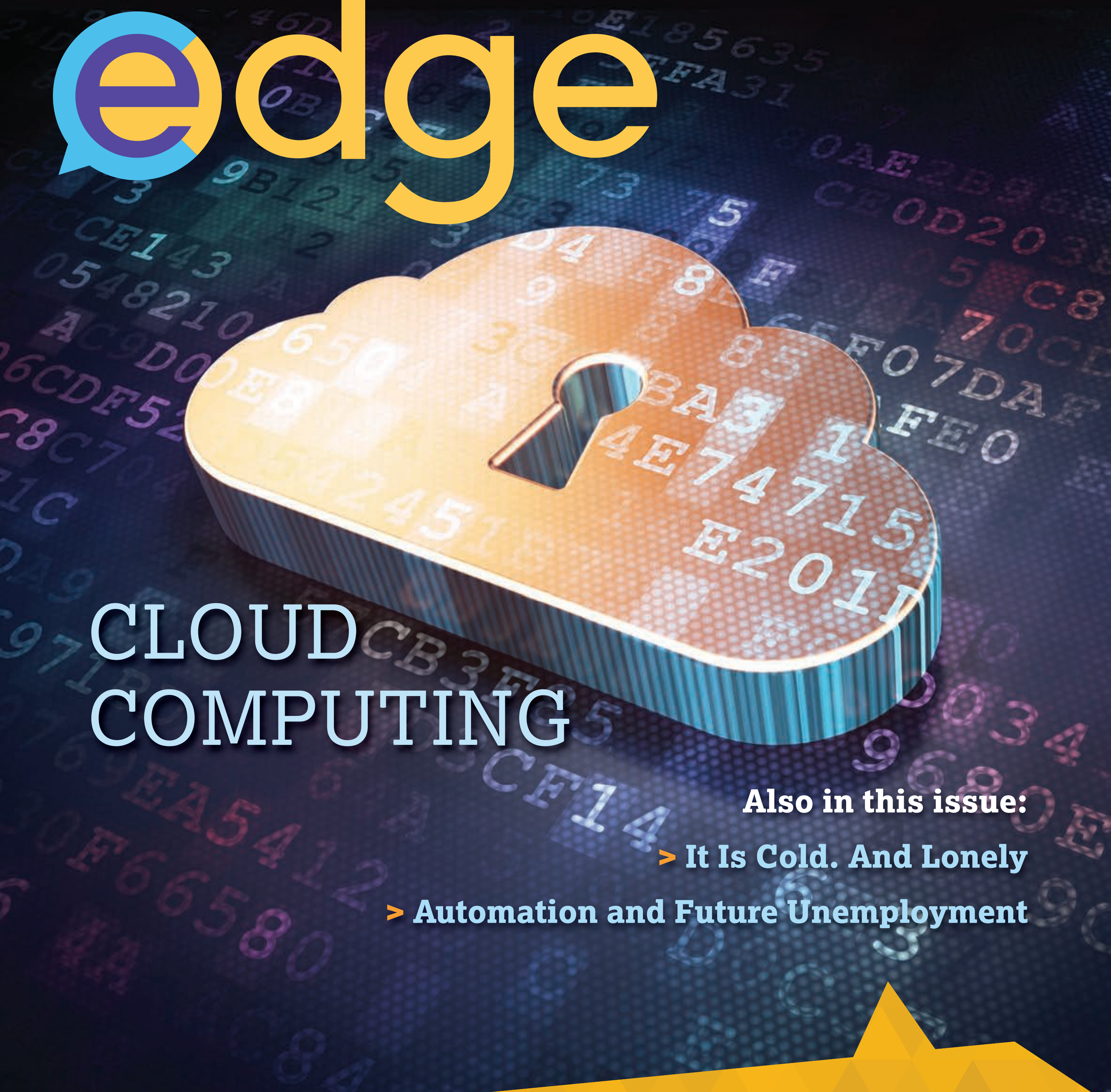
COMPUTING edge



CLOUD COMPUTING

Also in this issue:

> It Is Cold. And Lonely

> Automation and Future Unemployment

JUNE 2016

www.computer.org

IEEE

IEEE computer society

CELEBRATING 70 YEARS

COMPUTING edge

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

Printed with inks containing soy and/or vegetable oils

IEEE

IEEE computer society
CELEBRATING 70 YEARS

BPA WORLDWIDE

Circulation: ComputingEdge (ISSN 2469-7087) 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.

Postmaster: Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

JUNE 2016 • VOLUME 2, NUMBER 6

COMPUTING edge

70
Automation and Future Unemployment

Departments

Virtual Learning and Object Reconstruction

This installment highlighting the work published in IEEE Computer Society journals comes from IEEE Transactions on Learning Technologies *and* IEEE Transactions on Pattern Analysis and Machine Intelligence.

VIRTUAL ENVIRONMENTS, REAL LEARNING

H. Chad Lane, University of Illinois at Urbana–Champaign

In his familiar deadpan style, comedian Steven Wright once said, "In school they told me 'practice makes perfect,' and then they told me 'nobody's perfect.' So I stopped practicing." Although we certainly hope learners reach a different conclusion, his observation highlights the central role of practice during knowledge and skill acquisition. Studies on human learning have demonstrated that providing learners with opportunities to practice improves their performance over time, reveals gaps in their knowledge, uncovers their misconceptions, and leads them to correct their errors. In so many words, practice is learning.

Enter virtual learning environments (VLEs). With virtual practice of a target skill, educators and researchers can design realistic scenarios that address specific needs, provide levels of difficulty and fidelity that match developmental aspects of learning, and allow learners to practice as long as required or desired. Multiuser VLEs connected online essentially remove any physical barriers to collaboration, teaching, and learning.

These benefits are exemplified in Stephanie August and her colleagues' work at Loyola Marymount University, as described in "Virtual Engineering Sciences Learning Lab: Giving STEM Education a Second Life" (*IEEE Trans. Learning Technologies*, vol. 9, no. 1, 2016, pp. 18–30).

In their paper, the authors introduce the Virtual Engineering Sciences Learning Lab (VESLL), a VLE developed for use in Second Life, a widely used free online virtual environment. They extol the flexibility of open and free virtual worlds, explaining how they enable both traditional, lecture-style presentations and highly interactive, collaborative, and self-directed work. In stark contrast to massive open online courses and other less immersive platforms, learners move freely within the virtual space, using chat, action, and movement to accomplish their learning goals.

VESLL focuses on key engineering education topics, including number systems, circuit design, and differential equations. These tasks are embedded "in-world," allowing learners to design, share, and interact. The authors also report on a user evaluation of the system, breaking down the appeal of the approach from a learner's perspective. Further, preliminary knowledge assessments from use of VESLL suggest that the system is at least as effective as classroom learning—an important result considering that learners work entirely virtually.

This is an exciting period for educational technology research. Nonstop innovation in human–computer interaction, including computer graphics, virtual reality, and user-sensing technologies, has made new and powerful pedagogical approaches possible. Schools, museums, and workplace learning environments are some of the most important beneficiaries of computer science's broad advances, and should serve as critical contexts for future investigations.

H. CHAD LANE is an associate professor of educational psychology and informatics at the University of Illinois at Urbana–Champaign. Contact him at hclane@illinois.edu.

RECONSTRUCTING OBJECTS FROM A SINGLE PICTURE

David Forsyth, University of Illinois at Urbana–Champaign

How should computer programs reconstruct objects from a single picture? The objects we see in pictures are 3D, but the picture itself is a projection of the object to two dimensions. The geometry of projection isn't that complicated—for most cases, as long as we have at least two pictures of an object, there are programs that can produce excellent, detailed, and useful reconstructions.

 2469-7087/16/$33.00

However, a single picture is very different because many important phenomena are conflated to produce that picture. For example, a pixel might be dark because it has dark paint on it, there isn't much light, or the surface is tilted away from the light. In their paper "Shape, Illumination and Reflectance from Shading," Jon Barron and Jitendra Malik of the University of California, Berkeley, describe a novel and effective approach to untangling these various physical effects to produce—from a single picture—a reconstruction that contains the shape of an object, the patterns of reflectance on the object, and the illumination field in which the object sits (*IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, 2015, pp. 1670–1687). Recovering any of these is a classical problem in computer vision.

Surprisingly, inferring all of these properties together yields superior results. An image pixel's color is determined by the light arriving at that location in the camera—light that was reflected through the lens and to the camera from some surface in the world. The surface will reflect different fractions of the incident light at different wavelengths (the reflectance of the surface), changing the reflected light's color. The illumination field in which the object sits might vary with direction (think of a flashlight beam in a dark room) and might be colored. The surface will collect more light from directions that point directly toward it, and less light from directions that are nearly tangent. Each of these effects is quite difficult to model accurately in ways that admit useful inference; for example, light doesn't just arrive at a surface patch from a light source but rather is reflected from patch to patch, and most surfaces reflect light unevenly across the outgoing directions.

Reconstruction pixel by pixel won't work. There has been some success with approximate physical models, but there's been rather more disappointment caused by ill-behaved mathematics. Instead, Barron and Malik show that strong prior constraints apply to the spatial structure of surface reflectance and geometry, to the choice of colors that appear on the surface, and to the illumination field. The trick is to produce a reconstruction that exactly reproduces the image while having the best value of cost functions that score compatibility with these constraints. By doing so, the authors are leading research in this area away from arcane mathematical questions and toward considerations of the kinds of reconstructions that are more likely. As a lagniappe, there's a rant on one author's website detailing problems with the current processes of reviewing and publication in computer vision. C

DAVID FORSYTH is a professor of computer science at the University of Illinois at Urbana–Champaign. Contact him at daf@illinois.edu.

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

CS FOCUS

Magazine Roundup

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to cloud migration and microchip manufacturing. Here are highlights from recent issues.

Computer

Computing helps convert a torrent of data and observations into information that emergency-response teams can use to effectively direct and deploy aid. This is the focus of *Computer*'s May 2016 special issue on **emergency response technology**.

IEEE Software

DevOps aims to reduce the time required between making a change to a software system and placing the change into normal production, while still maintaining quality. This is the topic of *IEEE Software*'s May/June 2016 special issue on **software engineering for DevOps**.

Computing in Science & Engineering

CiSE's May/June 2016 special issue includes five of the best papers from the IEEE Special Technical Committee on Broadening Participation's **RESPECT (Research on Equity and Sustained Participation in Engineering, Computing, and Technology) 2015 conference**. The articles are the second part of a two-part series addressing research on broadening participation in computing at all levels of education, focusing on gender, race, and ethnic diversity.

IEEE Internet Computing

IEEE Internet Computing's May/June 2016 special issue surveys **cloud storage**–related topics and challenges.

IEEE Security & Privacy

For 37 years, the **IEEE Symposium on Security and Privacy** has been a forum for presenting computer-security and electronic-privacy developments and for bringing together leading researchers and practitioners. To expose these events to a wider audience, *IEEE S&P*'s March/April 2016 special issue presents papers from the 2015 symposium's workshops.

Published by the IEEE Computer Society

IEEE Cloud Computing

Both hybrid and community clouds entail federation, the ability to securely share computing resources. To better understand how to accomplish secure **cloud federation**, the author of "Cloud Federation Management and Beyond: Requirements, Relevant Standards, and Gaps," from *IEEE Cloud Computing*'s January/February 2016 issue, identifies six fundamental requirements and six deployment models. He also examines relevant existing standards and discusses further work required on best practices, tools, and standards.

IEEE Computer Graphics and Applications

IEEE CG&A's May/June 2016 special issue contains several articles describing research advances in **scientific visualization**, involving both established and prototype toolkits and systems. The articles look not only at the challenges of domain-specific scientific visualization applications but also at challenges that have broader applicability.

IEEE Intelligent Systems

The Internet of Things' (IoT's) rapid growth has yielded a massive increase in data generated by connected devices and sensors. Smart data lets users exploit this information to gain deep insights and make effective decisions. "Internet of Things to Smart IoT through Semantic, Cognitive, and Perceptual Computing," from *IEEE Intelligent Systems*' March/April 2016 issue, discusses key **smart data–related AI research** that could lead to a more intelligent IoT.

IEEE MultiMedia

Ubiquitous multimedia's wide-ranging applications and large data volumes present unprecedented challenges and unique opportunities for multimedia-computing research. This was the main theme of the 2015 IEEE International Symposium on Multimedia (ISM 2015). *IEEE MultiMedia*'s April–June 2016 special issue on **ubiquitous multimedia** gives the authors of the symposium's top papers a forum to present further research results.

IEEE Annals of the History of Computing

Due to a flow of information into, but not out of, the Soviet Union, Western policy analysts wary of growing Soviet computing prowess sought intelligence from many sources. Based on an examination of previously unexplored trip reports from Western computer experts who visited the Soviet Union, "Peering through the Curtain: **Soviet Computing through the Eyes of Western Experts**," from *IEEE Annals*' January–March 2016 issue, looks at Cold War interactions between Western computer specialists and their Soviet counterparts.

IEEE Pervasive Computing

The global demand for increasingly capable **mobile visual-computing** applications running on low-energy computing platforms shows no sign of slowing down. Thus, engineers must understand the technologies used to deliver these platforms and the challenges system architects face. This is the topic of "The Rise of Mobile Visual Computing Systems," which appears in *IEEE Pervasive Computing*'s April–June 2016 issue.

IT Professional

As **distributed systems** progress, they seem to continually revisit older concepts. "The Curious Case of Distributed Systems and Continuous Computing," from *IT Pro*'s March/April 2016 issue, examines the various types of distributed systems and their evolution, and predicts where the technology might be headed.

IEEE Micro

The articles in *IEEE Micro*'s March/April 2016 special issue are written by selected authors who made presentations at 2015's **Hot Chips 27** conference on high-performance microprocessors and related integrated circuits.

Computing Now

The Computing Now website (http://computingnow.computer.org) features **up-to-the-minute computing news** and blogs, along with articles ranging from peer-reviewed research to opinion pieces by industry leaders.

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

EDITOR'S NOTE

Looking for Answers in the Cloud

Cloud computing is maturing, and developers and users are finding new applications, as well as new challenges, for the technology. This *ComputingEdge* issue discusses where cloud computing is headed and the problems it will face in getting there.

With cloud computing emerging as the enterprise's enabling technology of choice, a fundamental challenge is understanding complicated cloud architectures, how they work, and the value they bring. This is discussed in *IEEE Cloud Computing*'s "Understanding Complex Cloud Patterns."

Although intended to ensure cloud service providers' security, reliability, and legal compliance, current certifications in the field quickly become outdated, according to *IEEE Security & Privacy*'s "Dynamic Certification of Cloud Services: Trust, but Verify!" Dynamic certification, on the other hand, provides automated monitoring and auditing to verify providers' ongoing adherence to requirements.

As the Open Container Initiative works to standardize containers' format and configuration, the increasingly popular technology could become the cloud infrastructure's "narrow waist," a uniform interface bridging the many current and emerging cloud services. This is the topic of *IEEE Internet Computing*'s "Toward a Standard Interface for Cloud Providers: The Container as the Narrow Waist."

The authors of *Computer*'s "Deciding When and How to Move HPC Jobs to the Cloud" propose a decision-support system to help users determine, based on efficiency and cost-effectiveness, when and how to run jobs that require high-performance computing on cloud-based resources rather than on-premise clusters.

IEEE Pervasive Computing's "Cloud-Based AI for Pervasive Applications" offers a brief introduction on how to include sophisticated computer vision, speech recognition, text analytics, and machine learning in pervasive-computing applications.

ComputingEdge articles on subjects other than cloud computing include the following:

- Given our increasing reliance on the oceans to solve problems such as finding new energy and mineral supplies, there is a growing need for applications that work well in marine settings, a topic addressed by "Underwater Visual Computing: The Grand Challenge Just around the Corner," from *IEEE Computer Graphics and Applications*.
- In *IEEE Software*'s "It Is Cold. And Lonely," Grady Booch says next-generation software-intensive systems will be taught, not programmed, which means developers will face challenges in building, delivering, and evolving them.
- "Automation and Future Unemployment," from *IT Professional*, examines the prediction that IT and related technologies are in the early stages of bringing about massive, systemic unemployment.

IEEE Computer Society 2016 Call for **MAJOR AWARD NOMINATIONS**

Help Recognize Computing's Most Prestigious

IEEE Computer Society awards recognize outstanding achievements and highlight significant contributors in the teaching and R&D computing communities. All members of the profession are invited to nominate individuals who they consider most eligible to receive international recognition of an appropriate society award.

Computer Entrepreneur Award
Sterling Silver Goblet
Vision and leadership resulting in the growth of some segment of the computer industry.

Technical Achievement Award
Certificate/$2,000
Contributions to computer science or computer technology.

Harry H. Goode Memorial Award
Bronze Medal/$2,000
Information sciences, including seminal ideas, algorithms, computing directions, and concepts.

Hans Karlsson Award
Plaque/$2,000
Team leadership and achievement through collaboration in computing standards.

Richard E. Merwin Distinguished Service Award
Bronze Medal/$5,000
Outstanding volunteer service to the profession at large, including service to the IEEE Computer Society.

