumes of data, create new and improved techniques and/or solutions for data collection, management and usage. http://bit.ly/MSJobs_Data_Applied_Science

**Design Verification/Validation Engineers:** Responsible for ensuring the quality of Microsoft hardware products. http://bit.ly/MSJobs_Hardware_Design_Verification_Eng

**Designers:** Develop user interface and user interaction designs, prototypes and/or concepts for business productivity, entertainment or other software or hardware applications. http://bit.ly/MSJobs_Design

**Evangelists/Technical Evangelists:** Collaborate with sales teams to understand customer requirements, promote the sale of products, and provide sales support. http://bit.ly/MSJobs_Tech_Evangelist

**Game/Systems Designer:** Create design documents for multiple major features on large projects and the entire design on smaller projects, ensuring consistency of design. http://bit.ly/MSJobs_Game_Design

**Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers):** Design, implement and test computer hardware that add strategic value to the company. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

**Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers):** Design, implement and test computer hardware. Requires Domestic and International travel up to 25%. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

**Machine Learning Scientist:** Design and deliver general and/or domain-specific machine learning algorithms and systems. http://bit.ly/MSJobs_Data_Applied_Science

**Premier Field Engineers:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Roving Employee—requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. http://bit.ly/MSJobs_Support_Delivery

**Researchers/Scientists:** Conduct research and lead research collaborations that yield new insights, theories, analyses, data, algorithms, and prototypes and that advance state-of-the-art of computer science and engineering, as well as general scientific knowledge. http://bit.ly/MSJobs_Research

**Service Engineers/Managers, Service Operations Engineers, and Systems/Operations Engineers/Site Reliability Engineer:** Research, design, develop, and test operating systems-level software, compilers, and network distribution software. (http://bit.ly/MSJobs_Service_Engineering) (http://bit.ly/MSJobs_IT_Serv_Eng)(http://bit.ly/MSJobs_IT_Serv_Ops)

**Support Engineers / Escalation Engineers:** Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

**Technical Account Managers:** Assure productive use of Microsoft technologies, focusing on delivery quality through planning and governance. http://bit.ly/MSJobs_Delivery_Relationship_Mgmt

**Associate Architect:** Responsible for selling consulting engagements for IT solutions which includes estimating, scoping, and writing statements of work that set expectations and limit risk. Requires domestic and international travel up to 50%. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5810/go/job

**Business Analytics Specialist:** Conduct analysis of business processes and functional requirements. https://jobs-microsoft.icims.com/jobs/5730/go/job

**Business Analytics Specialist:** Identify and synthesize data to support business requirements for adoption of cloud technologies. https://jobs-microsoft.icims.com/jobs/5801/job

**Business Analytics Specialist:** Build, analyze, and present data driven findings for financial plan design and segment program management. https://jobs-microsoft.icims.com/jobs/5755/job

**Category Manager:** Oversee launch readiness and front end experimentation. Requires domestic and international travel up to 25%. https://jobs-microsoft.icims.com/jobs/5867/go/job

**Consultant:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Roving Employee - requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5804/consultant/job

**Data Scientist:** Manipulate large volumes of data applying principles of mathematics and statistics. https://jobs-microsoft.icims.com/jobs/5773/job

**Designer (UX Designer):** Develop user interface and user interaction designs, prototypes and/or concepts for business productivity, entertainment or other software or hardware applications. Requires domestic and international travel up to 25%. https://jobs-microsoft.icims.com/jobs/5707/job

**Firmware Engineer II:** Responsible for developing or testing computer software and firmware applications, systems or services. https://jobs-microsoft.icims.com/jobs/5741/go/job

**Optical Engineer:** Design tools, technologies and processes for next generation optical systems. https://jobs-microsoft.icims.com/jobs/5731/go/job

**Product Engineer:** Design, implement & test computer hardware products that add strategic value to the company. Requires intl. travel up to 50%. https://jobs-microsoft.icims.com/jobs/5790/go/job

**Security Software Engineer:** Responsible for developing static and runtime analysis capabilities so that software security bugs in code can be found quickly and with high confidence. https://jobs-microsoft.icims.com/jobs/5782/go/job

**Senior Consultant:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Roving Employee—requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5794/go/job

**Senior IT Service Ops Analyst:** Examine how data processing operations can be applied to the needs of Microsoft product and engineering teams to improve the user experience and productivity with Microsoft's flagship products, including Windows, SCCM, and Intune. https://jobs-microsoft.icims.com/jobs/5832/go/job

**Senior Partners in Learning Manager:** Develop and manage global initiatives for School Leaders, working closely with corporate and field stakeholders. Requires domestic and international travel up to 50%. https://jobs-microsoft.icims.com/jobs/5791/job

**Senior Sales Excellence Manager:** Develop and maintain Information Technology tools, data platform, reporting, Business Intelligence (BI) and Infrastructure that support the operational systems for U.S. Dynamics sales and partner organizations. https://jobs-microsoft.icims.com/jobs/5774/senior-sales-excellence-manager/job

**Senior Support Escalation Engineer:** Install, configure, support and troubleshoot issues related to Microsoft technologies. Requires travel up to 50% with work to be performed at various unknown worksites throughout the U.S. https://jobs-microsoft.icims.com/jobs/5688/job

**Solution Architect:** Architect software, platform, services, hardware or technology solutions. https://jobs-microsoft.icims.com/jobs/5822/go/job

**Solution Architect/Specialist:** Enhance the Microsoft Customer relationship from a capability development perspective by articulating the value of Microsoft services and solutions & identifying competition gaps in targeted accounts. Roving Employee - Requires travel up to 100% with work to be performed at various and unanticipated worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5747/go/job

**Technology Solutions Professional:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. Requires travel throughout U.S. up to 50% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5772/technology-solutions-professional/job

## MOUNTAIN VIEW, PALO ALTO, SUNNYVALE, CA

**Applied Scientist:** Utilize knowledge in applied statistics and mathematics to handle large amounts of data using various tools. http://bit.ly/MSJobs_Data_Applied_Science

**Data Scientist:** Manipulate large volumes of data, create new and improved techniques and/or solutions for data collection, management and usage. http://bit.ly/MSJobs_Data_Applied_Science

**Design Verification/Validation Engineers:** Responsible for ensuring the quality of Microsoft hardware products. http://bit.ly/MSJobs_Hardware_Design_Verification_Eng

**Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers):** Design, implement and test computer hardware that add strategic value to the company. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

**Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers):** Design, implement and test computer hardware. Requires Domestic and International travel up to 25%. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

**Service Engineers/Managers, Service Operations Engineers, and Systems/Operations Engineers/Site Reliability Engineer:** Research, design, develop, and test operating systems-level software, compilers, and network distribution software. (http://bit.ly/MSJobs_Service_Engineering) (http://bit.ly/MSJobs_IT_Serv_Eng)(http://bit.ly/MSJobs_IT_Serv_Ops)

**Cloud Solution Architect:** Architect software, platform, services, hardware or technology solutions. Requires travel up to 25% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5771/job

## SAN FRANCISCO, CA

**Premier Field Engineers:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Roving Employee—requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. http://bit.ly/MSJobs_Support_Delivery

**Service Engineers/Managers, Service Operations Engineers, and Systems/Operations Engineers/Site Reliability Engineer:** Research, design, develop, and test operating systems-level software, compilers, and network distribution software. (http://bit.ly/MSJobs_Service_Engineering) (http://bit.ly/MSJobs_IT_Serv_Eng)(http://bit.ly/MSJobs_IT_Serv_Ops)

**Technical Account Managers:** Assure productive use of Microsoft technologies, focusing on delivery quality through planning and governance. http://bit.ly/MSJobs_Delivery_Relationship_Mgmt

**Senior Premier Field Engineer, Platforms:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Requires travel up to 75% with work to be performed at various unknown worksites throughout the U.S. https://jobs-microsoft.icims.com/jobs/5704/job

## TAMPA, FL

**Premier Field Engineer:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Requires travel up to 50% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5718/premier-field-engineer/job

**Solution Specialist:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. Requires up to 50% of domestic travel to perform work at unknown customer sites. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5710/job

## TEMPE, AZ

**Solution Architect:** Architect software, platform, services, hardware or technology solutions. https://jobs-microsoft.icims.com/jobs/5831/go/job

## WAUKESHA, WI

**Cloud Solution Architect:** Design, plan, and implement cloud deployments to move customer workloads to Azure. Roving Employee—requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5795/go/job

## FORT LAUDERDALE, FL

**Business Managers and Business Development Managers/Business Development and Strategy Analyst Manager:** Develop business opportunities for sales of software and services. http://bit.ly/MSJobs_Business_Development

**Solutions Sales Professional/Specialist/Technology Solutions Professionals:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. http://bit.ly/MSJobs_Solution_Sales

**Solutions Sales Specialist, Education:** Enhance Microsoft customer relationship from capability development perspective by articulating value of services & solutions & identifying competition gaps in targeted accounts. Requires travel up to 25% with work to be performed at various unknown worksites throughout the U.S. and Latin America. https://jobs-microsoft.icims.com/jobs/4796/go/job

## CAREER OPPORTUNITIES

Microsoft®

**Technology Solutions Professional:** Drive product win rates by proving the value of products to customers and partners. Requires domestic travel up to 75%. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5762/job

### RESTON, VA

**Sr. Reporting Architect:** Responsible for investigating and troubleshooting any issues on a large SAP Business Objects ("BO") Business Intelligence deployment. https://jobs-microsoft.icims.com/jobs/5678/go/job

### BOISE, ID

**Support Engineers / Escalation Engineers:** Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

### CAMBRIDGE, MA

**Data Scientist:** Manipulate large volumes of data, create new and improved techniques and/or solutions for data collection, management and usage. http://bit.ly/MSJobs_Data_Applied_Science

**Support Engineers / Escalation Engineers:** Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

### CHICAGO, IL

**Solutions Sales Professional/Specialist/ Technology Solutions Professionals:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. http://bit.ly/MSJobs_Solution_Sales

**Cloud Solution Architect:** Architect software, platform, services, hardware or technology solutions. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/4804/go/job

### CHARLOTTE, NC

**Support Engineers / Escalation Engineers:** Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

**Technical Advisor Support:** commercial SharePoint across the go-to-markets. https://jobs-microsoft.icims.com/jobs/5753/go/job

### CHEVY CHASE, MD

**Technical Solutions Professional, Azure:** Provide pre-sales technical and architectural support for Microsoft Data Platform and Cloud technologies and corresponding solutions. Requires travel up to 50% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5866/go/job

### NEW YORK, NY

**Solutions Sales Professional/Specialist/ Technology Solutions Professionals:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. http://bit.ly/MSJobs_Solution_Sales

**Cloud Solution Architect:** Architect and deploy Microsoft cloud solutions for customers. Requires travel up to 20% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5720/job

**Consultant:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Requires domestic and international travel up to 25%. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5792/go/job

**Consultant:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Requires Domestic and International travel up to 75%. https://jobs-microsoft.icims.com/jobs/5712/job

**Premier Field Engineer:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5869/premier-field-engineer/job

**Premier Field Engineer:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Requires travel up to 50% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5871/go/job

**Solution Specialist:** Enhance the Microsoft Customer relationship from a capability development perspective by articulating the value of Microsoft services and solutions & identifying competition gaps in targeted accounts. Roving Employee - Requires travel up to 100% with work to be performed at various and unanticipated worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5724/go/job

**Technology Solutions Professional Devices:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5695/job

### WILSONVILLE, OR

**Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers):** Design, implement and test computer hardware that add strategic value to the company. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