Harlan D. Mills Award
Plaque/$3,000
Contributions to the practice of software engineering through the application of sound theory.

Computer Pioneer Award
Silver Medal
Pioneering concepts and development of the computer field.

W. Wallace McDowell Award
Certificate/$2,000
Recent theoretical, design, educational, practical, or other tangible innovative contributions.

Taylor L. Booth Award
Bronze Medal/$5,000
Contributions to computer science and engineering education.

Computer Science & Engineering Undergraduate Teaching Award
Plaque/$2,000
Recognizes outstanding contributions to undergraduate education.

IEEE-CS/Software Engineering Institute Watts S. Humphrey Software Process Achievement Award
(Joint award by CS/SEI)
Plaque/$1,500
Software professionals or teams responsible for an improvement to their organization's ability to create and evolve software-dependent systems.

Deadline: 15 October 2016
Nomination Site: awards.computer.org
For more information visit: www.computer.org/awards

IEEE

IEEE computer society

Understanding Complex Cloud Patterns

CLOUD COMPUTING CONTINUES TO EMERGE AS THE ENTERPRISE'S ENABLING TECHNOLOGY OF CHOICE. The trend has been to leverage clouds as complex, highly heterogeneous, and distributed architectures, including hybrid and multiclouds. Moreover, as we leverage heterogeneous public clouds, there's a need to provide high-speed data transfer and data integration between public and private clouds. Thus we have the emerging use of the intercloud, or special-purpose networking between public and private clouds.

The use of these approaches and architectures has launched a new set of technologies within the world of the cloud to deal with governance, security, and management of these solutions. Thus, we have to layer technologies to deal with these issues. However, the largest challenge is just understanding these complex cloud architectures, how they work, and the value they're likely to bring. This article provides the information you need to make solid choices around the use of complex, heterogonous, and widely distributed architectures.

DAVID S. LINTHICUM

Cloud Technology Partners
david@davidlinthicum.com

Hybrid Clouds

Back in 2008, the industry first began the hybrid cloud computing model discussion as defined by the National Institute of Standards and Technology (NIST).[1] Cloud computing purists pushed back hard.[2] After all, they already thought private clouds were just another name for the datacenter. To them, the idea of hybrid clouds that used private clouds or traditional computing platforms was just as ridiculous.

However, fast forward to 2016, and the use of the hybrid cloud model is pretty widespread. This complex cloud architecture needs to be understood in terms of value, proper implementation, and its ability to leverage different architectural patterns to best balance the loads between private and public clouds.

NIST defines hybrid clouds as follows:

> The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).[1]

Over time, it became clear that hybrid cloud computing approaches have valid roles within enterprises, as IT tries to mix and match public clouds and local IT assets to get the best bang for the buck. Now it's the cloud computing providers who are pushing back on hybrid cloud computing, as they instead try to promote a pure public cloud computing model. However, these providers are inadvertently hurting the adoption of cloud computing. Although public cloud computing has valid applications, the path to it isn't all that clear to rank-and-file enterprises.

Hybrid clouds provide a clear use case for public cloud computing. Specific aspects of existing IT infrastructure (say, storage and compute) occur in public cloud environments, and the remainder of the IT infrastructure stays on premises. Take the case of business intelligence in the cloud: Although some people promote the migration of gigabytes of operational data to the cloud, many others find the hybrid approach of keeping the data local and the analytical processing in the cloud to be much more practical.

Using a hybrid model is a valuable approach to architecture since you can mix and match resources between local infrastructures, which is typically a sunk cost but difficult to scale, with infrastructure that's scalable and provisioned on demand. You place the applications and data on the best platforms, then span the processing between them.

The use of hybrid computing acknowledges and validates the fact that not all IT resources should exist in public clouds today—and some might never exist in public clouds. Given the compliance issues, performance requirements, and security restrictions, the need for local computing is a fact of life. The hybrid model helps us all better understand what compute cycles and data have to be kept local, and what can be processed remotely.

Of course, some cloud providers already have their eye on leveraging a hybrid model. These new kids on the block even provide management and operating systems layers specifically built for hybrid clouds. However, most public cloud providers are religious about pushing everything outside of the firewall (after all, that's where they make their money).[3]

Also keep in mind the concept of federated clouds, which are related to hybrid clouds. A federated cloud (also called cloud federation) is a configuration of multiple external and internal cloud computing services that align with the enterprise's business needs. A federation is the union of several smaller parts that perform a common action, and can be fine- or course-grained services, infrastructure resources (such as storage), or even applications.

For more information on specific hybrid cloud architectures, see my "Cloud Tidbits" column in this issue.[4]

Intercloud and Intracloud

Cloud-to-cloud data transfer can be either intracloud (such as within Amazon Web Services [AWS]) or intercloud (such as between AWS and Google). Although these intra- and intercloud services provide wider bandwidth than the open Internet, they typically cost more to leverage, and, in most cases, are add-on options where you're charged by the amount of traffic.

Intracloud data transfer performance is directly related to the size of the pipe within the cloud service, as well as to performance-enhancing services, such as cache services that might be in use. Intracloud moves data from place to place in the same cloud, typically from a set of machine instances to another set of machine instances, usually with storage.

Typically, intracloud data transfer is between tenants, virtual machines, or applications and datastores. The approaches and technology vary greatly, so you should run your own benchmarks to duplicate the scenario you plan to implement. Moreover, do a total cost of ownership (TCO) analysis to be sure that you won't be surprised by the data transfer bill your public cloud provider sends you at the end of the month.

Intercloud data transfer is even more complex, having to exchange data with cloud services that might not like each other. Moreover, the open Internet is the typical mode of transport, so the same issues arise here as with cloud to enterprise. I suspect that cloud providers will get better at this architecture in the future, for the sake of their mutual clients.[5] However, as we progress in time, more intercloud connects are becoming dedicated circuits running between providers just for the purpose of high-speed data transfer. This trend will continue as public cloud providers understand that it's in their best interest to work with other public cloud providers for the best outcome of their cloud-to-cloud solutions.

Multicloud

The use of multiple clouds has evolved from architectural patterns required to solve business problems. Many business departments want to use cloud computing within public clouds, outside of the company firewall. Today we have many types of public and private clouds that provide security, governance, and management tools to support other clouds, and which can be combined to create a composite solution.

Enterprises move to multicloud for a variety of reasons:

- Single cloud solutions typically don't provide the breadth and depth of functionality that enterprises require for all of their cloud computing solutions.
- The rise of cloud management platforms (CMPs) gives enterprises a single interface to help provision, manage, and scale complex environments.
- Usage-based pricing makes it easier for enterprise IT to evaluate the cost of cloud computing, including show-back and charge-back services.
- Companies that want to move applications into public clouds need a range of services, including different database, middleware, development, and compute services, and this drives the use of multiple cloud computing platforms.
- The growing use of platforms, infrastructure, and software in the cloud results in multiple forms of clouds. Consequently, IT must often support two or more public clouds for development and operations teams that use the cloud to create new business applications and services.

The *RightScale 2015 State of the Cloud Report* proved that the movement to multicloud is real, with 82 percent of those surveyed identifying multicloud as their current strategic direction.[6]

Enterprises moving to multicloud face some critical choices, including the types and brands of cloud to leverage and the approaches and technology to use in managing a multicloud solution. These approaches seem to have some common patterns of failure, as well as patterns of success.

Multicloud architectures have their own sets of pros and cons. The core question that many enterprises ask is: How much cloud heterogeneity is a good thing? Also, when does heterogeneity bring too much complexity and risk? Indeed, you can think of multicloud as a complex hybrid cloud that has more than two brands of clouds within the architecture.

Enterprises that tried to maintain homogenous on-premises IT infrastructures lost the battle a long time ago. Typical enterprise architectures have been built through years of solving tactical problems with whatever technology seemed to be right at the time. Over the years, these on-premises technology solutions became very heterogeneous, and thus very complex.

Multicloud shouldn't evolve the same way. Those charged with picking the right cloud technology should consider the solution patterns to fit the problem patterns, and use that as a guide to select and deploy the right public and private cloud technologies. Although some might find that a single cloud provider is the best solution,[6] most are driving cloud solutions using best-of-breed technology, and thus end up with a number of cloud types and brands.

Although this isn't the first time enterprises moving to new technology have faced the "homogeneous versus heterogeneous" question, the move to cloud computing brings some new challenges and confusion. Clouds are, indeed, platforms, but they also provide common resources that other cloud brands may share.

Clouds take more of a service-oriented approach to architecture, and the enterprise typically ends up with a common services catalog that might link back to many types and brands of clouds. Clouds represent a collection of services that can be mixed and matched to form applications, more so than monolithic applications themselves (see Figure 1).

For example, an enterprise could take storage services from one public cloud provider and mash them up with compute services from another provider, and perhaps introduce database services that are running on premises. The ability to build solutions out of best-of-breed cloud services provides a solid foundation to justify a multicloud approach and is the primary driver of multicloud use. The more clouds you leverage, the more services you have in your catalog, and thus application development becomes more of an assembly process. This allows enterprises to quickly build or change applications, providing the value of agility and speed-to-market for the business. Agility is the fundamental way that cloud computing provides value. The more clouds (such as with multicloud), the more the business is able to solve problems or adjust to changes in the market, which means it can make more money.

Finally, consider off- and on-premises cloud models, such as virtual private clouds (VPCs). Each model is sometimes preferred over private and public because you can have a dedicated zone in a public cloud provider for one customer. These models provide cloud computing's cost effectiveness, but also provide the data privacy that business might prefer.

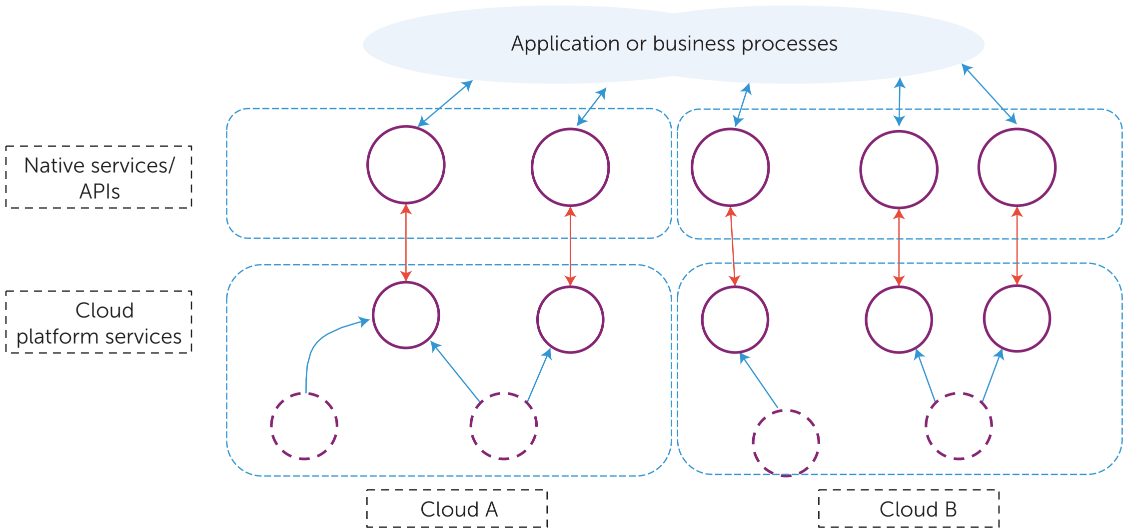


FIGURE 1. When leveraging cloud-based platforms, you can mix and match services to form business processes or applications. This drives the value of multicloud.

THE USE OF COMPLEX CLOUD ARCHITECTURES IS STILL AN EVOLVING SCIENCE. It's helpful to understand the core patterns, and perhaps a match to your ultimate IT solution as you evolve your own cloud computing strategy.

As a path forward, you should consider your "as is" state, including data, processes, and compute, and how those components map to cloud, or should map to cloud computing solutions. Although the task is laborious, it's well worth the time invested due to the efficiencies that can be gained from the use of these complex cloud architectures. Of course, the trade-offs are the cost of dealing with the complexities versus the value that can be had. Some total cost of ownership analysis should be done before moving down this path.

References

1. P. Mell and T. Grance, *The NIST Definition of Cloud Computing, Nat'l Inst. of Standards and Technology*, NIST Special Publication 800-145, 2011; http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.
2. E. Brown, "Final Version of NIST Cloud Computing Definition Published," *NIST Tech Beat*, 25 Oct. 2011; http://www.nist.gov/itl/csd/cloud-102511.cfm.
3. D. Linthicum, "Why the Hybrid Cloud Model Is the Best Approach," *InfoWorld*, 27 Jan. 2011; www.infoworld.com/article/2625289/cloud-computing/why-the-hybrid-cloud-model-is-the-best-approach.html.
4. D.S. Linthicum, "Emerging Hybrid Cloud Patterns," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 88–91.
5. D. Linthicum, "All You Need to Know about Cloud Data Transfers," *InfoWorld*, 23 Mar. 2012; www.infoworld.com/article/2619818/cloud-computing/all-you-need-to-know-about-cloud-data-transfers.html
6. *RightScale 2015 State of the Cloud Report*, RightScale, 2015; www.rightscale.com/lp/2015-state-of-the-cloud-report.

DAVID S. LINTHICUM *is senior vice president of Cloud Technology Partners. He's also Gigaom's research analyst, and he frequently writes for* InfoWorld *on deep technology subjects. His research interests include complex distributed systems, including cloud computing, data integration, service-oriented architecture, and big data systems. Contact him at david@davidlinthicum.com.*

This article originally appeared in
IEEE Cloud Computing, *vol. 3, no. 1, 2016.*

Editors: Mohamed Kaâniche, mohamed.kaaniche@laas.fr | Aad van Moorsel, aad.vanmoorsel@ncl.ac.uk

Dynamic Certification of Cloud Services:

Trust, but Verify!

Sebastian Lins, Pascal Grochol, Stephan Schneider, and Ali Sunyaev | University of Cologne

In late 2014, hundreds of celebrities' private pictures were publicly disclosed when brute-force attacks on usernames, passwords, and security questions compromised Apple's iCloud.[1] After a comprehensive investigation, Apple reported that its cloud service didn't lock out account access properly after a number of failed login attempts. This attack attracted global media attention, renewing the debate on cloud computing's security and privacy.

Cloud service use offers benefits for both individual customers and organizations. Organizations, in particular, gain financial and technical advantages by harnessing the cloud's ubiquitous, on-demand provision of up-to-date computing resources and services, such as pay-per-use networks, applications, and storage. However, incidents such as the iCloud breach undermine these advantages, fostering security, privacy, and reliability concerns as well as doubts about cloud service providers' trustworthiness. Ultimately, such doubts prevent many organizations from adopting cloud services.

Secure Cloud Services in an Ever-Changing and Hostile Environment

In response to these concerns, cloud service certifications (CSCs) have been developed, for example, Cloud Security Alliance's Security, Trust & Assurance Registry (https://cloudsecurityalliance.org/star) and EuroCloud Star Audit (https://eurocloud-staraudit.eu). CSCs foster service adoption by establishing trust in the cloud market and increasing its transparency.[2] They attempt to assure a cloud service provider's high level of security, reliability, and legal compliance for a validity period of one to three years. However, existing CSCs are retrospective—they reflect only the technical and organizational measures fulfilled at the time they were issued. CSC conditions and requirements might not be met later in the validity period, which in turn threatens certification reliability and trustworthiness. In assuring ongoing certification adherence, CSCs face several challenges:

- *Inherent cloud computing characteristics.* Cloud services are part of an ever-changing environment due to inherent cloud computing characteristics such as on-demand provisioning and entangled supply chains.[3] Furthermore, cloud services have a faster technology life cycle than other industries.
- *Ongoing architectural changes.* Changes to hardware or software configurations or to subservice providers might cause certification violations or security vulnerabilities.[4] Such security vulnerabilities might go undetected for a long time if appropriate monitoring mechanisms aren't in place. Current certifications can't track and manage continuous software

 Published by the IEEE Computer Society 2469-7087/16/$33.00 © 2016 IEEE

deployments, especially for agile software development and cloud applications.

- *Environmental threats.* Cloud computing and IT environment changes, such as the emergence of new vulnerabilities, require cloud providers to adapt their services. Otherwise, major security incidents might threaten the cloud service or reveal harmful vulnerabilities, voiding the certification.
- *Legal and regulatory landscape changes.* Cloud services have highly dynamic legal and regulatory landscapes. Existing laws are being adjusted and new laws proposed to cope with the challenges resulting from society's digital transformation and IT's growth. The privacy and security of the EU–US Safe Harbor data-sharing agreement was questioned recently. These legal and regulatory dynamics might affect cloud service customers' and providers' responsibilities and require successive updates to certification criteria.
- *Deliberate discontinuance.* Cloud service providers might deliberately discontinue adherence to CSC criteria to realize benefits; for example, they might reduce the number of service desk staff to save money.