### FARGO, ND

**Consultant, Partner Technical:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. https://jobs-microsoft.icims.com/jobs/5833/go/job

### HOUSTON, TX

**Solutions Sales Professional/Specialist/ Technology Solutions Professionals:** Enhance the Microsoft customer relationship from a capability development perspective by articulating the value of our services and solutions and identifying competition gaps in targeted accounts. http://bit.ly/MSJobs_Solution_Sales

**Technical Account Manager:** Assure productive use of Microsoft technologies, focusing on delivery quality through planning and governance. Requires travel up to 25% throughout the U.S. https://jobs-microsoft.icims.com/jobs/5735/go/job

### IRVINE, CA

**Premier Field Engineer:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Requires travel throughout U.S. up to 50% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted https://jobs-microsoft.icims.com/jobs/5825/premier-field-engineer/job

**Premier Field Engineer, Lync:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Requires domestic travel of up to 75% to perform work at various unknown worksites. https://jobs-microsoft.icims.com/jobs/5692/job

**Technology Solutions Professional:** Improve the Enterprise Mobility business metrics (revenue and scorecard) through excellence in technical sales strategy and execution. Requires travel throughout U.S. up to 50% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5778/technology-solutions-professional/job

### IRVING, TX

**Service Engineers/Managers, Service Operations Engineers, and Systems/Operations Engineers/Site Reliability Engineer:** Research, design, develop, and test operating systems-level software, compilers, and network distribution software. (http://bit.ly/MSJobs_Service_Engineering) (http://bit.ly/MSJobs_IT_Serv_Eng)(http://bit.ly/MSJobs_IT_Serv_Ops)

**Support Engineers / Escalation Engineers:** Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

**Premier Field Engineer, SharePoint:** Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Requires domestic travel up to 50%. https://jobs-microsoft.icims.com/jobs/5768/job

**Senior Consultant – ERP:** Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Requires up to 75% travel with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5719/job

**ISELIN, NJ**

**Cloud Solution Architect:** Architect and deploy Microsoft cloud solutions for customers. https://jobs-microsoft.icims.com/jobs/5746/go/job

**Technical Account Managers:** Assure productive use of Microsoft technologies, focusing on delivery quality through planning and governance. http://bit.ly/MSJobs_Delivery_Relationship_Mgmt

**LOS ANGELES, CA**

**Senior Consultant Deliver:** design, planning, and implementation services that provide IT solutions to customers and partners. Requires domestic and international travel up to 50%. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5824/go/job

**RENO, NV**

**Partner OPS Manager:** Launch Plan, initiate, and manage information technology (IT) projects. https://jobs-microsoft.icims.com/jobs/5815/partners-ops-manager-launch/job

**Multiple job openings are available for each of these categories. To view detailed job descriptions and minimum requirements, and to apply, visit the website address listed. EOE.**

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE DEVELOPMENT DIRECTOR**

positions in **Cupertino, CA**.

Job duties include: Own critical upcoming architecture evolutions of software platform and drive definition, strategy, and execution to retool software data platform to meet growing and demanding needs.

Apply by e-mailing resume
to vernon.hui@oracle.com,
referencing 385.18332.

Oracle supports workforce diversity.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**APPLICATIONS DEVELOPER**

positions in **West Conshohocken, PA**.

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications.

Apply by e-mailing resume
to dhaval.shah@oracle.com,
referencing 385.16491.

Oracle supports workforce diversity.

**CLASSIFIED LINE AD SUBMISSION DETAILS:** Rates are $425.00 per column inch ($640 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to: Debbie Sims, Classified Advertising, *Computer* Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 816-2138; fax (714) 821-4010. Email: dsims@computer.org.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *Computer* may reject any advertisement containing any of these phrases or similar ones: "…recent college grads…," "…1–4 years maximum experience…," "…up to 5 years experience," or "…10 years maximum experience." *Computer* reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. *Computer* assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. *Computer* encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# Faculty Position in Distributed and Secure Hardware Systems

## at the Ecole polytechnique fédérale de Lausanne (EPFL)

The School of Engineering (STI) of EPFL invites applications for a faculty position in electronic design for distributed and secure hardware systems at the tenure-track assistant professor level at the EPFL Lausanne campus.

Research activities should relate to one or more of the following subjects: safe and secure circuit design and design tools, hardware components and design methods for Internet of Things (IoT), automotive electronic systems, dependable system-on-chip architectures.

Research activities with an emphasis on experimental approaches and links with industry will be strongly encouraged so as to benefit from the unique experimental facilities that will be made available at the EPFL Lausanne campus in electronics and communication engineering.

As a faculty member of the School of Engineering, the successful candidate will be expected to initiate an independent and creative research program and participate in undergraduate and graduate teaching. Internationally competitive salaries, start-up resources and benefits are offered.

EPFL, with its main campus located in Lausanne, Switzerland, is a dynamically growing and well-funded institution fostering excellence and diversity. It has a highly international campus at an exceptionally attractive location boasting first-class infrastructure. As a technical university covering essentially the entire palette of engineering and science, EPFL offers a fertile environment for research cooperation between different disciplines. The EPFL environment is multi-lingual and multi-cultural, with English often serving as a common interface.

Applications should include a cover letter with a statement of motivation, curriculum vitae, list of publications and patents, concise statement of research and teaching interests, and the names and addresses of at least five referees. Applications must be uploaded in PDF format to the recruitment web site:

**go.epfl.ch/iel-search**

Formal evaluation of candidates will begin on **December 1st, 2016** and continue until the position is filled.

Enquiries may be addressed to:

**Prof. Giovanni De Micheli**
Search Committee Chair
Email: **iel-search@epfl.ch**

For additional information on EPFL, please consult the web sites: **www.epfl.ch**, **sti.epfl.ch** and **iel.epfl.ch**.