In light of such factors, CSCs' long validity periods might make cloud customers question the reliability and trustworthiness of issued certificates. To address traditional certifications' drawbacks, the German Federal Ministry of Education and Research funded a project to dynamically certify cloud services called New Generation Certification (NGCert; www.ngcert.eu), which uses third-party auditors to continuously certify cloud services. NGCert builds on previous research projects, including the Advanced Security Service Certificate for SOA (www.assert4soa.eu), which introduced novel techniques and tools to certify

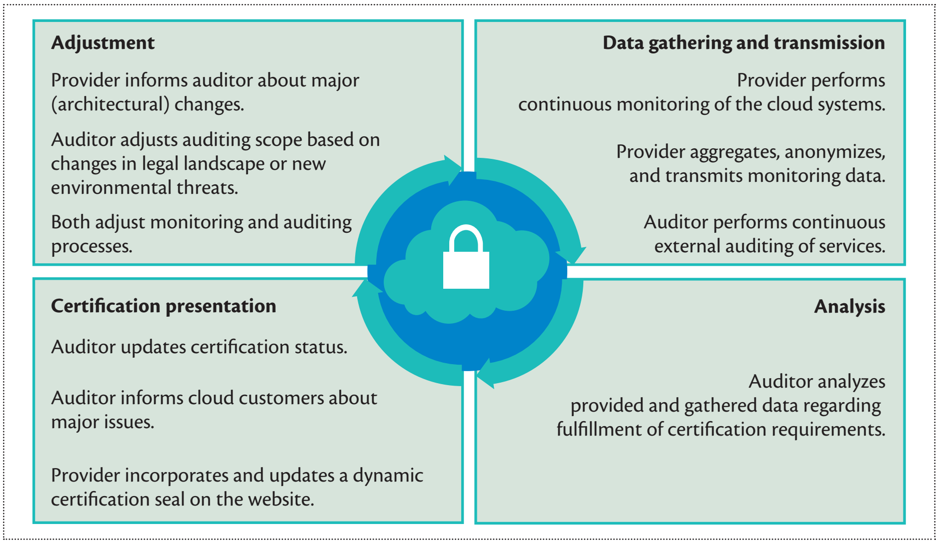


Figure 1. The dynamic certification process. Following these steps can help ensure continuously secure and reliable cloud services and establish trustworthy cloud service certifications.

the security properties of complex service-oriented applications, and the Certification Infrastructure for Multilayer Cloud Services (www.cumulus-project.eu), which developed models, processes, and tools that support the certification of cloud services' security properties.

Dynamic Certification

We believe that dynamic certification is required to ensure continuously secure and reliable cloud services and establish trustworthy CSCs. The dynamic certification process entails automated monitoring and auditing techniques and transparent provision of audit-relevant information to verify providers' ongoing adherence to certification requirements.[5] As Figure 1 shows, dynamic certification has four major dimensions: *semi- or fully automated data gathering and transmission, semi- or fully automated data analysis, certification presentation*, and *process adjustment*.

Semi- or Fully Automated Data Gathering and Transmission

To realize dynamic certification, cloud service providers must establish an internal monitoring and auditing department. This department performs extensive, frequent monitoring operations related to virtualized environments, intrusion detection, service-level agreements, compliance, networks, and so on. Hence, providers must have appropriate cloud-monitoring tools and architectures and data-logging facilities. In addition to monitoring processes, the department might implement internal auditing processes that gather monitoring data from different systems and aggregate, filter, and anonymize audit-relevant data. For example, it might deploy an internal auditing team of software agents who gather data across implemented cloud-monitoring tools and then prepare monthly auditing reports. Later in the process, this department serves as a data transmission and communication interface between providers and involved auditors. Moreover, auditors can perform external continuous audits to gather audit-relevant data; for example, they can deploy software agents to validate cloud infrastructure changes.[6] Likewise, third-party auditors can externally validate the integrity of data stored in a cloud.[7] However, providers' resistance to monitoring and auditing—because of security and privacy concerns and system heterogeneity—will restrict external continuous auditing.

Semi- or Fully Automated Data Analysis

To assess ongoing certification adherence, auditors request that providers transfer data and submit reports at regular intervals. Therefore, auditors should offer a Web interface for uploading provider data. Similarly, providers might transfer monitoring logs that auditors can analyze to assess CSC criteria adherence. Auditors can realize synergy effects by connecting to and building on providers' existing monitoring systems (for example, Nagios). Auditing mechanisms such as decision support systems must be implemented to assess cloud services automatically, cope with identified deviations, and trigger alerts in cases of nonadherence.

Certification Presentation

Auditors must regularly inform cloud customers about certification adherence (or lack thereof). Dynamic certification can grant customers insight beyond what they could analyze themselves or what providers offer because auditors possess greater knowledge, technical expertise, and access to sensitive data. Subsequently, auditors and/or providers should provide a Web front end presenting up-to-date information about auditing processes and results and general cloud service operation. In addition, cloud customers have expressed interest in receiving information about how and when data is gathered and analyzed to increase comprehensibility and accountability. More important, in cases of critical certification violations or major security incidents, auditors should inform customers immediately.

Process Adjustment

Finally, the dynamic certification process must be adjusted frequently to cope with an ever-changing environment. On one hand, emerging environmental threats or legal and regulatory landscape changes might induce auditors to adjust their auditing scope by, for example, adding new certification criteria. In these cases, auditors must set time limits for providers to deal with such events. Time limits should conform to the event's criticality, required efforts, and costs and should be documented in dynamic certification agreements between auditors and providers. On the other hand, architectural changes to cloud services—for example, adding hardware components or new service functionalities—can lead providers and auditors to adjust their monitoring and auditing processes.

A Win–Win Situation

To be widely adopted, dynamic certification of cloud services must be technologically and economically feasible. Providers and auditors need the motivation and expertise to participate in continuous monitoring and auditing. To motivate providers, the process's perceived benefits must be higher than its perceived expenditures. If increasing numbers of customers demand transparent, trustworthy certified cloud services, providers might start to open up to dynamic certifications. Considering the drawbacks of current CSCs, dynamic certification is particularly beneficial for cloud service customers, providers, and auditors. Following are some of dynamic certification's benefits.

Dynamic Certification Increases Transparency and Trustworthiness for Cloud Service Customers

Cloud customers cede governance to the cloud service provider. This lack of control coupled with a lack of transparency about how and where data is stored and processed can lead customers to fear that their data will be compromised or leaked. Compared to a traditional certification process, dynamic certification counteracts this lack of control by increasing the transparency of providers' operations. In this way, dynamic certification might increase customers' trust in cloud services.

Dynamic Certification Leads to Improved Quality and Cost Savings for Cloud Service Providers

Cloud service systems and processes can be improved by implementing suitable monitoring techniques and evaluating ongoing feedback about their performance. In addition, providers receive expert external assessments of their systems. Service providers can reduce costs through successive service improvements, improved customer support, and fewer individual customer audits. Finally, by offering customers more transparent services, cloud providers differentiate themselves in the market and gain a competitive advantage.

Dynamic Certification Increases Auditors' Certification Reliability and Auditing Efficiency

Typically, adherence to certification criteria is evaluated annually. Hence, certification deviations or breaches might not be detected until long after their occurrence. In contrast, continuous auditing allows auditors to actively detect and investigate critical defects as they occur. Moreover, automated continuous auditing is cost effective because auditors can test larger samples and examine data more efficiently than with manual auditing. Through timely detection and continuous assurance of certification adherence, dynamic certification can improve the trustworthiness of auditors' CSCs. Furthermore, the up-to-date information provided in auditor reports will be more relevant to customer decision makers.

Challenges: A Long Way to Go

Dynamic certification's benefits notwithstanding, several challenges must be tackled to ensure its practical application.

Automation

To be cost effective, dynamic certification requires strong automation and formalization of monitoring and auditing processes. Thus, providers must implement comprehensive logging and monitoring systems that gather audit-relevant data continuously. However, automating auditing processes is challenging and not achievable for every auditing process, as previous research has shown.[8]

Data Manipulation Risks

When cloud service providers transfer monitoring data to auditors, it's important that they implement log and database protection mechanisms to prevent malicious data manipulation such as withholding evidence or euphemizing data. Consequently, providers must prove high data integrity and confidentiality levels. Furthermore, cloud customers believe that an independent institution should establish dynamic certification guidelines and regularly review implemented mechanisms and transferred data. However, cloud providers, customers, and auditors generally assume that manipulation risks are low because ongoing manipulation entails high costs and penalties and risks customers revealing tampered data when using the system.

Security Issues

Providers face several security challenges, including protecting deployed data exchange interfaces, authorizing external auditors, and securing data transmission. Moreover, when audit-relevant data is provided via defined interfaces, providers have to ensure its availability to auditors at all times. For example, attackers might target interfaces by performing distributed denial-of-service attacks that disrupt continuous auditing. In the worst-case scenario, this might lead to nonadherence to requirements, because auditors will lack the corresponding audit information. Likewise, auditors must implement comprehensive security mechanisms to ensure data integrity and confidentiality and to guard information against improper modification by external attackers. In particular, when auditors store multiple providers' certification data, they become prime targets for attacks. Attackers might seek to modify audit-relevant data to disrupt or bias auditor assessments, resulting in certification nonadherence or customer dissatisfaction. Likewise, attackers might tamper with the data presented to customers to falsely indicate bad service behavior.

Dynamic certification is required to ensure continuously secure and reliable cloud services and establish trustworthy cloud service certifications.

Service Individualism and Complexity

When implementing dynamic certification in practice, auditors are faced with highly individual and complex cloud service systems due to customized or legacy systems and incorporated third-party services. Therefore, auditors must adjust their auditing methodologies to the providers' context and capabilities. Flexible, standardized auditing systems can reduce adjustment efforts by letting auditors easily integrate or exclude providers.

Recommendations

We recommend the following steps to cope with certification's prevailing challenges and enable its practical application.

Set Up Continuous Monitoring and Data Transmission Systems

Providers have already equipped their service centers with sophisticated monitoring technologies to gather service data and quickly detect malicious attacks, failures, and outages. Providers and auditors should evaluate how they might leverage existing monitoring systems for dynamic certification by matching available monitoring data with required audit-relevant data. Dynamic certification requires providers to gather data continuously as well as collect additional information to evaluate certification adherence. Thus, providers should implement, for example, a comprehensive layered logging framework and continuous monitoring techniques to ensure that all audit-relevant data is available.[9,10] More important, to ensure that providers don't manipulate monitoring logs, providers should build on computer forensics solutions such as secure logging as a service, which stores virtual machines' logs and provides access to forensic investigators.[11]

Develop External, Continuous Auditing Architectures and Methods

Although research has focused on implementing and evaluating automated auditing of information systems since the early 90s, it has mostly examined continuous auditing for internal purposes only. In the context of cloud computing, researchers recently proposed methodologies that enable third-party authorities to audit data integrity, data location compliance, and cloud infrastructure changes. Some researchers have already begun evaluating which of these methodologies might be used for dynamic certification.[3,4,12] Building on these findings, auditors and providers should jointly discuss and develop a comprehensive architecture to continuously audit a broad variety of CSC criteria.

Increase Transparency by Integrating Customers into Certification Process

Existing monitoring systems are designed for internal monitoring purposes only, and the gathered monitoring information is kept in-house to be inspected solely by system administrators. A few cloud providers, including Amazon (http://status.aws.amazon.com) and SalesForce (https://trust.salesforce.com), have begun informing customers about service availability, service performance, and major security issues. Auditors and providers must bring customers into the dynamic certification process by providing detailed information about certification adherence and cloud service behavior, for instance, through customer-centered information dashboards.

Adjust and Set Up New Auditing Business Models

Introducing dynamic certification lets auditors adjust existing business models or set up new ones. For example, auditors might provide auditing-as-a-service capabilities for customers once the required dynamic auditing infrastructure is established. This service model would allow customers to review auditing results on demand for a fee. First movers in dynamic certification should focus on creating barriers that prevent new market entries and price wars. Besides leveraging high initial infrastructure investments, auditors should try to achieve economies of scale by integrating many cloud service providers into their auditing infrastructure, and generate switching barriers for providers (for example, proprietary auditing services). In contrast, new market entries or late adopters should follow a niche strategy by concentrating on one auditing domain (such as dynamic infrastructure auditing or continuous privacy auditing) with specialized service offers or lower auditing fees.

Review and Verify Dynamic Certification Processes

Similar to traditional certifications, independent third-party authorities or accreditation bodies must verify that auditors can continuously audit providers; behave ethically; and employ reliable, secure, and trustworthy dynamic auditing practices. Therefore, these authorities should regularly review and verify dynamic certification processes. Furthermore, they must prevent auditors from behaving opportunistically or maliciously (for example, by misusing providers' data or blackmailing providers to buy IT solutions) and ensure that auditing organizations are independent.

The ever-changing cloud environment, fast technology cycles, regulatory changes, and increased adoption of business-critical applications demand highly reliable cloud services. Dynamic certification of cloud services can prove providers' high level of reliability and security to potential customers. However, methods to efficiently and continuously assess cloud services are still in their infancy. Organizations such as the Cloud Security Alliance and EuroCloud are developing processes and techniques for continuous auditing of cloud services. We believe that introducing dynamic certification is a step toward more trustworthy and transparent cloud computing environments. ■

References

1. "Apple Media Advisory: Update to Celebrity Photo Investigation," Apple Inc., Sept. 2014; www.apple.com/pr/library/2014/09/02Apple-Media-Advisory.html.
2. A. Sunyaev and S. Schneider, "Cloud Services Certification," *Comm. ACM*, vol. 56, no. 2, 2013, pp. 33–36.
3. S. Lins et al., "What Is Really Going on at Your Cloud Service Provider? Creating Trustworthy Certifications by Continuous Auditing," *Proc. 48th Hawaii Int'l Conf. System Science* (HICSS 15), 2015, pp. 5352–5361.
4. P. Stephanow and N. Fallenbeck, "Towards Continuous Certification of Infrastructure-as-a-Service Using Low-Level Metrics," *Proc. 12th IEEE Int'l Conf. Advanced and Trusted Computing* (ATC 15), 2015; www.ngcert.de/wp-content/uploads/2015/07/Towards-continuous-certification-of-Infrastructure-as-a-Service-using-low-level-metrics.pdf.
5. I. Windhorst and A. Sunyaev, "Dynamic Certification of Cloud Services," *Proc. 8th Int'l Conf. Availability, Reliability and Security* (ARES 13), 2013, pp. 412–417.
6. F. Doelitzscher et al., "Validating Cloud Infrastructure Changes by Cloud Audits," *Proc. IEEE World Congress on Services* (SERVICES 12), 2012, pp. 377–384.
7. B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," *IEEE Trans. Cloud Computing*, vol. 2, no. 1, 2014, pp. 43–56.
8. C.E. Brown, J.A. Wong, and A.A. Baldwin, "A Review and Analysis of the Existing Research Streams

in Continuous Auditing," *J. Emerging Technologies in Accounting*, vol. 4, no. 1, 2007, pp. 1–28.

9. R.K.L. Ko et al., "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," *Proc. IEEE World Congress on Services* (SERVICES 11), 2011, pp. 584–588.
10. "Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations," Nat'l Inst. Standards and Technology, Sept. 2011; http://csrc.nist.gov/publications/nistpubs/800-137/SP800-137-Final.pdf.
11. S. Zawoad, A.K. Dutta, and R. Hasan, "SecLaaS: Secure Logging-as-a-Service for Cloud Forensics," *Proc. 8th ACM SIGSAC Symp. Information, Computer and Communications Security* (ASIA CCS 13), 2013, pp. 219–230.
12. S. Lins et al., "Trust Is Good, Control is Better: Creating Secure Clouds by Continuous Auditing," to be published in *IEEE Trans. Cloud Computing*, 2016.

Sebastian Lins is a PhD student in the Department of Information Systems at the University of Cologne. Contact him at lins@wiso.uni-koeln.de.

Pascal Grochol is a PhD student in the Department of Information Systems at the University of Cologne. Contact him at grochol@wiso.uni-koeln.de.

Stephan Schneider is a postdoctoral researcher in the Department of Information Systems at the University of Cologne. Contact him at schneider@wiso.uni-koeln.de.

Ali Sunyaev is a professor in the Department of Information Systems at the University of Cologne. Contact him at sunyaev@wiso.uni-koeln.de.

This article originally appeared in IEEE Security & Privacy, *vol. 14, no. 2, 2016.*

Editor: Yong Cui • cuiyong@tsinghua.edu.cn

Toward a Standard Interface for Cloud Providers

The Container as the Narrow Waist

Silvery Fu and Jiangchuan Liu • *Simon Fraser University*

Xiaowen Chu • *Hong Kong Baptist University*

Yueming Hu • *South China Agricultural University*

Containers underwent a whirlwind adoption across cloud providers recently, as the Open Container Initiative works toward standardizing the container format and configuration. The container could become the cloud infrastructure's "narrow waist," bridging a proliferation of existing and emerging cloud services.

Cloud computing has experienced a rapid growth in the past decade; it's estimated that 95 percent of companies have introduced cloud services to facilitate their business in 2015.[1] As a constellation of cloud technologies, applications, and business models keep emerging, there's a strong demand on standardization towards a set of core technical specifications, which are to be adopted and shared by cloud vendors.[2] Pioneering works include the Open Virtualization Format (OVF),[3] Open Cloud Computing Interface (OCCI; http://occi-wg.org), and IEEE's P2301 (Cloud Profiles) and P2302 (Intercloud) working drafts.[4] In particular, OVF is an established standard, adopted by the American National Standards Institute (ANSI) and International Organization for Standardization (ISO) that defines an open and portable format for packaging and distributing software run on virtual machines (VMs). A typical .ovf package includes descriptors for hardware requirements, network, storage, and security settings. IEEE P2301 aims to unify the variety of design options adopted in cloud computing systems, by organizing those options into profiles, which provide guidance on developing standard-based products for cloud vendors. These efforts (mostly *virtualization-oriented*) have helped improve the interoperability among cloud ecosystem participants. Enhancements and the establishment of new standards, however, are still required to better address application deployment and portability, as well as resource management with minimized overhead.

New opportunity arises from the advancement of cloud infrastructures. *Containerization*, a technology that enables fine-grained resource control and isolation by encapsulating applications inside containers, has undergone a whirlwind adoption and gained native support across major cloud providers, including Amazon Elastic Compute Cloud (EC2) Container Service (ECS; http://aws.amazon.com/ecs), Google Container Engine (https://goo.gl/oxBS2e), and Microsoft Azure Container Service.[5] The container, in its simplest form, is a collection of OS kernel utilities configured to manage the resources that an application uses. The Open Container Initiative (OCI; www.opencontainers.org), launched under the auspices of the Linux Foundation in mid-2015, aims to establish open industry standards for the container's runtime and format. Following the container's popularity, it has readily gained sponsorship from more than 30 companies and organizations, including leading cloud providers and application platforms.

With this in mind, here we provide a high-level overview of container technology, along with its standardization status. We discuss the container's integral modules and explain why we envision it becoming the "narrow waist" of the cloud infrastructure, bridging a proliferation of existing and emerging cloud services.

 2469-7087/16/$33.00 © 2016 IEEE

The Container: Rationale and Key Modules

In general, a container offers cloud providers a lightweight tool to achieve resource multiplexing and control, as an alternative to virtualization (particularly those hypervisor-based). A closer relative to the container is an operating system process, because both of them essentially encapsulate a (single) application runtime. What the container offers additionally is the capabilities of controlling and isolating OS resources assigned to the runtime, meanwhile including complete dependencies in a container instance. As such, we can also refer to a container as a *virtual environment* (VE).

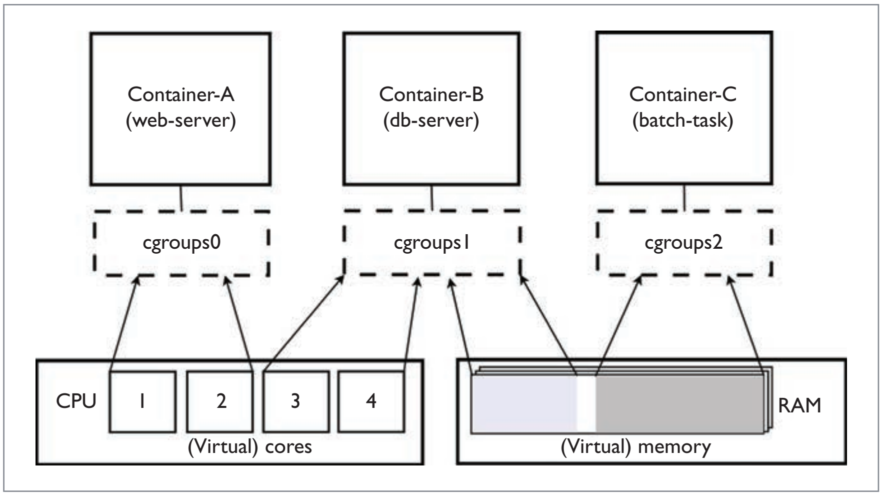


Figure 1. The control groups (cgroups) module for CPU and memory. The module defines a collection of kernel resource controllers.

The idea of the container and OS-level virtualization isn't new. Linux-VServer[6] and OpenVZ (https://openvz.org) are two previous container-based virtualization platforms. Yet the container has only come to the fore in recent years, for two reasons. First, it has shifted from the original role as a "hypervisor-free" VM, where a single container instance had to be built full-fledge to support a full OS (as in VServer and OpenVZ), to a lightweight runtime environment for applications. Second, recent platforms significantly simplify the procedure for container creation and management. These two advancements meet the growing need of deploying cloud-based distributed applications with "just enough" performance overhead and maintenance cost. They've been jointly achieved by Docker (www.docker.com), the most established and popular container by far.

Docker relies on utilities of the modern Linux kernel to create and manage container runtime. (Eventually Docker developed native implementation of these modules, as a solution for cross-platform support.[7]) First and foremost, the *control groups* (cgroups) module defines a collection of kernel resource controllers for (including but not limited to) CPU, memory, and network and disk I/O. User-level code is allowed to customize these controllers through cgroups' virtual file system. At the runtime, cgroups is assigned to a process through function hooking, by which resource accesses of the process will trigger the corresponding hooks. As such, without intervening performance-critical execution paths, cgroups is able to achieve resource tracking and control efficiently. Figure 1 gives an example showing the use of cgroups with container runtime. Here, `cgroups1` defines the control groups of two CPU cores 3 and 4, and a limited amount of memory shaded in light gray. It's assigned to ContainerB that encapsulates a database server runtime (denoted by *db-server*). During its life cycle, the CPU usage of db-server is limited to cores 3 and 4 (which in turn will be used exclusively by db-server) and the memory footprint is limited by the given amount. As `cgroups0` and `cgroups2` indicate, it's also possible to define the cgroups solely for CPU, memory, and other manageable system resources, or arrange them together in different combinations. The hardware resources here can be virtualized, too, as we detail later.

Docker further leverages the namespaces isolation feature, forcing processes to have separate namespaces for system resources, including (but not limited to) a process identifier (PID), interprocess communication (IPC), and network. The resources allocated to the application runtime inside a container can't be addressed by the other containers, and vice-versa. With the use of cgroups and namespaces isolation, a container runtime can readily be hosted. Docker has donated the implementation of these modules to the OCI project in a collection called *runC*, serving as the cornerstone to a standardized container runtime. Notably, both cgroups and namespaces isolation have been used independently and flexibly to achieve resource control[8] or isolation,[9] and many other container platforms are built based on these modules (in addition to OCI, see https://github.com/coreos/rkt and https://linuxcontainers.org), making the container techniques versatile.

To facilitate container creation and management, Docker has also designed and implemented the container (image) format. Each container runtime is created from an image predefined, which includes all the dependencies that the target application requires. Besides, the images can be stored in publicly accessible repositories and conveniently dis-

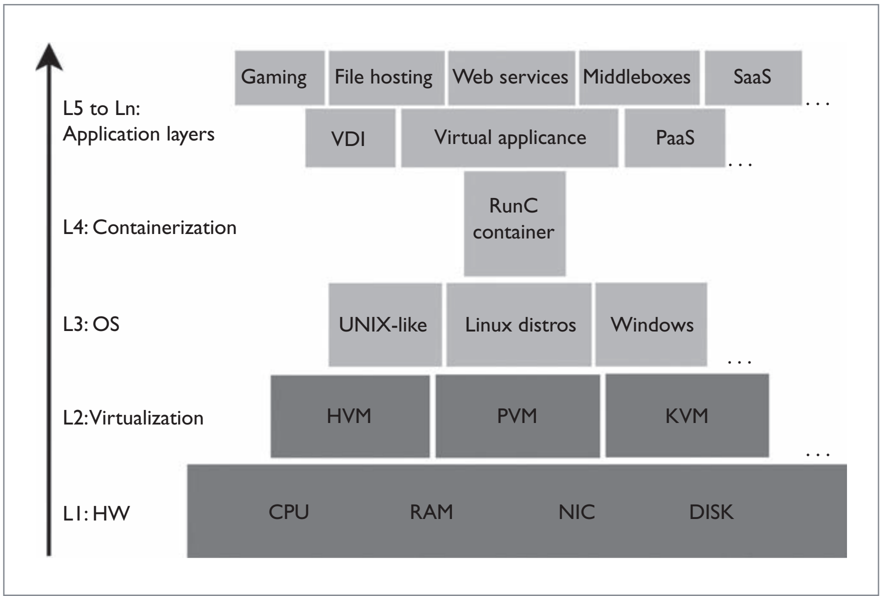


Figure 2. Cloud layered model: we envision the container (L4) as the narrow waist. It will bridge the underlying infrastructure layers and the application layers above.

tributed. Finally, Docker utilizes a layered file system to allow efficient sharing between container images, which significantly reduces the storage overhead.

Containerization versus Virtualization

To date, machine virtualization remains the most common way to manage hardware resources for cloud providers (for example, Xen[10] for Amazon EC2 public cloud), attracting significant standardization efforts (including OVF). Containers share such common design goals and features with VMs as resource isolation and imaging. Yet the new generation of containers represented by Docker, are built with important, distinct tenets.[6]

At the high level, containerization is upward-facing and application-driven, while virtualization is downward-facing and hardware-driven. Hypervisor-based virtualization (such as Xen) enables multiple users to create VMs that share the same physical hardware, where distinct OSs, ranging from the proprietary to open sourced, are hosted in an isolated fashion. Containerization permits only applications to be encapsulated in containers, which leads to greatly reduced deployment overhead and much higher instance density on a single machine. Unfortunately, it disallows a full OS stack to be run separately from the host OS, prohibiting a multi-OS setting.

At the low level, containerization leverages the host OS utilities to achieve resource encapsulation and management. Hypervisors, on the contrary, run directly on top of the hardware in the most-privileged mode, taking charge of accessing and managing the underlying hardware resources, akin to the role of an operating system kernel. The VMs and guest OS kernels now run in a less-privileged mode, such that any privileged system calls from guest OSs will be trapped to the hypervisor's kernel and executed in isolation. This process can be done in two ways: either through modifying the guest OS kernel and drivers to enforce privileged calls being sent to the hypervisor directly — which is known as *para-virtualization* (PV); or trapping those calls by special hardware extensions, known as *hardware-assisted virtualization* (HVM). As such, virtualization functions at the border of hardware and OS. It's able to provide strong performance isolation and security guarantees with the narrowed interface between VMs and hypervisors. Containerization, which sits in between the OS and applications, incurs lower overhead, but potentially introduces greater security vulnerabilities, such as namespace-agnostic system calls.[11]

In short, containerization isn't necessarily a replacement to virtualization; rather, these two complement each other, and are to be placed into a unified framework for cloud vendors and users.

The Container as the Narrow Waist

The renowned OSI model (ISO/IEC 7498-1) has helped delineate and standardize the Internet. As Figure 2 shows, we envision that a similar standardized reference model for the cloud infrastructure will emerge eventually. Such a model will be layered, with each layer emphasizing distinct infrastructural issues and functionalities, including resource multiplexing, isolation, orchestration, and application supports, respectively. An important design consideration for a layered model is the disposition of heterogeneity. For instance, the heterogeneity of network protocols converges at the IP layer, such that any transport layer protocols considering only the semantics of IP protocol will still be able to run on today's Internet infrastructure. Likewise, we must decide where a unified interface (that this, the "narrow waist") should be placed in the layered cloud infrastructure model. The key insight here is that a better part of the heterogeneity is introduced by the OS layer (L3) and application layers (L5 and above), where different applications rely on a variety of dependencies and OS supports. OSs themselves are also largely distinct from each other. Therefore, we envision container (L4) to become the narrow waist of this layered cloud model, bridging the underlying infrastructure layers and the application layers above.

The benefits of having such a unified/standardized interface are many-fold. Underneath, different host OSs can be utilized for the container, ranging from Unix-like ones to Windows-like proprietary ones, as long as they implement the container interface. Depending on user requirements, these host OSs are either placed on top of bare-metal non-virtualized machines or hypervisor-based VMs, where the first option gives the highest and close-to-native performance for demanding applications such as HPC workloads. Private clouds with managed user access could also benefit from this thinner model. With an additional hypervisor, a cluster yields stronger isolation and security guarantees, thereby being particularly appealing for public cloud providers. Moreover, the providers might opt for different hypervisors — ranging from full-virtualization to PV, and HVM to Kernel Virtual Machines (KVMs) — to meet various operational requirements.

On top of the interface, deploying and shipping an application runtime across distinct OSs and hardware architectures will become straightforward, because applications are agnostic to all of the underlying infrastructure details. Developers are spared from redundant and repetitive runtime environment configurations, system administrators are granted the ability of lightweight resource tracking and control, and cloud providers are allowed to support higher-level services such as platform as a service (PaaS) effortlessly. From the resource orchestration perspective, both the open source and proprietary cluster managers have been relying on the container as the basic scheduling unit.[12,13] In particular, the idea of two-level scheduling separates the cloud provider's concern in achieving efficient resource multiplexing[14] from the user's concern in attaining optimized workload locality, resource utilization, as well as cost savings. Further, the container interface will facilitate cloud services' transition beyond infrastructure as a service (IaaS), PaaS, and software as a service (SaaS), to highly modularized and distributed architectures such as microservice and "serverless" applications (see AWS Lambda; https://aws.amazon.com/lambda). Through defining a standard format of declaring container configuration and runtime, the promising OCI is a project that aims to establish the interface so that any host OS that implements it accordingly can support the container, bringing the aforementioned benefits to the cloud ecosystem. Earlier we detailed the key modules for implementing this interface, and it's worth noting here that the reference container implementation provided by OCI, *runC*, originates from Docker's `libcontainer` project.

Note that both virtualization and containerization are included in the reference model. When they're used together (for example, in a public cloud), with the underlying physical hardware and the OS in between, they form the hybrid-virtualization layers sitting at the bottom of the standard model (analogous to the link layers in the OSI model). This layering essentially places containers inside hypervisor-based VMs, which are then run on top of the underlying hardware; the rationale is that these two technologies, with distinct design goals and characteristics (as illustrated earlier), are in fact complementary to each other.

This hybrid layering is intuitive yet powerful, by which cloud users are allowed to orchestrate their resources provisioned without any assistance from the underlying infrastructure providers. As Figure 3 shows, the

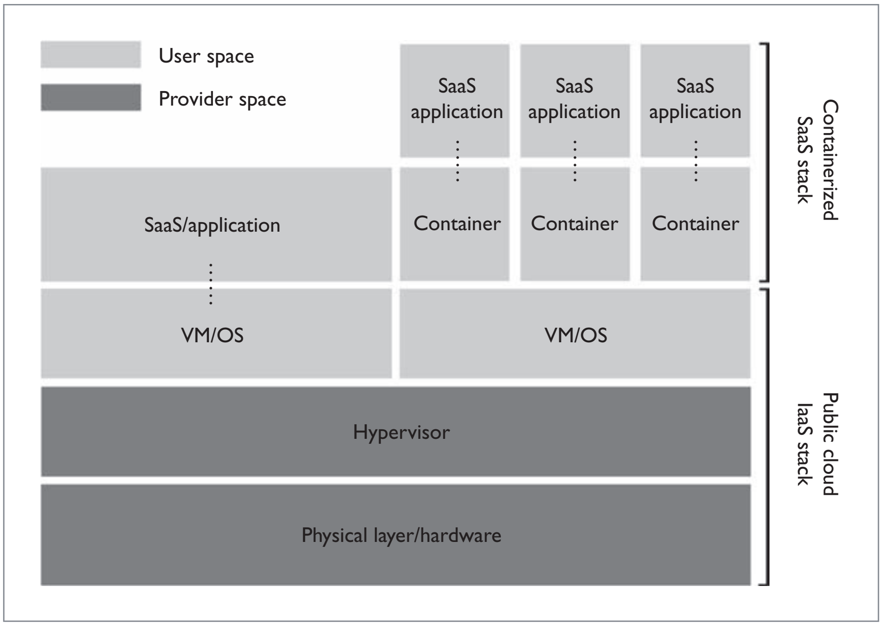


Figure 3. Hybrid virtualization layering in a public cloud. IaaS stands for infrastructure as a service, PaaS stands for platform as a service, and SaaS stands for software as a service.

Table 1. Performance of hybrid virtualization on a public Elastic Compute Cloud (EC2).