*EPFL is committed to increasing the diversity of its faculty, and strongly encourages women to apply.*

THE UNIVERSITY OF
ALABAMA IN HUNTSVILLE

# The University of Alabama in Huntsville

The Department of Computer Science of The University of Alabama in Huntsville (UAH) invites applicants for a tenure-track faculty position at the Assistant Professor level beginning January 2017. The incumbent will augment the department's emphases in at least one of the following areas: cloud computing, particularly secure cloud computing; mobile computing, particularly secure mobile computing; or data science, particularly big data applications. Outstanding candidates who couple cybersecurity with other areas of computing could also be considered.

A Ph.D. in computer science or a closely related area is required. The successful candidate will have a strong academic background, perform funded research, be able to carry out research in areas typical for publication in well-regarded academic conference and journal venues, and be keen on undergraduate education.

The department has a strong commitment to excellence in teaching, research, and service; the hire should have good communication, strong teaching potential, and research accomplishments.

UAH is located in an expanding, high technology area, next door to one of the largest research parks in the nation. Nearby are the NASA Marshall Space Flight Center, the Army's Redstone Arsenal, and many high-tech industries. UAH also has an array of research centers, including in information technology, modeling and simulation, etc. In short, collaborative research opportunities are abundant, and many well-educated and highly technically skilled people are in the area. There is also access to excellent public schools and housing.

UAH has approximately 8000 students. UAH Computer Science offers the BS, MS, and PhD degrees in Computer Science and the MS and PhD degrees in modeling and simulation. Approximately 200 undergraduate majors and 175 graduate students are associated with the unit. Faculty research interests are many and include cybersecurity, mobile computing, data science, software engineering, visualization, graphics and game computing, multimedia, AI, image processing, pattern recognition, and distributed systems. Recent NSF figures indicate the department ranks 30th in the nation in overall federal research funding.

Interested parties should submit a detailed resume with references to Chair, Search Committee, Dept. of Computer Science, The University of Alabama in Huntsville, Huntsville, AL 35899. Qualified female and minority candidates are encouraged to apply. Initial review of applicants will begin starting in late Summer 2016 and will continue until a suitable candidate is found.

The University of Alabama in Huntsville is an affirmative action/equal opportunity employer/ minorities/ females/ veterans/ disabled.

## Apple Inc. has the following job opportunities in Cupertino, CA:

**Software Engineer Applications (Req# 9J3MGR)** Crte & dvlp iOS apps for iAd Workbench.

**Software Engineer Applications (Req# A4Z36R)** Dsgn, build & suppt new critical infrstrctral systms & frameworks.

**Software Development Engineer (Req# A4D32A)** Dsgn & dvlp radio frqncy SW for mobile devices.

**Mechanical Quality Engineer (Req# 9MJMA8)** Des future Apple prod from a quality side (des for quality). Travel req'd: 30%.

**Systems Design Engineer (Req# 9GGV7H)** Anlyz Radio Frqncy (RF) sys prfrmnc for var wireless technologies. Eval prototype sys by prfrmng exprmnts &tests in the lab.

**Software Engineer (Req#9WJSYL)** Des & Dev security server solutions for various Apple prdcts incldng iOS devices & Mac.

**Systems Design Engineer (Req# 9A6LQ2)** Des & dev SW for telecom sys. Travel req 20%.

**Engineering Project Specialist (Req# 9UYNUA)** Supp retail eng team deliv mob & web-bsd sol for cust.

**Engineering Project Lead (Req# 9X5SRA)** Resp for leading cross-func data center infrastr projects w/ Apple internal groups.

**Product Design Engineer (Req# 9TDUWS)** Dev & prctice analytcal methds based on multiphyscs modelng & simulation.

**Hardware Development Engineer (Req#9ZZ4L3)** Dsgn & dvlp integration processes for advncd key display mod from concept to high vol prod. Travel req: 40%

**Software Engineer Applications (Req# 9MFV8Z)** Des & dev SW for Apple Sales & Market sys.

**Software Engineer Applications (Req#9QCQDS)** Dev a variety of high quality & high perfrmng mobile servcs & apps for intrnl use.

**Software Engineer Systems (Req# 9XQQAE)** Supp prod dev. Ident probs & verify sols thru alg des for Apl prods.

**Software Development Engineer (Req#A98VEG)** Des & dev obj-oriented SW arch for embedded camera capture & media frameworks.

**Software Development Engineer (Req#9ZRVBJ)** Dsgn & dev geospatial search infrastrctre servcs & APIs for large scale Maps data processing.

**Product Quality Lead (Req#9FVUR8)** Ensr that accessories meet Apple qlty expcttns. Spcfy & implmnt qlty plan to spprt new prdct intro team. Travel req 25%.

**Systems Design Engineer (Req# A7H4UL)** Dsgn instrumntation to measre & calibrate acoustic perfrmnce of Apple dvices. Travel Req'd 15%

**Software Quality Assurance Engineer (Req#A7T2KX)** Def test approaches & test plans for func test areas, expanding current test cases and exec req tests.

**Software Development Engineer (Req#9TX4UQ)** Plan & excute certfction tstng on the iOS cellular prdcts, iPhone & iPad.

**Professional Services Consultant (Req#9SAT24)** Improve data qty, create & leverage new reporting tools & enable faster decision making.

**Machine Learning Engineer (Req# 9X6SC3)** Support global sales planning & operations teams with analytics to inform decision-making.

**Software Engineer Applications (Req#9Z4VSR)** Des, dev & test distrib & scal feat for iTunes Store servers & testing platform.

**Software Development Engineer (Req#9XC37K)** Res, dev, & improve sig processing & spch rec tools.

**Engineering Project Analyst (Req# 9SZVJL)** Coord ongoing efforts to research, ID, qualify & negotiate new data lic.

**Software Development Engineer (Req#9EVUPC)** Architect & des client code frameworks for iOS & Mac OS apps to eval the qual of the prod.

**Product Design Engineer (Req# 9DCTN6)** Des mech compnts & enclosure of both prtble & dsktp comp prods. Travel req 20%.

**Hardware Development Engineer (Req#9TWNG9)** Resch, dev & intgrt display mod prcss, equip, & mats. Travel req 15%.

**Web/iOS Developer (Req#9K4QN4)** Dsgn & dev iOS apps in Objective-C & Swift to support Sales Training & Communications, & support web dev as needed.

**Software QA Engineer (Req#9UB273)** Dsgn & dev tests, tools, & frameworks to enable reduction of manual testing. Work w/ dvlpmnt team to rvw functionality.

**Mechanical Quality Engineer (Req#9DZVHX)** Contrib to the des. of future Apple prod's w/ a focus on des. for quality. Travel req'd 30%.

**Software Development Engineer (Req#9DH38Q)** Prfrm cell certification testing of iOS dvcs against 3GPP protocol specs.

**Technical Product Manager (Req# 9E3QVW)** Work as platform advocate for reporting & data tools rspnsbl for platform design & arch.

**Software Development Engineer (Req#9LWPCU)** Dsgn, dvlp, & mdfy carrier configs, tools, & frmwrks.

**Software Engineer Systems (Req# 9XZVDY)** Dsgn & dev SW prducts for Applecare retail & repair centers.

**Software Engineer Applications (Req#9TNU95)** Des and dev sw systems based on machine learning and algorithms.

**Software Development Engineer (Req#9T67HY)** Resp for studying & improving the speed & memory of WebKit.

**Software Engineer Applications (Req# 9VBS8J)** Des, dev & support Apple's messaging infrastructure.

**Software Development Engineer (Req#9X83JZ)** Des & dev scal J2EE based data services using SOA for Apple advertising (iAd) on mobile devices.

**ASIC Systems Analyst (Req#A2TU38)** Analyze tech data req's for NICE ICM sys to promote efficiencies taking into acct biz envirnmnt & IT framewrk.

**Hardware Development Engineer**

**(Req#9SYRNK)** Des and dev OLED display technologies. Travel Required: 20%.

**Software Development Engineer (Req#9NW445)** Dev & des SW for Apple Maps framework.

**Software Development Engineer (Req#A853DN)** Dsgn & implmnt Apple's client & server strtgy fr the SW updte sys of Apple prdcts.

**Software Development Engineer (Req#9XUTJU)** Dsgn & dvlp Cellular SW features.

**Software Engineer Systems (Req# 9U3QK8)** Bld, spcfy, dsgn, dvlp, & launch Apple's sensing tech prdct chrctrization & prdction instrumentation sw. Travel req 25%.

**ASIC Design Engineer (Req# 9LUQEU)** Interface w/ all dsciplines to get fnctional prdcts to millions of customers quickly .

**Hardware Development Manager (Req#9L5N5L)** Lead & superv HW Dev Eng in the pow sys des environment.

**Hardware Development Engineer (Req#9Y9M9W)** Des & execute HW & eng valid &character tests for keys solutions, frm proof-of-concept to mass prod. Travel req'd 25%

**ASIC Design Engineer (Req#9R353A)** Drive chip lvl electrical anlyss of high perf processors.

**Software Engineer Systems (Req# 9TDV2L)** Dsgn & dvlp cmpnnts of a lrg and cmplx SW app.

**Software Engineer Applications (Req#9HRV77)** Des & dev Public Key Infrastructure (PKI) & cryptographic svcs.

**Software Developer (Req#9ZY3HW)** Dsgn & dvlp data anlytc SW for the Manufacturing Operation Mngmnt team.

**Software Engineer Applications (Req#9X5SSQ)** Supp NoSQL database infrastructure. Build SW & sys to coord infrastr & app thru automation.

**Software Development Engineer (Req#A4Z35E)** Dsgn & build scalable Hadoop based data prcssing infrastructure.

**Software Engineer Applications (Req#9V4U6J)** Dsgn, dvlp & spprt SW for Apple's Payment Gateway (APG).

**Software Quality Assurance Engineering Manager (Req#9Y7RL7)** Mng team of engs. Comm resps, priorities & deadlines to direct reps.

**Hardware Development Engineer (Req#9QJ3EZ)** Resp for camera tst tech devlpmnt, prdctn tst implmnttn, sys valdtn, dsgn verifictn, & bnchmrkng of Apple camera prdcts. Travel req 25%.

**Software Quality Assurance Engineer (Req#A4S3ZW)** Des, implmnt, execute test plans & test suites based on specification docs.

**Software Development Engineer (Req#9GWR5S)** Des & Build iTunes desktop apps.

**Systems Design Engineer (Req# 9SYTNP)** Eval the latest iPad, iPhone, and Apple Watch HW systms in field & Lab. Travel req'd: 25%.

**Reliability Engineer (Req#9DCSXH)** Prfrm reliability evals of comp HW to idntfy high risk failure modes early in the dsgn lifecycle. Trvl req 15%.

**Hardware Development Engineer (Req#9D2UHP)** Archtct, dsgn & dvlp intgrtn prcsss for advncd key modules, from cncpt to high vlme prdctn. Trvl req 30%.

**Software Engineer Applications (Req#9ZZ4DJ)** Research, dsgn, dvlp, implmnt, rvw & trblsht SW for highly avilbl & large scale core srvcs.

**Software Engineer Applications (Req#9KRR8X)** Dsgn & dvlp SW for Big Data apps.

**Software Engineer Applications (Req#A7A4R2)** Dvlp, create, implmnt, & spprt the web app dvlpmnt of Sales Training App using large scale & high prfmng, OO intrnt technlgs.

**Software Development Engineer (Req#9URUXN)** Dsgn & dvlp wrlss drvrs in kernel & user space running on App Processor & on wrlss chpsts.

**Spatial Data Integration Engineer (Req#9VVTCK)** Resp for spatial data intgrtn for Apple lctn-bsed srvcs.

**Software Engineer Applications (Req#9TSTW8)** Des, dev & impl apps to prevent fraud transactions in Apple Online Store, iTunes & other business areas of Apple Inc.