Resource type (benchmark)	Bare VM*	Hybrid VM
CPU (7z compression)	92.18 Mbytes/s	92.26 Mbytes/s
Memory (Sysbench, read)	10.84 Gbytes/s	10.85 Gbytes/s
Memory (Sysbench, write)	10.49 Gbytes/s	10.08 Gbytes/s
Disk (Bonnie++, rewrite)	119.95 Mbyte/s	118.22 Mbytes/s
Network (Iperf, TCP Send)	126.60 Mbytes/s	126.59 Mbytes/s
Network (Iperf, TCP Recv)	126.53 Mbytes/s	126.50 Mbytes/s

* VM = virtual machine.

cloud resource stack is, more often than not, separated into the user's and provider's space. For security and overall cloud performance, (public) providers don't let users freely launch any operations in their space, making resource consolidation and orchestration in the user space a headache (left half in the user space). In the hybrid layering (right half), the container adds another layer of abstraction in between the VM instance and applications, decoupling application-specific scheduling and VM scheduling. A direct use of this model is PaaS, which relies on the container to establish any service runtime environment effortlessly, and in the meantime the PaaS providers are exempted from handling the physical infrastructure. IaaS providers (such as Amazon ECS) can also benefit from this paradigm. Without involving redundant and unnecessary OS processes, the scaling out is more efficient for containers than VMs (see, for example, the Google Container Engine), which enables finer-grained billing. Noticeably, the hybrid layering is able to coexist with the virtualization-only solution, as Figure 3 indicates.

The additional layer might cause performance overhead and thus reduce the public cloud's cost-effectiveness. Previous works[11] have demonstrated that the container is able to achieve much-improved close-to-native performance, as compared to the hypervisor counterpart. However, the joint performance of the hypervisor and the container has yet to be explored. To quantitatively evaluate the overhead, we conducted experiments in the Amazon EC2 public cloud. Our metrics included CPU, memory, disk, and network I/O. To alleviate interference from the underlying cloud infrastructure, we repeated the experiments in four cloud instances, provisioned with the same resource offerings but at different times of the day. We containerized each EC2 instance with the newest version of the Docker container, and used microbenchmarks to measure and compare the performance inside and outside the container. As Table 1 shows, using the container introduces only negligible overhead with respect to CPU, memory, and disk and network I/O. (In our experiments, the CPU and memory-read performance are slightly higher for hybrid layering; but we conjecture that this is caused by the public cloud performance variance.) As we discussed, these resources are managed using cgroups through efficient function hooking (except for the network module, in which the Docker tags each network packet with a class identifier, and uses the *Linux traffic controller* to manage the packets from and to a particular container).

A container is a lightweight, flexible, and application-driven tool for fine-grained resource control and OS-level isolation. It augments the cloud infrastructure by addressing separate issues as compared to virtualization, which is a necessity when security is still one of the biggest concerns for cloud users.[1] The explored hybrid virtualization layering combines these two technologies with negligible performance overhead, setting the basis for the layered cloud reference model. In practice, we can place tens of or even hundreds of containers on the same physical or virtual machine.

Our experimental results indicate that using a container won't penalize the application performance when there's relatively low resource contention at the OS level. There is, however, contention at the hypervisor level from other tenants. It remains to be discovered — when we raise the density of co-located containers — exactly how the performance overhead will increase. Joining forces with major cloud vendors, the OCI project will further tackle these implementation and compatibility issues for the container. As today's applications are growing more distributed and cloud-based, it's expected that the container as the narrow-waist interface will improve the cloud infrastructure.

References

1. Harvard Business Review Analytic Services, "Cloud Computing Comes of Age," *Harvard Business Rev. Analytic Services Report*, tech. report, 2015; https://goo.gl/nZ4dIC.
2. S. Ortiz, "The Problem with Cloud-Computing Standardization," *Computer*, vol. 44, no. 7, 2011, pp. 13–16.
3. *Open Virtualization Format White Paper*, white paper, Distributed Management Task Force, v. 2.0.0a, document DSP2017, 3 July 2013; www.dmtf.org/sites/default/files/standards/documents/DSP2017_2.0.0a.pdf.
4. IEEE Cloud Computing Initiative, "Standards in Cloud Computing," IEEE Standards Assoc., 2016; http://cloudcomputing.ieee.org/standards.
5. R. Gardler, *Azure Container Service: Now and the Future*, blog, Microsoft, 29 Sept. 2015; https://goo.gl/QaDvQU.

6. S. Soltesz et al., "Container-Based Operating System Virtualization: A Scalable, High-Performance Alternative to Hypervisors," *Proc. European Conf. Computer Systems*, vol. 41, no. 3, 2007, pp. 275–287.
7. N. Peterson, *Windows Containers*, Microsoft, 11 Dec. 2015; https://goo.gl/YAsGmo.
8. D. Lo et al., "Heracles: Improving Resource Efficiency at Scale," *Proc. Int'l Symp. Computer Architecture*, 2015, pp. 450–462.
9. B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks," *Proc. Sigcomm Workshop on Hot Topics in Networks*, 2010, article no. 19.
10. P. Barham et al., "Xen and the Art of Virtualization," *Proc. Sigops*, vol. 37, no. 5, 2003, pp. 164–177.
11. W. Felter et al., "An Updated Performance Comparison of Virtual Machines and Linux Containers," *IBM Research Report*, tech. report RC25482, IBM Research Division, IBM, 21 July 2014.
12. B. Hindman et al., "Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center," *Proc. Usenix Conf. Networked Systems Design and Implementation*, 2011, pp. 295–308.
13. A. Verma et al., "Large-Scale Cluster Management at Google with Borg," *Proc. European Conf. Computer Systems*, 2015, article no. 18.
14. M. Armbrust et al., "A View of Cloud Computing," *Comm. ACM*, vol. 53, no. 4, 2010, pp. 50–58.

Silvery Fu is concurrently a master's student at Simon Fraser University and an undergraduate student in the dual degree program from Simon Fraser University and Zhejiang University. His interests focus on cloud computing topics surrounding virtualization, networking, and online gaming. Contact him at dif@sfu.ca.

Jiangchuan Liu is a university professor in the School of Computing Science, Simon Fraser University, and an NSERC E.W.R. Steacie Memorial Fellow. He's also the EMC-Endowed Visiting Chair Professor of Tsinghua University. His research interests include multimedia systems and networks, cloud computing, social networking, online gaming, Big Data, wireless sensor networks, and peer-to-peer networks. Liu has a PhD in computer science from the Hong Kong University of Science and Technology. He has served on the editorial boards of IEEE Transactions on Big Data, *IEEE Transactions on Multimedia, IEEE Communications Surveys and Tutorials, IEEE Access, IEEE Internet of Things Journal,* and *Computer Communications*. Contact him at jcliu@sfu.ca.

Xiaowen Chu is an associate professor in the Department of Computer Science, Hong Kong Baptist University. His research interests include distributed and parallel computing and wireless networks. Chu has a PhD in computer science from the Hong Kong University of Science and Technology. He's a senior member of IEEE and an associate editor of *IEEE Access*. Contact him at chxw@comp.hkbu.edu.hk.

Yueming Hu (corresponding author) is a professor of land information technology at South China Agricultural University. He's also the director of the Guangdong Province Key Laboratory of Land use and consolidation, and the chairman of the Guangdong Mapping and Geoinformation Industry Technology Innovation Alliance. Yueming has a PhD in soil geography from Zhejiang University. Contact him at ymhu@scau.edu.cn.

This article originally appeared in IEEE Internet Computing, *vol. 20, no. 2, 2016.*

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

Deciding When and How to Move HPC Jobs to the Cloud

Marco A.S. Netto, Renato L.F. Cunha, and Nicole Sultanum, IBM Research

Now used for high-performance computing applications, the cloud presents a challenge for users who must decide, based on efficiency and cost-effectiveness, when and how to run jobs on cloud-based resources versus when to use on-premise clusters. The authors propose a decision-support system to help make these determinations.

The cloud began as a platform to host Web applications but has since been used for many other types of programs, including those for high-performance computing (HPC). These applications have become an integral part of numerous domains including seismic research for oil and gas exploration, high-resolution solid and fluid mechanics, social-media analytics, and molecular dynamics. While HPC users often have access to on-premise computing clusters, these resources might be insufficient for their application executions, known as jobs, or might force their jobs to wait a long time in a queue.[1] Thus, HPC researchers are exploring the benefits of moving resource-intensive jobs, to the cloud.[2–4]

Renting HPC clusters in the cloud has recently become easier and less expensive. For instance, users can rent a 20-node cluster built with virtualized machines that have 64 Gbytes of RAM and 16 cores each for US$14 per hour. The same cluster using physical machines would rent for US$23 per hour. These prices drop another 10 percent for monthly rentals.

Organizations can rent cloud resources to augment their local computing capacity to meet increasing demand, creating a hybrid operation. However, this creates challenges such as how to decide which jobs should be moved to the cloud and when.

Here, we examine these challenges and describe a tool we're currently developing to help users determine whether their jobs should be run in on-premise or cloud-based clusters.

CHALLENGES FOR HPC CLOUD USERS

Moving HPC jobs to the cloud is a cultural, as well as technical, issue that affects users and IT infrastructure administrators. Users frequently act as if on-premise HPC resources are cost-free and thus don't always utilize them carefully. On the other hand, they appear more aware that the cloud is not free and must be accessed wisely. Users would benefit from a tool that helps them decide whether and how to use on-premise or cloud resources for various tasks.

 Published by the IEEE Computer Society

EDITOR SAN MURUGESAN
BRITE Professional Services;
san@computer.org

Such a tool would be of value, because although the cloud offers many advantages, it also presents challenges. For example, cloud use entails latency. Tightly coupled parallel applications require processors to communicate among themselves via a high-speed network. Without such a network, many parallel applications don't scale well, causing users to choose to work with on-premise resources.

A significant bottleneck occurs between the user infrastructure (including systems ranging from laptops to clusters) and the cloud. This can ruin the experience for users, who expect quick access to HPC cloud applications' output for purposes such as visualization and analysis.

UberCloud—an online community and marketplace where engineers and scientists discover, try out, and buy computing as a service—reports challenges that companies face when moving HPC workloads to the cloud.[5,6] The US Department of Energy's *Magellan Report on Cloud Computing for Science* contains analyses on running HPC and data-intensive applications in the cloud.[7]

A potential problem for users is estimating the cost of running HPC applications in the cloud. They generally don't know a priori how long their applications will have to run, as many programs might present irregular behaviors that make predicting execution times difficult.[8] Even when users try to estimate this, they frequently can't predict how many application instances will be required because they might need to make multiple computations with different input parameters.

In traditional HPC facilities, such as universities and research centers, users already struggle with estimating how much time they'll need to use on-premise resources. This is more complex in the cloud, in which users are charged based on application running time but the pricing models aren't necessarily linear or certain. For example, on an hourly subscription, they will be charged for a full hour of usage even if their task took only 20 minutes. In some cases, cloud providers will charge less for resources that they have the right to terminate at any time, which are suitable for fault-tolerant applications and services. Thus, estimating costs for the cloud is a daunting, yet crucial, task.

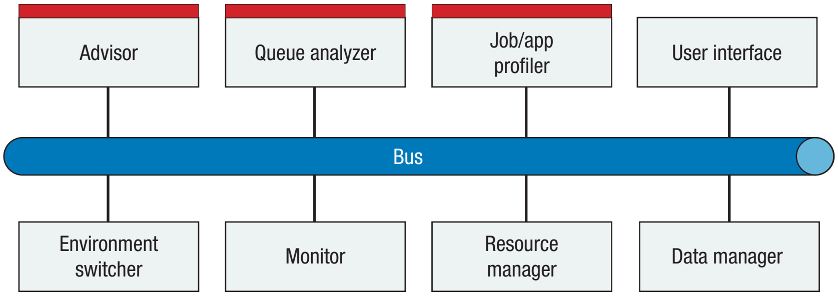


Figure 1. Decision-support system for running high-performance computing (HPC) applications. System components communicate with one another through a common bus. Via an interface, users input their job and business requirements and receive information about the cost of running their task utilizing the HPC cloud versus utilizing on-premise resources, and about which option makes more sense. The components with the red band are the most challenging to design and implement.

HPC DECISION-SUPPORT SYSTEM

Implementing a hybrid HPC cloud entails several challenges. A major, and largely overlooked one, is awareness of the potential cost of using the cloud, given various job-allocation scenarios. This is necessary for users to determine when running a job in the cloud makes economic sense. To help users, we're developing a decision-support system (DSS) designed to forecast the cost of running HPC applications in the cloud under several possible configurations, allowing for easier comparison among alternatives. The DSS also specifies levels of uncertainty about job-execution time and cost estimates caused by forecasting-model imperfections. Figure 1 shows the DSS components.

Advisor

This main DSS component helps users determine the least expensive way to execute jobs while still generating timely results. To accomplish this, the advisor receives information from other components about the time required to access resources, such as on-premise clusters' queue lengths and the way jobs would run in different computing-system environments and configurations.

The advisor doesn't try to tell users what the best job-allocation solution is because the criteria for that may be too unclear or subjective to express in computational terms. Instead, it looks at the information gathered from other components and provides easy-to-understand suggestions about where best to run jobs such as "if your job takes more than 4 hours, you

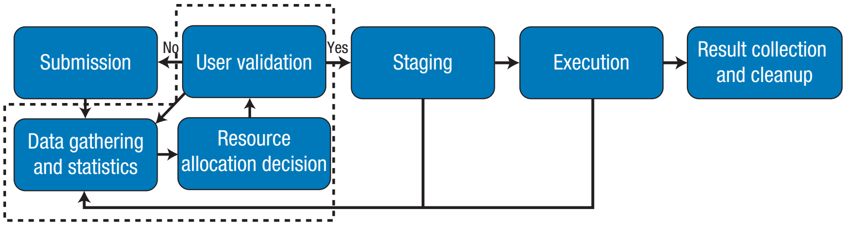


Figure 2. HPC decision-support system's job flow. The dashed line represents where the DSS's advisor, queue analyzer, and job/app profiler operate. If the users don't validate the system's job-allocation suggestions, they resubmit the job. If they do validate them, the job goes to staging, from which data and applications are transferred. Cloud resources are provisioned if the job is running in the cloud. Otherwise, the application is queued in the on-premise resource manager.

should run it in the on-premise cluster." Ultimately, the user is in charge of the process.

Queue analyzer

A major advantage of using cloud systems is resource availability, as utilizing on-premise clusters frequently entails long waits in queues. The queue analyzer predicts the wait times that various jobs would experience. The challenge is determining how various cluster-management policies affect prediction accuracy.

A few existing systems such as QBETS (queue bounds estimation from time series)[9] and the US National Science Foundation Extreme Science and Engineering Discovery Environment's Karnak[10] propose to tackle this challenge. We're developing our own queue analyzer based on these systems' techniques.

Job/app profiler

Benchmarks are necessary to determine the best way to use cloud resources: they help identify how a job would perform in different environments and configurations. However, extensive benchmark execution is neither timely nor economically sustainable. Thus, the profiler combines benchmarks with information obtained from other sources. For example, users could provide information on budgets and resource-access time limits; and historical data of past job executions could yield information on resource-access times, execution times, and the number of processors allocated.[11] This eliminates the need to execute benchmarks for resource-allocation decisions not of interest to users.

User interface

Sometimes neglected by the HPC community, the user interface (UI) is critical for providing good, clear time- and cost-management information. Developers could employ data visualization techniques to create meaningful, functional, and information-rich interfaces.

Environment switcher

Between job submission and execution, the on-premise cluster's queue status could change or users might decide to run tasks in another environment. In such cases, the system must be able to move jobs between environments. If the job hasn't started yet, the switcher interfaces with the resource manager to remove the job from or add it to the cluster queue. If the job is already running, the switcher relies on checkpointing to save the job's execution state in one environment and restore it in another.

Monitor

To provide the status of jobs being executed, the system should monitor itself. For example, if a job reaches a problematic state or the system deviates from its predicted behavior, the monitor could issue a notification.

Resource manager

To execute jobs in the cloud, the system must provision cloud resources with the necessary operating system and libraries. Our tool uses cloud-provider APIs, which contain functions to allocate, release, and configure cloud resources. These functions allow the integration of the cloud resources with an on-premise cluster-management system such as the Platform Load Sharing Facility (LSF), the Portable Batch System (PBS), the Terascale Open-Source Resource and Queue Manager (TORQUE), and the Simple Linux Utility for Resource Management (SLURM).

Data manager

Most nontrivial jobs must read input data and produce output data. Output information must be available for use after the system executes a job. Simply copying all data before job execution and then copying the new output information afterward might not be cost-effective if there is a lot of information, especially if the link between on-premise and cloud resources has low throughput. Instead, the system needs data synchronization to function efficiently.

Another potentially more cost-effective alternative would be placing the information that is likely to be used in an inexpensive cloud-based object storage service with high data-access rates for cloud instances.

CONTROL FLOW

Figure 2 shows our proposed system's control flow. Once the user submits a job, the system gathers user constraints—such as the deadline for job completion and the available budget—while also fetching queued data and estimated execution times from other components. If the job/app profiler doesn't already have information about the type of job being run, it might execute benchmarks, ask users to provide predictions, or take a conservative approach and provide an initial overestimate of execution times. In addition, the data manager calculates estimated data-transfer times—which influence job-placement decisions when the amount of information moving between on-premise and cloud resources is large—by using historical data from previous transfers stored in the data manager.

With all this information, the advisor calculates and ranks the cost of running the job in different environments,[1] presenting various options to the users, who then make a choice based on their own time- and cost-related priorities. After user validation, the system moves the job to the selected environment.