**ASIC Design Engineer (Req#9H3TDF)** Dev SW & and verify GPU HW.

**Software Development Engineer (Req# 9FV535)** Dsgn, architect, & implement SW features and improvements.

**Systems Design Engineer (Req#9FM4JC)** Create & exec det'ld tst plns for antenna pasv & OTA perf.

**Software Engineer Applications (Req#9VJSW5)** Des & dev SW to enhance & scale content pltfrms.

**Software Engineer Applications (Req#9SYRLB)** Dsgn, dvlp, & maintain core srch components that serve millions of customers worldwide.

**Software Development Engineer (Req#A3WNK8)** Debug & provide sol's for Op Sys, HW, & embdd sys.

**Software Development Engineer (Req#9CZU87)** Dvlp tech for large scale syss, spkn lang, big data, & A.I.

**Software Development Engineer (Req# 9FNU5B)** Dsgn & dvlp Bluetooth SW & drivers for dvcs & accessories.

**Hardware Development Engineer (Req# 9NYMUS)** Dsgn, dvlp, & chrctrze LED & laserbsd optical syss. Trvl req 25%.

**Software Development Engineer (Req#9CRTFW)** Rspnsble for tstng & validation of pre-release SW.

**Systems Design Engineer (Req# 9DPVWR)** Rsponsble for electromagnetic compatibility (EMC) dsgn of elctroncs prdcts including comptrs, cellular phnes, & other dvics.

**Software Development Engineer (Req#9XD3PQ)** Des & dev real-time comm SW for FaceTime.

**Software Development Engineer (Req#A58NHC)** Des & dev SW for mobile advert sys.

**Software Engineer Applications (Req#9SV5LE)** Dsgn, dev & deploy data warehouse & anlytics solutions for mltple business groups at Apple.

**Refer to Req# & mail resume to**
**Apple Inc., ATTN: L.J.,**
**1 Infinite Loop 104-1GM,**
**Cupertino, CA 95014.**
**Apple is an EOE/AA m/f/**
**disability/vets.**

## CAREER OPPORTUNITIES

### Apple Inc. has the following job opportunities in San Francisco, CA:

**Software Engineer Applications (Req#9W8TDU)** Des & dev SW for large-scale online web services, data pipelines & data warehouse sys.

### Apple Inc. has the following job opportunities in Sacramento, CA:

**Information Systems Engineer (Req#9ZT3CA)** Dsgn, dvlp & support SW deployment processes.

### Apple Inc. has the following job opportunities in Austin, TX:

**Technical Support Engineer (Req#A2N2PB)** Spprt various Channel Service bus actvts related to the Service Transformation prgrms. Travel req. 20%.

Refer to Req# & mail resume
to Apple Inc., ATTN: L.J.,
1 Infinite Loop 104-1GM,
Cupertino, CA 95014.
Apple is an EOE/AA m/f/
disability/vets.

Microsoft®

Microsoft Corporation currently has the following openings (job opportunities available at all levels, including Principal, Senior and Lead levels):

### REDMOND, WA

**Business Program Managers:** Plan, initiate, and manage technology and business projects. http://bit.ly/MSJobs-Buss_Oper_Prog_Mgmt

**Business Program Manager:** Plan, initiate, and manage technology and business projects. Telecommuting Permitted. Position allows employee to reside anywhere in the U.S. and telecommute to perform work exclusively from home. https://jobs-microsoft.icims.com/jobs/6260/go/job

**Program Managers (All levels, including Leads and Managers):** Coordinate program development of computer software applications, systems or services working with development and product planning teams. Requires Domestic and International travel up to 25%. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

**Program Managers (All levels, including Leads and Managers):** Coordinate program development of computer software applications, systems or services working with development and product planning teams. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

**Business Program Manager:** Plan, initiate, and manage technology and business projects. Requires domestic and international travel up to 25%. https://jobs-microsoft.icims.com/jobs/5696/job

**Hardware Engineering Program Manager:** Coordinate program development of hardware products or systems, working w/ development & product planning teams. Requires dom., regional & intl. travel up to 50%. https://jobs-microsoft.icims.com/jobs/5849/go/job

**Program Manager II:** Coordinate program development of computer software applications, systems or services, working with development and product planning teams. Requires domestic and international travel up to 50%. https://jobs-microsoft.icims.com/jobs/5855/go/job

**Senior Business Program Manager:** Plan, initiate, and manage technology and business projects. Requires domestic, regional, and international travel up to 25%. https://jobs-microsoft.icims.com/jobs/5843/go/job

**Senior Program Manager:** Lead the end-to-end strategy and execution plan development for the EBC customer experience, demos, interactive vignettes and displays and in-room technology experiences. https://jobs-microsoft.icims.com/jobs/5786/job

**Senior Research Program Manager:** Responsible for participating in multiple research projects and interface with researchers and engineers. https://jobs-microsoft.icims.com/jobs/5776/go/job

**Operations Program Managers:** Plan, initiate, and manage information technology (IT) projects. (http://bit.ly/MSJobs_Ops_PM)

### MOUNTAIN VIEW, PALO ALTO, SUNNYVALE, CA

**Program Managers (All levels, including Leads and Managers):** Coordinate program development of computer software applications, systems or services working with development and product planning teams. Requires Domestic and International travel up to 25%. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp) (http://bit.ly/MSJobs_IT_ProgMgr)

**Program Managers (All levels, including Leads and Managers):** Coordinate program development of computer software applications, systems or services working with development and product planning teams. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

### CAMBRIDGE, MA

**Program Managers (All levels, including Leads and Managers):** Coordinate program development of computer software applications, systems or services working with development and product planning teams. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

**Multiple job openings are available for each of these categories. To view detailed job descriptions and minimum requirements, and to apply, visit the website address listed. EOE.**

## TECHNOLOGY

# Expedia, Inc.

currently has openings for the following opportunities in our **Bellevue, WA office (various/levels/types:)**

**Software Engineers: (728.SWE-AUG)** Design, implement, and debug software for computers including algorithms and data structures.