During execution, the monitor evaluates the job's running time and cost, and if these values rise above a user-defined threshold, the system triggers an alarm. This could lead to additional data gathering and calculation, causing the advisor to suggest alternative configurations in the execution environment.

In the short run, organizations must benchmark their frequently used applications to evaluate the cost benefit of migrating them to the cloud. They must also track the frequency with which applications are executed because this impacts decisions about whether and how to use cloud or on-premise clusters. For example, frequently utilized applications that demand a lot of resources should run in on-premise clusters to reduce cloud-related costs.

For hybrid environments, resource-allocation policies should carefully match jobs and environments. For instance, tightly coupled parallel applications or data-intensive applications should use on-premise resources because slow network connections would cause them to take too long to run in the cloud, which would increase expenses.

Several of these decision-making processes are still done manually or are scripted by experts because, in these cases, estimating execution times and the amount of resources required is complex and difficult. Thus, tools like the one we're developing are necessary to improve and automate more of these decisions.

HPC cloud use is an important topic of investigation. More efficient resource utilization could benefit organizations. And observing users' behavior could shed light on their hard-to-model, subjective criteria for job allocation. Systems could leverage this understanding to make job-configuration suggestions more relevant and helpful, enabling efficient and well-informed decision making. ■

REFERENCES

1. A. Marathe et al., "A Comparative Study of High-Performance Computing on the Cloud," *Proc. 22nd Int'l Symp. High-Performance Parallel and Distributed Computing* (HPDC 13), 2013, pp. 239–250.
2. A. Gupta et al., "The Who, What, Why and How of High Performance Computing Applications in the Cloud," *Proc. 5th IEEE Int'l Conf. Cloud Computing Technology and Science* (CloudCom 13), 2013, pp. 306–314.
3. M. AbdelBaky et al., "Enabling High-Performance Computing as a Service," *Computer*, vol. 45, no. 10, 2012, pp. 72–80.
4. C. Vecchiola, S. Pandey, and R. Buyya, "High-Performance Cloud Computing: A View of Scientific Applications," *Proc. 10th Int'l Symp. Pervasive Systems, Algorithms, and Networks* (ISPAN 09), 2009, pp. 4–16.
5. W. Gentzsch and B. Yenier, *The Uber-Cloud HPC Experiment: Compendium of Case Studies*, tech. report, Tabor Communications, 2013.
6. W. Gentzsch and B. Yenier, The *UberCloud Experiment: Tech. Comp. in the Cloud—2nd Compendium of Case Studies*, tech. report, Tabor Communications, 2014.
7. M. Leads et al., *The Magellan Report on Cloud Computing for Science*, tech. report, US Department of Energy-Office of Science-Office of Advanced Scientific Computing Research, 2011.
8. C.B. Lee and A. Snavely, "On the User-Scheduler Dialogue: Studies of User-Provided Runtime Estimates and Utility Functions," *Int'l J. High Performance Computing Applications*, vol. 20, no. 4, 2006, pp. 495–506.
9. D. Nurmi, J. Brevik, and R. Wolski, "QBETS: Queue Bounds Estimation from Time Series," *Proc. 13th Int'l Workshop Job Scheduling Strategies for Parallel Processing* (JSSPP 07), 2007, pp. 76–101.
10. W. Smith, "A Service for Queue Prediction and Job Statistics," *Proc. Gateway Computing Environments Workshop* (GCE 10), 2010; doi: 10.1109/GCE.2010.5676119.
11. D. Tsafrir, Y. Etsion, and D. Feitelson, "Backfilling Using System-Generated Predictions Rather than User Runtime Estimates," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 6, 2007, pp. 789–803.

MARCO A.S. NETTO is a researcher and manager of the Industrial Cloud Technologies Group at IBM Research–Brazil and an IBM Master Inventor. Contact him at mstelmar@br.ibm.com.

RENATO L.F. CUNHA is a researcher in IBM Research–Brazil's Industrial Cloud Technologies Group. Contact him at renatoc@br.ibm.com.

NICOLE SULTANUM worked at IBM Research–Brazil and is now a University of Toronto PhD candidate. Contact her at nicolebs@cs.toronto.edu.

This article originally appeared in Computer, *vol. 48, no. 11, 2015.*

Standards for Hybrid Clouds

IN HYBRID CLOUD SETTINGS, APPLICATIONS REQUIRE CAREFUL PARTITIONING AS TO WHICH COMPONENTS WILL OPERATE ON WHICH SIDE OF AN ORGANIZATION'S BOUNDARY. The control and deployment settings available within the internal private part of a hybrid cloud might not match those available from a given public cloud provider, especially if that provider only supports access through specific proprietary interfaces. This column will explore emerging and established standards applicable to these settings.

In the most recent instance of this column, I gave examples of application programming interfaces (APIs) in example scenarios related to hardware infrastructure control.[1] These examples began with adding control interfaces to individual machines or small collections of devices in a home or small office setting and ranged up to consider APIs used in large-scale datacenters. The next step is to consider hybrid settings.

Hybrid cloud scenarios arise when an organization decides for cost, security, or other considerations to keep some portion of its operations physically housed on site, while making use of resources provided outside of the organization for other features. The interaction between these two settings creates the conditions for use of a hybrid cloud approach.

ALAN SILL

Texas Tech University,
alan.sill@standards-now.org

Infrastructure Boundaries in Clouds

Software that needs to operate across boundaries in hybrid cloud settings can do so more easily when the interfaces between these components operate the same way on each side of the boundary. No matter how well-designed, controllable, understandable, or easy to use the software features of a particular public cloud provider might be, it's nonetheless a distinct disadvantage if the interfaces and APIs of that provider don't match those used by internal components on the internal, private side of a hybrid cloud deployment.

Such interfaces therefore work best when they can be deployed in a standardized, well-documented, preferably machine-readable, self-documenting way, and are easily adaptable to both sides of the hybrid cloud boundary. Of course, a cloud infrastructure has components that span both hardware and software. One goal of most cloud deployments is to disassociate these settings from each other, and therefore allow deployment and scaling of software features in ways that are as independent as possible from details of the underlying hardware.

Such dissociation is never complete. Considerations such as cost, workflow, and optimal fit of software tools to their operating environments enter into choices of deployment details such as instance size, service abstraction, and virtualization paradigms such as containers versus virtual machines. Most public cloud providers offer a wide range of user-selectable choices for these deployment details.

Because of this range, there's often a greater degree of freedom for a software designer to design around, ignore, suppress, or simply specify details of the underlying hardware configuration on which their software will run in public cloud settings than in the context of private or hybrid clouds.

The reverse of this argument is also true. In private cloud settings, the physical infrastructure is more likely to be designed to match the exact needs of a particular task or business problem, whereas in public clouds, the workflow itself generally has to be

 Published by the IEEE Computer Society 2469-7087/16/$33.00 © 2016 IEEE

adapted to make the best use of the tools and interfaces made available by the external provider.

The most obvious feature of hybrid cloud settings is therefore that control and orchestration have to work across multiple underlying infrastructure variations, necessitating the choice of interfaces that can connect these different portions of the application or system. This feature places a premium on use of tools that are designed to work across boundaries and can function in multiple settings.

Hybrid Cloud Frameworks

These factors have led to a renewed emphasis on infrastructure frameworks that can be deployed on a variety of underlying infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) provider software stacks. The technical term associated with this process is standardization. Observers in software development and business analysis arenas sometimes shy away from using this term for fear of applying it too early, but standardization is now the explicit stated goal of some of the most modern and up-to-date activities in cloud computing.

Several software frameworks that started as individual projects are now moving in this direction. Examples include Kubernetes, Docker, and Mesos, and have led to standardization efforts such as the Open Container Initiative (OCI, www.opencontainers.org), which was created to serve as an industry forum for standardization of container-oriented virtualization.

These efforts have already borne fruit. The recent version 1.0 release of the rkt application container framework (www.github.com/coreos/rkt) by CoreOS, for example, explicitly includes standards compatibility, stating that "rkt implements the appc specification, supports the Container Networking Interface specification, and can also run Docker images."

Another notable example of successful industry-based standardization is provided by the Cloud Native Computing Foundation (CNCF, www.cncf.io), which focuses on technologies that are designed to support cloud-based workflow patterns and scale well in cloud settings. A newly formed project, the Open API Initiative (OAI, www.openapis.org), which aims to extend previous efforts of the Swagger community in the area of API description and discovery, also looks promising.

The emergence of these community-based tool development and standardization efforts is the natural reaction to the complexity that results from having multiple available approaches to a given problem. Such complexity often creates conditions such as the examples I've given, in which overall simplification can be achieved by introducing a common, well-specified unifying approach. I've discussed methods to identify and target areas in which conditions are suited for standardization in previous columns.[2]

Cloud IaaS and PaaS Standards

This column has covered formal standards related to cloud IaaS and PaaS before.[3] Some of these, such as the Open Cloud Computing Interface (OCCI) from the Open Grid Forum (OGF, www.ogf.org), the Topology and Orchestration Standard for Cloud Applications (TOSCA) from the Organisation for Advancement of Structured Information Systems (OASIS, www.oasis-open.org), and the Cloud Data Management Interface (CDMI) from the Storage Networking Industry Association (SNIA, www.snia.org), have since developed strong followings with adoption in several open source software projects. Others, such as the Cloud Infrastructure Management Interface (CIMI) from the Distributed Management Task Force (DMTF, http://www.dmtf.org) continue to make progress in their formal definition and documentation.

Each of these standards aims to address a different aspect of the complexity and confusion that currently characterizes cloud IaaS and PaaS interfaces. An extremely diverse situation currently holds among cloud providers, with each having its own tooling, deployment methods, and infrastructure control software. As discussed previously, whenever the diversity of interfaces complicates rather than simplifies programming tasks, we can expect standard protocols, APIs, and related interface tooling to emerge to simplify the deployment and operation of applications, and this condition also holds in hybrid cloud settings.

Some API designers even build in deliberately designed restrictions to prevent discovery of, or direct access to, internal features. Such restrictions can limit the usefulness of even the most powerful cloud provider services, unless the deployment also provides support for well-specified client interfaces to enable remote use by other portions of an extended system. Under such conditions, we can expect interest in cross-cutting standards that work across boundaries such as those mentioned to grow, and where successful, to flourish.

Software Tools for Hybrid Workflows

The temptation of any programmer when presented with a situation in which different parts of a problem are best solved by different individual tools is to write a script, program, or workflow tool that combines the various steps in an organized way. A related condition can also be obtained when there is some reason to build choices based on factors related to economics or speed into the solution, allowing different providers to be used or different resources to be drawn on for individual steps in the solution.

> Circumstances have led the interfaces used by various cloud providers to remain separate and mutually incompatible for a very long time.

In hybrid cloud settings, the characteristic that unifies the choices to be made in a given workflow is the ability to use a common interface as the point of selection among internal options. Equivalently, the output from one step of the operation should ideally be presented in a way that feeds naturally into the next step of the workflow. Such considerations lead naturally to standardization of input and output formats and to the need for tools to adapt between interfaces where cloud provider tooling isn't yet standardized and interoperable.

This isn't a new situation, but circumstances have led the interfaces used by various cloud providers to remain separate and mutually incompatible for a very long time. As a result, several libraries and, more recently, multifunctional workflow tools have emerged to bridge over the differences between the APIs of different cloud providers. Examples include libcloud (libcloud.apache.org), deltacloud (deltacloud.apache.org), fog (www.fog.io), and the Spinnaker framework (www.spinnaker.io), recently open-sourced by Google.

Libcloud is a Python module that provides a layer of abstraction covering many different cloud provider APIs, allowing programmers to write applications that work across multiple providers. Libcloud provides an application binary interface that allows cloud interfaces to be treated as drivers, rather than forcing the code to deal directly with each cloud provider's API. To use libcloud in cloud software, the developer must include the Python module directly in the application's code base rather than treat it as a remote procedure. Libcloud has progressed to a 1.0.0 preview release and remains under active development. It has also evolved to include support for block and object storage, load balancer, and domain name services.

Deltacloud was an Apache project similar in intent to libcloud, but written in a different language (Ruby) and more limited in implementation options. It included an implementation of the CIMI standard as well as Amazon Elastic Compute Cloud (EC2) and its own internal API, but otherwise didn't progress toward community adoption. The project hasn't seen significant activity in over two years, and has probably been abandoned. Although not directly related, the fog Ruby gem has emerged with similar functionality and support for a much larger range of cloud providers.

Ultimately, whether or not they're technically successful, such multiprovider "Swiss Army knife" adapter toolkits will obviously depend on being kept up to date with changes to the underlying APIs of the different cloud providers, and hence differ in intent and function from true API standards. Whether or not they can be used in a particular setting highly depends on the situation in which they're deployed, and on the willingness of the package providers and users to adapt to changes to the provider APIs.

Spinnaker differs from these other adapter toolkits in that it aims at a broader set of problems in cloud deployment that are generally referred to as continuous delivery. Instead of just trying to adapt

This article originally appeared in
IEEE Cloud Computing*, vol. 3, no. 1, 2016.*

an API call for infrastructure control or services to the APIs of different providers, Spinnaker includes many other aspects of software integration, delivery, and deployment along with cluster management in its feature set.

Although it includes a set of adapters for different cloud providers, Spinnaker also incorporates workflow concepts such as pipelines, stages, scheduling, and triggers for various steps of cloud software management. It also includes interfaces to code repositories, continuous integration engines, image management, and an internal orchestration engine to accomplish a wide variety of tasks that are important in maintaining a cloud software deployment and delivery. Each of these stages can in principle be hosted internally within an organization or on one of several supported cloud providers, making Spinnaker attractive for use in hybrid cloud settings.

As standards emerge from any of the previously discussed efforts, it would seem natural to see them included in such multifunctional tools, and eventually (if successful) to supplant and replace differences among the different cloud providers. It's easy to see some of the container standardization efforts mentioned here, for example, as candidates to be deployed within and among these tools.

AS ALWAYS, THIS DISCUSSION ONLY REPRESENTS MY OWN OPINION DERIVED FROM WORKING WITH THE PROJECTS, STANDARDS, AND SOFTWARE PRODUCTS MENTIONED. I'm interested in hearing your opinion and experience, and I'm sure other readers of the magazine would also appreciate such input.

Please respond with your opinions on this topic or on those explored in previous columns. Let us know what you think, and please also include any news you think the community should know about the general areas of cloud standards, compliance, or related topics. We're always open to article submissions. I'm happy to review ideas for such submissions, or for proposed guest columns. I can be reached for this purpose at alan.sill@standards-now.org.

References

1. A. Sill, "When to Use Standards-Based APIs (Part 2)," *IEEE Cloud Computing*, vol. 2, no. 6, 2015, pp. 80–84; doi:10.1109/MCC.2015.120.
2. A. Sill, "Socioeconomics of Cloud Standards," *IEEE Cloud Computing*, vol. 2, no. 3, 2015, pp. 8–11, doi:10.1109/MCC.2015.59.
3. A. Sill, "Cloud Standards News and Updates," *IEEE Cloud Computing*, vol. 2, no. 1, 2015, pp. 72–75, doi:10.1109/MCC.2015.4.

ALAN SILL *is interim senior director at the High Performance Computing Center at Texas Tech University, where he's also site director for the US National Science Foundation Cloud and Autonomic Computing Center and adjunct professor of physics. Sill has a PhD in particle physics from American University. He is an active member of IEEE, the Distributed Management Task Force, and the TeleManagement Forum, and he serves as president for the Open Grid Forum. For further details, visit http://cac.ttu.edu or contact him at alan.sill@standards-now.org.*


The Promise of Edge Computing

Weisong Shi, Wayne State University
Schahram Dustdar, TU Wien

The success of the Internet of Things and rich cloud services have helped create the need for edge computing, in which data processing occurs in part at the network edge, rather than completely in the cloud. Edge computing could address concerns such as latency, mobile devices' limited battery life, bandwidth costs, security, and privacy.

The term Internet of Things (IoT) was first introduced to the technology community in 1999 in reference to automated supply-chain management.[1] The concept of enabling a computer to sense information without human intervention was then applied to other fields such as healthcare, home technology, environmental engineering, and transportation.

With IoT implementation now becoming more widespread, we're entering the post-cloud era, in which devices will generate a lot of data at the end of the network and many applications will be deployed at the edge to process the information. Cisco Systems predicts that an estimated 50 billion devices will connect to the Internet by 2020.[2] Some of the applications they run might require very short response times, some might involve private data, and some might produce huge quantities of data. Cloud computing can't support these IoT applications. *Edge computing*, on the other hand, can do so and will promote many new IoT applications.

WHY DO WE NEED EDGE COMPUTING?

Edge computing will become important for several reasons.

Cloud services

Moving all computing tasks to the cloud has been an efficient way to process data because there's more computing power in the cloud than in the devices at the network edge. However, although data-processing speeds have risen rapidly, the bandwidth of the networks that carry data to and from the cloud hasn't increased appreciably. Thus, with edge devices generating more data, the network is becoming cloud computing's bottleneck.