**Database Developers: (728.DBD-AUG)** Coordinate changes to computer databases, test and implement the database applying knowledge of database management systems.

**Business Development Managers: (728.1553)** Use data-driven analysis to identify, evaluate, and drive strategic business opportunities, alliances and/or joint ventures.

**Application Engineers: (728.894)** Identify opportunities to automate application release process and help to implement potential solutions.

**BI Developers: (728.1942)** Design and develop reporting solution using SQL Server, SSIS, SSAS, and SSRS as per business requirements.

**Regional Head of Partners: Connect: (728.947)** Implement process in region and ensure that process has buy in from relevant departments it effects. 20-25% travel to various unanticipated sites throughout the U.S. and internationally required.

**Business Analysts: (728.1740)** Formulate and apply mathematical modeling and other optimizing methods to develop and interpret information.

**Directors, Product Management – Technical: (728.1960)** Collaborate closely with business stakeholders and peer product managers to help define product strategy that meet customers' needs while satisfying scalability, reliability, performance, and resource costs.

**Managers, Software Development: (728.1770)** Manage team of developers in implementing high quality web-based applications and high volume transactional services, including design, development, and deployment of new business functionality.

**Program Managers: (728.1335)** Responsible for microcomputer software product design features and coordinating development of software among functional groups through product release.

**Directors, International Tax: (728.1796)** Provide tax advice, develop international tax strategy, and drive the implementation of solutions. Travel to various unanticipated sites throughout the U.S. and internationally required.

**Technical Product Managers: (728.TPM-AUG)** Gather detailed business requirements from stakeholders and work closely with technology staff to translate requirements into functional designs and specifications.

**Reporting and Analysis Managers: (728.2373)** Support, influence, and challenge business decisions with data and analyses.

Send your resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference position and Job & Job ID# listed above.

---

# Egencia LLC

currently has openings for the following opportunities in our **Bellevue, WA office (various/levels/types:)**

**Software Engineers: (728.SWE-EGB-AUG)** Design, implement, and debug software for computers including algorithms and data structures.

**Database Developers: (728.DBD-EGB-AUG)** Coordinate changes to computer databases, test and implement the database applying knowledge of database management systems.

Send your resume to: Egencia/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference position and Job & Job ID# listed above.

## Hotwire, Inc.

currently has openings for the following opportunities in our **Bellevue, WA office (various/levels/types:)**

**Software Engineers: (728.SWE-HW-AUG)**

Design, implement, and debug software for computers including algorithms and data structures.

**Database Developers: (728.DBD-HW-AUG)**

Coordinate changes to computer databases, test and implement the database applying knowledge of database management systems.

**Transport Managers: (728.1564)**

Plan, and coordinate transportation activities of Hotwire organization that provide transportation services. Travel to various unanticipated sites throughout the U.S. required.

Send your resume to: Hotwire/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004.
Must reference position and Job & Job ID# listed above.

---

## Orbitz Worldwide, LLC

currently has openings for the following opportunities in our **Chicago, IL office (various/levels/types:)**

**Software Engineers: (728.SWE-ORB-AUG)**

Design, implement, and debug software for computers including algorithms and data structures.

**Database Developers: (728.DBD-ORB-AUG)**

Coordinate changes to computer databases, test and implement the database applying knowledge of database management systems.

**Web Designers: (728.2188)**

Define visual hierarchy and treatment for range of optimizations and new features of products throughout entire lifecycle.

**SEO Analysts: (728.2101)**

Responsible for execution of search engine optimization ("SEO") strategies and tactics and supporting SEO manager in day-to-day functions by analyzing, optimizing, and reporting on natural search performance.

Send your resume to: Orbitz/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004.
Must reference position and Job & Job ID# listed above.

SOFTWARE

**Oracle America, Inc.**

has openings for

## SOFTWARE DEVELOPER

positions in **Mineola, NY**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to
ankush.chadha@oracle.com,
referencing 385.14493.

Oracle supports workforce diversity.

TECHNOLOGY

**Oracle America, Inc.**

has openings for

## CONSULTANT

positions in **Minneapolis, MN**.

Job duties include: Analyze requirements and deliver functional and technical solutions. Implement products and technologies to meet post-sale customer needs. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to
scott.king@oracle.com,
referencing 385.19352.

Oracle supports workforce diversity.

TECHNOLOGY

**Oracle America, Inc.**

has openings for

## APPLICATIONS DEVELOPERS

positions in **Chicago, IL**.

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications. Write code, complete programming and perform testing and debugging of applications.

Apply by e-mailing resume to
scott.bockelman@oracle.com,
referencing 385.19134.

Oracle supports workforce diversity.

TECHNOLOGY

## Intuit Inc.

has openings for the following positions in **Santa Clara County, including Mountain View, California** or any office within normal commuting distance:

**Product Managers (Job code: I-440):** Design and bring to market revenue generating, customer-driven software products and services that deliver great customer experiences.

Positions located in **San Diego, California**:

**Offering Program Managers (Job code: I-1981):** Plan and drive the delivery of high-quality software releases, working closely with product managers and engineering managers to ensure products exceed customers' expectations. **Software Engineers (Job code: I-2057):** Apply software development practices to design, implement, and support individual software projects. **Senior Software Engineers in Quality (Job code: I-300):** Apply senior level software engineering practices and procedures to design, influence, and drive quality and testability of products and services.

Positions located in **Plano, Texas**:

**Senior Network Engineers (Job code: I-1975):** Implement and support new network technologies in our on-premise, hybrid and cloud environments. Some travel may be required to work on projects at various, unanticipated sites throughout the United States.

To apply, submit resume to Intuit Inc., Attn: Olivia Sawyer, J203-6, 2800 E. Commerce Center Place, Tucson, AZ 85706.
You must include the job code on your resume/cover letter. Intuit supports workforce diversity.

EDITOR DAVID ALAN GRIER
George Washington University; grier@gwu.edu

AUDIO

# The Means of Production

**David Alan Grier,** George Washington University

*Technical leadership remains one of the great problems of software production.*

In the study of software development, Ralph Flanders is a minor historical figure who's connected to agile methods only through a long chain with many rusted and incomplete links. Generally known as a business leader and politician, his moment of glory came in 1954 when he introduced the motion into the US Senate that would censure his colleague Senator Joe McCarthy. Yet, early in his career, after an apprenticeship as an industrial engineer, he identified several key problems that hindered industrial production, which are the same problems that hinder software development.

The connection between industrial engineering and software engineering is strong. Early software developers, including John von Neumann's colleague Herman Goldstine, shaped their early notions of software with concepts from industrial production. The word "program" came from the term that described a plan for factory operation. Flow charts came from the diagrams that traced the movement of material through a plant. Sequencing originally referred to the steps of factory operation. The terms "program quality" and the "software life cycle" were borrowed from the work to develop production plans that would create uniform goods.

Flanders did most of his engineering work in the 1920s, when Henry Ford was the dominant industrial engineer. Ford argued for production methods that placed workers under strict production control: each worker had one place on the production line, did one job repeatedly, and learned how to do that job efficiently. Flanders argued that this approach was ill-suited to the production of many products. He claimed that the most challenging problems of manufacturing were not inefficient workers but instead manufacturing overhead. Flanders wrote that many plants saw only "a moderate increase in output" when they switched to Ford's methods, and yet these plants saw a great increase in overhead, particularly in "the number of foremen and in the amount of clerical labor they were called upon to do."

See **www.computer.org/computer-multimedia** for multimedia content related to this article.

Flanders made recommendations that are familiar to anyone who has followed recent trends in lean and agile methodologies. He urged manufacturers to organize production by product, not by process, and to design their product to be as simple as possible. He suggested that factories use small teams and encourage team members to develop a wide and comprehensive set of skills. Finally, he recommended that groups work as independently as possible so that the group leader could supervise the work rather than attend to meetings, memos, and reports. "The foreman is bounded only by his production orders and his schedule," Flanders stated; within "these limits he is king of his territory."

Flanders appears to have had limited influence as an engineer, though his ideas might have been thwarted first by the reduced production of the Great Depression and second by the overwhelming demands of World War II. However, his concern for technical leadership echoed throughout the engineering literature: it was picked up by management consultant Peter Drucker in the 1950s, Russian engineer S.P. Mitrafanov in the 1960s, and software engineer Frederick P. Brooks in the 1970s. A "major part of the cost [of software] is communication and correcting the ill effects of miscommunication," Brooks wrote in his seminal work on software production. Hence, he concluded, computer software needs to be built by "as few minds as possible."

Technical leadership remains one of the great problems of software production, for it requires a different set of skills for conventional mid-level management, skills that handle the tasks of planning, allocating, and coordinating assets. Technical leaders need to understand the details of their product, teach key skills to their workers, and assess the output of their team. Excessive communication, coordination, and reporting reduce the efficiency of these leaders—often fatally so.

We should care about Flanders because he cared about technical leadership long before it was fashionable to have such concerns. Like many who followed him, he recognized that such leadership was difficult and that we could make it easier only by finding better ways of organizing the means of production. C

**DAVID ALAN GRIER** is an associate professor at George Washington University. Contact him at grier@gwu.edu.

# Move Your Career Forward

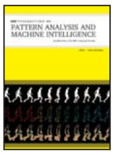
## IEEE Computer Society Membership

## Explore These Resources on Emerging Computing Paradigms

### Build Your Knowledge

***Transactions on Multi-Scale Computing Systems***

*TMSCS* is a peer-reviewed journal devoted to computing systems that exploit multiscale and multi-functionality. Research topics relate to high-performance computing, computational sustainability, storage organization, and efficient algorithmic information distribution and processing.



***IEEE Transactions on Pattern Analysis and Machine Intelligence***

With an impact factor placing it in the top 5 of all electrical and electronic engineering journals, *TPAMI* features research in computer vision and image processing, pattern analysis and recognition, and selected areas of machine intelligence, with a particular emphasis on machine learning for pattern analysis.

***Computing in Science Engineering***

In a clear and accessible format, CiSE presents scientific and computational contributions to the hard sciences, which share a common need for efficient algorithms, system software, and computer architecture to address complex problems.

***IEEE Symposium on Foundations of Computer Science***

**9-11 October 2016, New Brunswick, NJ, USA**
This annual event is the flagship conference sponsored by the IEEE Computer Society Technical Committee on the Mathematical Foundations of Computing (TCMF). Held annually in the fall, FOCS is a key forum for original research on the theory of computation.

**FOR DIRECT LINKS TO THESE RESOURCES, VISIT**

www.computer.org/computer-resources

IEEE computer society

CELEBRATING 70 YEARS

# solidThinking®

## Activate | Compose | Embed

### System Modeling & Simulation

Simulation & optimization of multi-disciplinary systems

Extensive block libraries including Modelica support

Co-simulation via FMI and Multi-body

### Visual Math Environment

Matrix Analysis

Differential Equations

Signal Analysis

Control Design

```
[T,X] = ode45(@Func,T,X, options);
y = readvector(file,1,1,3);
p = polyfit(X,Y,6);
x1 = polyval(p,X);
t1=linspace(0,t,1000);
y1=fft(h);
y2=sin(2*pi*fc*t1);
plot (abs (ft));
```

### Embedded Development

Diagram to Code

Interactive HIL

Visual Real-Time Operating System

StateCharts

Learn more at **solidThinking.com**

An Altair Company