As an example, cameras in an autonomous vehicle capture a huge amount of video data, which the system must process in real time to yield good driving decisions. If the vehicle must send the data to the cloud for processing, the response time would be too long. And a large number

Published by the IEEE Computer Society 2469-7087/16/$33.00 © 2016 IEEE

of autonomous vehicles in one area would further strain network bandwidth and reliability.

Processing data at the network edge would yield shorter response times, more efficient processing, and less pressure on the network. Recent work on micro-datacenters (mDCs; small, modular datacenters designed to optimize networked devices' performance via the cloud)[3] and cloudlets (small cloud datacenters at the Internet's edge designed to work with mobile devices)[4] has studied this.

The IoT

Billions of electrical devices—as well as other devices such as air-quality sensors, LED bars, and streetlights—will become part of the IoT and will produce, as well as consume, data. Conventional cloud computing won't be efficient enough to handle the sheer volume of data they'll generate. For example, the Cisco Global Cloud Index estimates that by 2019, people, machines, and things will produce 500 zettabytes (a zettabyte is 10^{21} bytes) of data but that global datacenter IP network traffic will reach only 10.4 zettabytes.[5]

Typically, data producers generate raw information and transfer it to the cloud, and data consumers send requests for information to the cloud. However, this structure won't work with the IoT because of the large data volumes involved.

Using cloud computing with the IoT will also raise concerns about the privacy of transferred data. In addition, most of the IoT's end nodes are power constrained. Offloading some computing tasks to the network edge could be more energy efficient. The OpenFog Consortium's proposed fog computing paradigm[6]—an infrastructure in which some application services are handled by devices at the network edge and some by a cloud-based datacenter—would enable this.

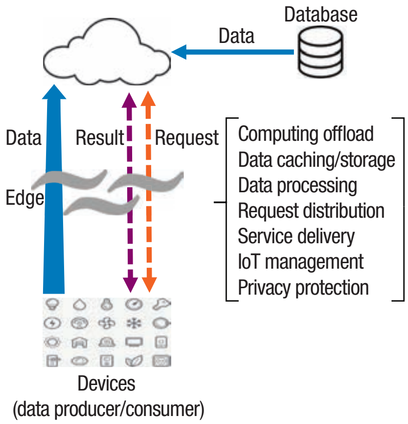


Figure 1. In edge computing, the cloud collects data from existing databases, as has been done traditionally, and also from end devices such as sensors and mobile phones. The devices act as both data consumers and data producers. Thus, requests between end devices and the cloud are bidirectional, instead of just from end devices to the cloud as in the past. Nodes at the network edge perform many computing tasks—including data processing, caching, device management, and privacy protection—to reduce traffic from devices to the cloud. IoT: Internet of Things.

From data consumer to data producer

In cloud computing, devices such as mobile phones at the network edge traditionally only consume data, such as by enabling a user to watch a video. Now, though, users are also producing data with their mobile devices, such as by uploading posts and photos to social-networking sites.

This change requires more functionality at the network edge. For example, many people share images or videos via a cloud-based social-networking service such as Facebook, Twitter, or Instagram. However, uploading a large image or video clip could require a lot of bandwidth. In this case, a device at the network edge could reduce the video clip's resolution, and thus its size, at the edge before uploading it to the cloud. In another example, properly equipped edge devices could better protect privacy by processing sensitive data collected by wearable personal health devices, rather than uploading the data over a vulnerable network to the cloud.

WHAT IS EDGE COMPUTING?

Edge computing, shown in Figure 1, refers to the enabling technologies that allow computation to be performed at the network edge so that computing happens near data sources. It works on both downstream data on behalf of cloud services and upstream data on behalf of IoT services. An edge device is any computing or networking resource residing between data sources and cloud-based datacenters. For example, an edge device could be a smartphone sitting between body sensors and the cloud, or an mDC or cloudlet between a mobile device and the cloud.

In edge computing, the end device not only consumes data but also produces data. And at the network edge, devices not only request services and information from the cloud but also handle computing tasks—including processing, storage, caching, and load balancing—on data sent to and from the cloud. The edge must be designed well enough to handle such tasks efficiently, reliably, securely, and with privacy in mind. It thus must support requirements such as differentiation, extensibility, isolation, and reliability.

Edge computing could yield many benefits. For example, researchers have shown that using cloudlets to offload computing tasks for wearable cognitive-assistance systems improves response times by between 80 and 200 ms[4] and reduces energy consumption by 30 to 40 percent. CloneCloud technology reduces response times and power usage by 95 percent for tested applications, in part via edge computing.[7]

Edge-computing practitioners should be aware of several issues, including system reliability. Also, energy-constrained edge devices could fail because of short battery life or inadequate wireless communications. In these cases, developers should still enable the system to provide its basic functions.

Security and privacy are additional concerns. On one hand, edge computing better protects data because processing occurs closer to the source than in cloud computing. However, supporting security and privacy is more difficult in edge computing due to network topology, the many inexpensive personal mobile devices in the system, and sensor unreliability.

IMPLEMENTATION EXAMPLES

Two examples demonstrate practical edge-computing implementations.

Online shopping

Online shopping could benefit from edge computing. For example, a customer might make frequent shopping-cart changes. Traditionally, these changes occur in the cloud, which then updates the shopping-cart view on the user's device via the network. This process might take a long time depending on network speed and server loads. Delays might be even longer for mobile devices because of wireless networks' relatively low bandwidth.

Cloud computing itself could also cause latency, which diminishes the user experience. Avoiding this is important because shopping with mobile devices is becoming increasingly popular. Caching shopping-cart data at the edge and offloading shopping-cart updates from cloud servers to edge nodes dramatically reduces latency. Numerous research projects have addressed how this type of offloading enhances performance and reduces energy consumption in a mobile-cloud environment.[7–10]

Data at the edge node could subsequently be synchronized with the cloud in the background.

Finding a missing child

Edge computing could help officials find missing children. Today, cameras deployed in public areas in cities—as well as cameras in some vehicles—could capture a missing child's image. However, this frequently isn't leveraged because the camera data usually isn't uploaded to the cloud due to privacy concerns or the cost of transferring the information. Even if the images are in the cloud, accessing and searching such a huge quantity of data could take a long time, which isn't acceptable when looking for a missing child.

With edge computing, the data could be pushed to the many edge devices in a target area. They could search the data they receive and report the findings to the cloud, yielding the results much faster than using only cloud computing.

OPEN ISSUES

Edge computing needs killer apps to reach its potential. Several application domains are worth exploring such as disaster response and management, body cameras for police officers, smart vehicles, and connected health systems.

Programmability

In edge computing, as several research projects have demonstrated, programmers must partition the functions of their applications between the edge and the cloud.[7–9] Most early efforts in this area were done manually and carefully tuned, which isn't scalable or extensible. Thus, easy-to-use programming frameworks and tools are required.

Naming

There are many edge devices and applications. Currently, however, there's no efficient, standardized naming system for edge-computing devices and applications so that they can be easily found. Thus, edge practitioners usually must learn many communication and network protocols to communicate with their systems' heterogeneous elements. Edge computing needs a naming scheme that will handle device mobility, highly dynamic network topology, and privacy and security protection, while also enabling scalability.

Privacy and security

The network edge presents security and privacy challenges. For example, a hacker could learn a lot by reading data going to and from a smart-home system. By capturing electricity or water usage, the hacker could easily determine if the house is probably vacant and thus vulnerable to burglary. A lack of efficient tools is one of the challenges to protecting data security and privacy at the network edge.

Edge computing could scale from a single person to a smart home to even an entire city. Given that a city with 1 million people will produce an estimated 180 petabytes of data per day by 2019,[5] the benefits could be enormous.

However, to realize this vision, the systems, network, and application communities must work together, joined by the many groups that could benefit from the technology such as those in environmental and public health, law enforcement, fire protection, and utility services.

In the past few years, this process has begun. For example, proponents formed the OpenFog Consortium (www.openfogconsortium.org) in November 2015 to promote an ecosystem to accelerate the adoption of open fog computing by bringing together companies, universities, and individual researchers. In addition, new conferences are planned, such as the IEEE/ACM Symposium on Edge Computing (SEC), to be held in October 2016; and the Mobile Edge Computing Congress (MEC), to be held in September 2016. If this trend continues, edge computing will be on its way to fulfilling its promise. C

REFERENCES

1. K. Ashton, "That Internet of Things Thing," *RFID J.*, vol. 22, no. 7, 2009, pp. 97–114.
2. *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*, white paper, Cisco Systems, Apr. 2011.
3. A. Greenberg et al., "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM Computer Communication Rev.*, vol. 39, no. 1, 2008, pp. 68–73.
4. M. Satyanarayanan et al, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, 2009, pp. 14–23.
5. *Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 White Paper*, white paper, Cisco Systems, 2014.
6. F. Bonomi et al., "Fog Computing and Its Role in the Internet of Things," *Proc. 1st Ed. MCC Workshop Mobile Cloud Computing* (MCC 12), 2012, pp. 13–16.
7. B.-G. Chun et al., "CloneCloud: Elastic Execution between Mobile Device and Cloud," *Proc. 6th Conf. Computer Systems* (EuroSys 11), 2011, pp. 301–314.
8. A. Rudenko et al., "Saving Portable Computer Battery Power through Remote Process Execution," *ACM SIGMOBILE Mobile Computing and Communications Rev.*, vol. 2, no. 1, 1998, pp. 19–26.
9. G. C. Hunt and M.L. Scott, "The Coign Automatic Distributed Partitioning System," *Proc. 3rd Symp. Operating Systems Design and Implementation*, 1999, pp. 187–200.
10. S. Kosta et al., "ThinkAir: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading," *Proc. 31st IEEE Int'l Conf. Computer Comm.* (Infocom 12), 2012, pp. 945–953.

WEISONG SHI is a professor of computer science at Wayne State University, where he heads the Data Science Initiative. Contact him at weisong@wayne.edu.

SCHAHRAM DUSTDAR is a professor of computer science at TU Wien (Vienna Technical University), where he heads the Distributed Systems Group. Contact him at dustdar@dsg.tuwien.ac.at.

This article originally appeared in Computer, *vol. 49, no. 5, 2016.*

Cloud-Based AI for Pervasive Applications

Albrecht Schmidt, *University of Stuttgart*

AI, computer vision, text analytics, and speech recognition are becoming more important for creating compelling pervasive computing applications. User expectations will only increase when it comes to what they expect from their environment and their applications, given their experiences with mobile devices (including Google Now, Microsoft Cortana, and Apple Siri) and given popular competitions between AI and humans (such as the Jeopardy win by IBM Watson). Also, as AI and computer vision advance, they enable exciting new applications—especially in ubiquitous computing. Here, I look at cloud-based products that offer APIs that let developers include AI in their prototypes and products.

ENCAPSULATING AI

Over 10 years ago, when talking with a computer vision researcher, I was surprised to learn about some of the problems the computer vision community had already solved. The disappointment came when I wanted to build ubiquitous computing applications that could rely on these "solved problems." The effort to include an algorithm, described in a research paper, into an application took significant effort (and, in many cases, required help from computer vision experts). The available libraries (early versions of openCV) required a solid understanding of the domain, and often the time invested into implementing test software resulted in disappointment, because the algorithms didn't perform well on the specific types of images used in the application.

Over the past few years, the quality of libraries and toolkits—including the Waikato Environment for Knowledge Analysis (WEKA) and openCV—has massively improved, lowering the hurdle to do machine learning and computer vision. With the cloud-based services highlighted in this article, the bar has been lowered again.

Several cloud-based services have become available that promise to encapsulate and hide the hard part of AI, computer vision, text analytics, and speech recognition.

Several cloud-based services have become available that promise to encapsulate and hide the hard part of AI, computer vision, text analytics, and speech recognition and that offer simple-to-use APIs. Including sophisticated AI functionality in applications thus becomes easy. The basic approach is to transfer data (text, images, and sounds) to the server, together with a set of parameters; the calculations are then executed on the server, and the results are sent back to the client. This approach has few requirements in terms of the processing and storage needed on the mobile or embedded devices that might run the applications. Using this cloud-based centralized approach benefits not only the application developer but also the company running the service, which receives numerous data samples.

TRY BEFORE YOU BUY (OR INVEST TIME)

On the one hand, using someone's library or service is appealing, because it promises functionality with less effort than implementing the functionality yourself. On the other hand, the library or service might not work as expected, meaning the time invested was wasted.

Building trust into a library or service is key to increasing uptake. With traditional software libraries and also with services, you often must implement a simple test program—or at least adapt the demo program provided—to see if they are suitable. Here, the cloud-based services offer new ways for assessing the suitability, and Project Oxford, IBM's AlchemyAPI for AI, and the Open Calais API are exemplary. Developers can try them out with their own data and images using the real interface before even starting with an implementation. In this way, they can see online if the services work as expected, so the initial time investment is small. Figure 1 shows a screenshot of the live demo of the Emotion API. The Web-based user interface shows visually what was detected, and it provides the raw data that will be received.

Published by the IEEE Computer Society 2469-7087/16/$33.00

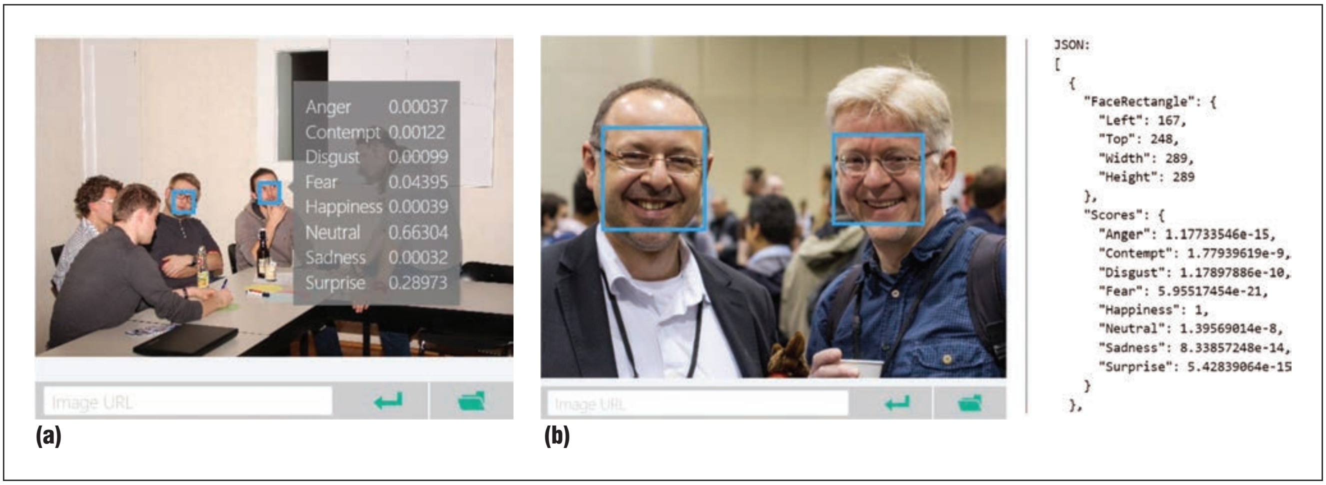


Figure 1. Two examples of trying emotion recognition with the online interface of Project Oxford. To try the service out, you can just upload a photo and see the results. The service (a) overlays the detected values over the photo and (b) also provides the results as a JavaScript Object Notation (JSON) object. (Screenshot from www.projectoxford.ai.)

SIMPLICITY FOR THE DEVELOPER

The Web-based APIs I discuss here are powerful but designed to be simple to use for developers. However, they should also be easily usable for even novice Web programmers and app developers. A common approach is that the service is called with a basic Web request that transmits the data and sets the parameters. The server responds with the results in a format that is human readable and easy to parse, such as JavaScript Object Notation (JSON) or XML.

Calling the Web-based services requires a key. Using this key, the services provider can track and potentially bill for the API calls made. For experimenting, free keys with a limited number of requests can be obtained (for example, IBM offers 1,000 API events per day for free, Microsoft offers 5,000—10,000 API calls per month, and Thomson Reuters offers 5,000 API calls per day) and for productive use, payment options are available (for example, IBM bills from about 0,005€ to 0,00015€ per API event, depending on the number of calls made).

Figure 2. Services (all in the beta stage) provided in the Project Oxford. (Source: www.projectoxford.ai; used with permission.)

For many applications in pervasive computing scenarios, this might be an economically interesting alternative to natively implementing the functions or to running your own server. When the functions are called continuously, this approach might not make much sense—from an economic or performance perspective. However, if calls are linked to user actions (such as a smart coffee maker that bills coffee automatically to

INNOVATIONS IN UBICOMP PRODUCTS

```
var request = require('request');
var imgAsJSONObject = { "Url" : "http://www.hcilab.org/myPhoto01.jpg" };
request({
        url: "https://api.projectoxford.ai/emotion/v1.0/recognize",
        method: "POST",
        headers: { "Ocp-Apim-Subscription-Key" : "****************" },
        json: true,
        body: imgAsJSONObject
}, function (error, response){
        console.log(response.body);
});
```

(a)

```
[ { faceRectangle: { height: 116, left: 135, top: 81, width: 116 },
    scores: {
        "anger": 0.0020877,
        "contempt": 0.0001845,
        "disgust": 0.0000081,
        "fear": 0.0100902,
        "happiness": 0.9529958,
        "neutral": 0.00000412,
        "sadness": 0.0005182,
        "surprise": 0.1081329 } } ]
```

(b)

Figure 3. A JavaScript implementation that uses functionality of the Emotion API: (a) a small JavaScript example program and (b) the response received for the image shown in Figure 1b. The delay for the API call was in my test between 1 and 2 seconds.

```
POST vision/v1/ocr?language=unk&detectOrientation=true
Content-Type: application/json
Host: api.projectoxford.ai
Content-Length: 48
Ocp-Apim-Subscription-Key: ***********

{ "Url": "http://www.hcilab.org/photoWithText.jpg" }
```

Figure 4. The HTTP Post request for the optical character recognition API. A valid subscription key can be obtained from the website of the service.

users it sees), this approach is interesting, because the effort for the implementation is minimal. For example, using the project Oxford Face-Recognition API and implementation would be straightforward, and neither cost nor delay would matter.

PROJECT OXFORD

Some time back, I came across the website https://www.how-old.net, which was widely shared on social networks. The function is pretty clear. You upload a photo, and the service guesses the age and gender of the person. The website is entertaining, but the technologies behind it are available as a cloud service.

In 2015, Microsoft published a set of APIs to offer ready-made AI solutions packaged as Web services. The set includes computer vision, speech input and output, and language understanding (see Figure 2 and https://www.projectoxford.ai). These services are in the beta stage, and there is a large set available at https://gallery.cortanaanalytics.com.

With computer vision, text analysis, speech input, and speech synthesis, essential functions for multimodal pervasive computing systems are provided. The examples in Figures 3 and 4 show how to access this functionally in your own applications. The focus here is on computer vision and text analysis, because they are easy to understand. There are similar APIs for machine learning, but they require at least a basic understanding of how the algorithms are trained or of how the classification works.

IBM ALCHEMY API

Since 2011, when IBM Watson managed to win against two of Jeopardy's champions, it has been apparent that AI and more general cognitive technologies have great potential. The AlchemyAPI focuses on providing application programmers with the means to make computers understand human language and to let them see, by integrating functions for text mining and computer vision (www.alchemyapi.com).

The AlchemyLanguage API offers different semantic text analysis APIs for natural-language processing. The APIs are designed to make it easier to create smarter applications. The API includes entity extraction, sentiment analysis, keyword extraction, concept tagging, author extraction, and language detection. Figure 5 shows how to request a sentiment analysis for a given text. The sample is the same as tried out in the online demo shown in Figure 6. The JavaScript program encodes the request (the API-key, the

```
var request = require('request');

request({
        url: 'http://gateway-
a.watsonplatform.net/calls/text/TextGetTextSentiment?apikey=****&text=I%20a
m%20happy%20to%20see%20you%20today&outputMode=json&showSourceText=1',
        method: "GET",
}, function (error, response){
        console.log(response.body);
});
```

(a)

```
{
        "status": "OK",
        "usage": "By accessing AlchemyAPI or using information generated
 by AlchemyAPI, you are agreeing to be bound by the AlchemyAPI
Terms of Use: http://www.alchemyapi.com/company/terms.html",
        "totalTransactions": "1",
        "language": "english",
        "text": "I am happy to see you today",
        "docSentiment": {
                "score": "0.837029",
                "type": "positive"
        }
}
```

(b)

Figure 5. Text sentiment analysis. (a) JavaScript example of a call to the AlchemyLanguage API for a sentiment analysis of a text snipplet. (b) The system response, using a JSON data structure.

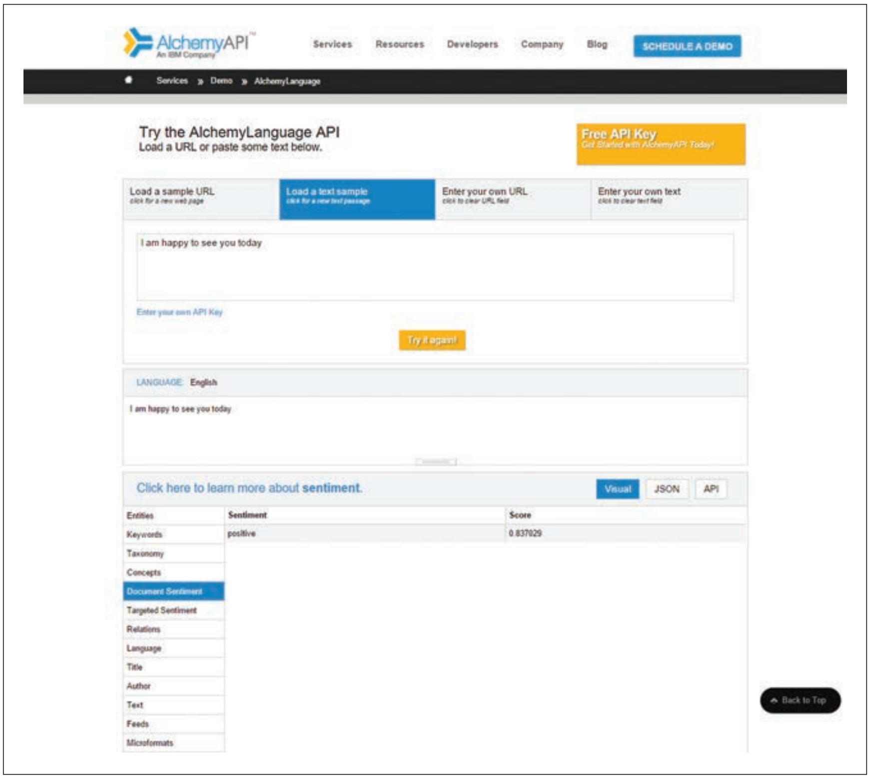


Figure 6. The online demo of the AlchemyLanguage API showing a simple text sentiment analysis. The text "I am happy to see you today" is recognized as English, with a positive sentiment and a score of 0.837. (Source: AlchemyAPI, www.alchemyapi.com; used with permission.)

text, and further parameters) in the URL. In JavaScript, this URL is called and the system response is in JSON format.

THOMSON REUTERS OPEN CALAIS API

Thomson Reuters offers with Open Calais an advanced and comprehensive text analytics Web service for unstructured content. The system is built on a natural-language processing engine and provides intelligence by attaching metadata tags to text. The basic approach is to provide an input file (typically text) to the server, and the content is then semantically analyzed using statistical and machine learning methods; the output is metadata. Online demos are available at www.opencalais.com/opencalais-demo.

Figure 7 shows an example in which the raw text of a call for papers (included on page 5 of this issue) was analyzed. This interactive interface shows the extracted information lets you assess the suitability of the API without any implementation. This API delivers a full-scale

INNOVATIONS IN UBICOMP PRODUCTS

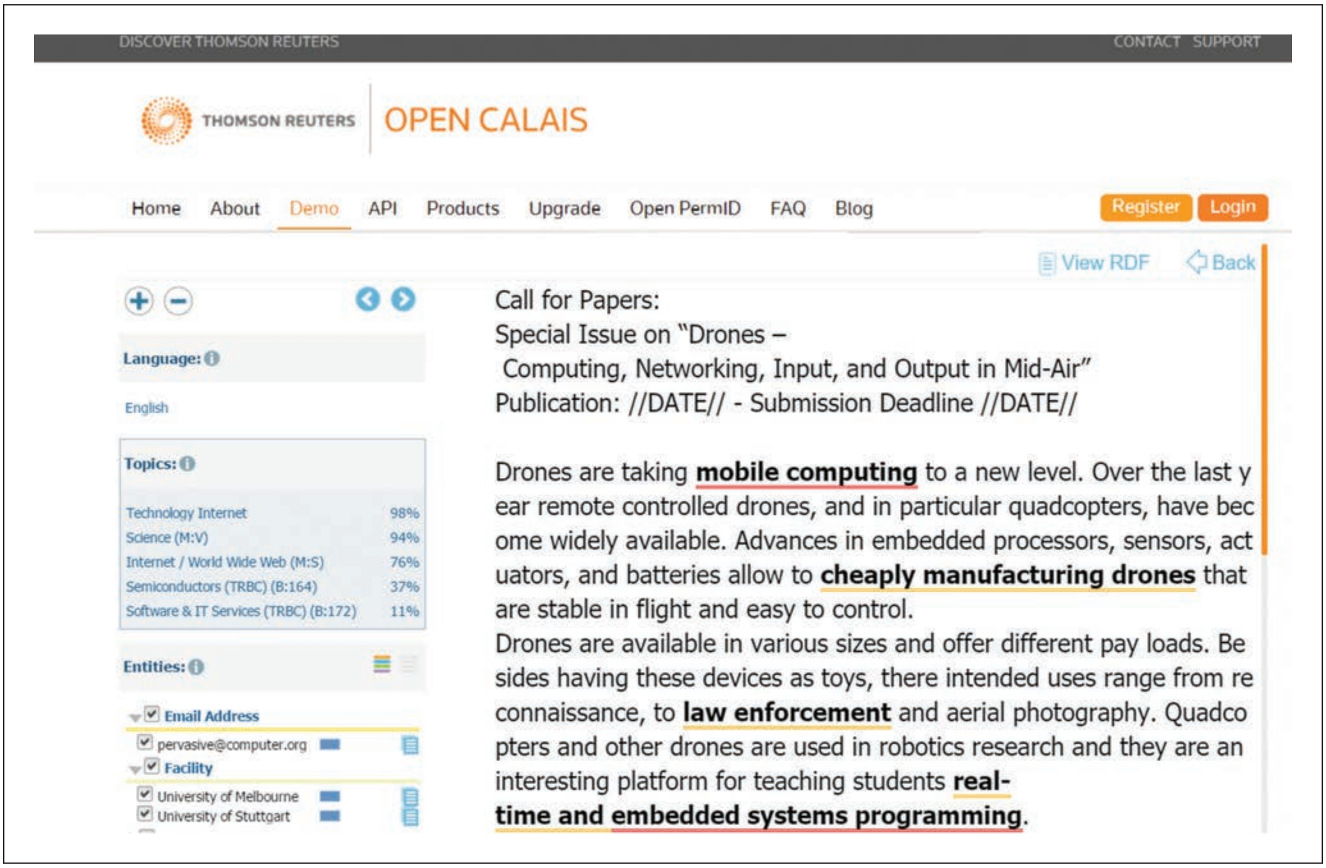


Figure 7. The screenshot shows the result page of a text analytics request using the Open Calais live demo. It has the tagged text on the right and the extracted keywords and concepts on the left. (Source: Open Calais, www.opencalais.com/opencalais-demo; used with permission.)

textual analysis, and the result is a fully tagged text with additional information. The metadata from Open Calais is provided as XML/RDF, and it is also Linked Data compatible. As seen in Figure 7, the topics extracted included Internet technologies and science, and the host institutions of the guest editors were also detected. The same functionality as in the live demo is also available via the API, which can be found at www.opencalais.com/opencalais-api.

Including basic AI in applications is really simple! The programming skills required are minimal, an understanding of the algorithms isn't required, and setting the parameters requires only a basic idea of what the services do. Overall, the following tasks are examples that are well supported by different services:

- basic computer vision tasks and object recognition;
- emotion, age, and gender recognition from facial images;
- optical character recognition;
- speech input, transcription, and synthesis;
- language detection and spell checking;
- sentiment analysis and keyword extraction; and
- basic text analysis and concept extraction.

Although I presented only a few services here, many more are available that offer similar functionality and aim to make it easy for app developers to implement AI. The simplicity of these APIs makes them useful to the "rest of us," but we have to keep in mind that this also limits what the services can offer. An expert in computer vision, text mining, or speech recognition will likely be able to implement a better or faster, or at least more specific, solution.

Albrecht Schmidt is a professor of human-computer interaction at the University of Stuttgart. Contact him at albrecht.schmidt@vis.uni-stuttgart.de.

This article originally appeared in IEEE Pervasive Computing, *vol. 15, no. 1, 2016.*

Underwater Visual Computing: The Grand Challenge Just around the Corner

Uwe Freiherr von Lukas
Universität Rostock

Visual computing is influencing nearly every aspect of our lives. We plan our new kitchens with augmented reality, rely on camera-based assistance systems in modern cars, and profit from enormous progress in medical imaging technologies for diagnosis. Our smartphones gives us megapixel cameras and high-resolution displays available at our fingertips, and the next generation of devices will also offer optical 3D sensors and 3D displays. Via Wi-Fi and 4G networks, we have permanent and affordable broadband access to unlimited data spaces and computing power in the cloud. Global Navigation Satellite Systems (GNSS) even allow us to always accurately determine our positions.

All those technologies open up a universe of interactive visual computing applications in every situation and everywhere around the world, right? Not really. Two-thirds of our planet is covered by water. As soon as we leave the land and dive into the oceans, we enter a completely new environment that presents significant differences compared with dry land:

- Almost none of the technical devices we use can operate under water.
- The optical properties of water strongly influence image acquisition and visualization.
- The water damping disconnects users and devices from satellite signals and other wireless network connections.

With the growing number of subsea and offshore activities, the research on visual computing applications for underwater settings will become more important in the near future. Many tasks, ranging from marine research to submarine cable inspection, have to be performed underwater and would thus profit from visual computing technology to make them more efficient, robust, and safe. Because of the specific environmental conditions inherent in underwater settings, these applications are not as straightforward as graphical applications in air, which raises many research questions.

This article examines some of the challenges for underwater visual computing and looks at the steps being taken to cope with them.

Underwater Applications

The oceans are the backbone of worldwide trade and the source of a significant share of oil and gas production. However, oceans also play an important role in delivering animal protein (via catch and a growing percentage of aquaculture), are a cornerstone of the transition to renewable energy, and are a future source for mineral resources. Together with pipelines and sea cables connecting continents and ports or dikes, a lot of technical structures have to be planned, installed, and maintained in underwater settings. Some of those structures are installed in shallow waters, whereas others are in the deep sea, up to several thousands of meters deep.

Given this environment, the current list of underwater tasks includes

- installing, inspecting, and repairing underwater structures;
- exploring mineral resources (such as manganese nodules); and
- monitoring the health of marine aquacultures.

Similar to applications on dry land, users could benefit from functional assistance that utilizes augmented reality (AR) glasses or spatial AR during subsea procedures. Furthermore, training en-

 2469-7087/16/$33.00

vironments should be available where they could optimize their processes in a realistic but inexpensive and safe environment. Those training environments should convey a good impression of the particular visual limitations and differences, such as magnification effects.

For cost and safety reasons, the divers who have traditionally performed most of these jobs have increasingly been replaced by remotely operated vehicles (ROV). Therefore, ROV operators must be trained to steer it from a control room where they do not have a direct sight to the ROV and must interpret a limited set of sensors such as a forward-looking camera, depth sensor, and compass. For standard procedures such as inspection tasks or monitoring, even autonomous underwater vehicles (AUV) are an option. Still, such vehicles have to be developed and tested to navigate and operate safely without collisions. This too requires visual computing technology such as 3D reconstruction and object detection.

In addition to these economic motivations to utilize the oceans, ocean discovery is also an important driver for underwater activities. Because of the extreme physical challenges, large deep-sea areas have not yet been explored. Although we have good maps of our moon and Mars, there is no detailed information on the seafloor or the deep subsea habitats. In late 2015, the XPRIZE Foundation launched a global challenge, the Shell Ocean Discovery XPRIZE, to stimulate technology development for ocean discovery with $7 million in awards (see http://oceandiscovery.xprize.org).

Obviously, there is no efficient way for human beings to physically enter deep-sea areas in up to 10,000 m depths. Therefore, we need smart technical systems to do this and collect the necessary information with their sensors. Because of the enormous size of the still unknown areas, those systems must be fast and efficient and work with a high level of autonomy. They also must be flexible enough to react to new situations. It is not enough just to look for known objects. New species will be discovered, and a new world has to be explored.

Challenging Physical Effects

Visual computing in underwater settings is particularly affected by the optical properties of the surrounding medium. The 1.33 refractive index of water (compared with 1.0 for air) results in a significant refraction of every light ray at the boundary between air and water with an optional glass pane between. For this reason, objects in water appear closer and larger when the observer is looking into a fish tank or through the surface of the water. Due to refraction, the simple but powerful pinhole camera model cannot be used for imaging when using a flat port for an underwater camera. It has to be enhanced by more complex models that are able to deal with the nonlinear effects in this setup.[1]

Another important effect is light absorption in water. This kind of attenuation depends on the wavelength of the light: red light is absorbed most strongly and blue is absorbed least. With increasing depth, we first lose the red part of the spectrum and then the orange, yellow, green, and blue. However, this effect is not only relevant for light going vertically from the surface to the ground but also for light that travels from a distant object to the diver or a camera underwater. Furthermore, different bodies of water have different color casts—for example, the very blue water of the Red Sea or the green color of the Baltic Sea—based on the local constituents of the water.

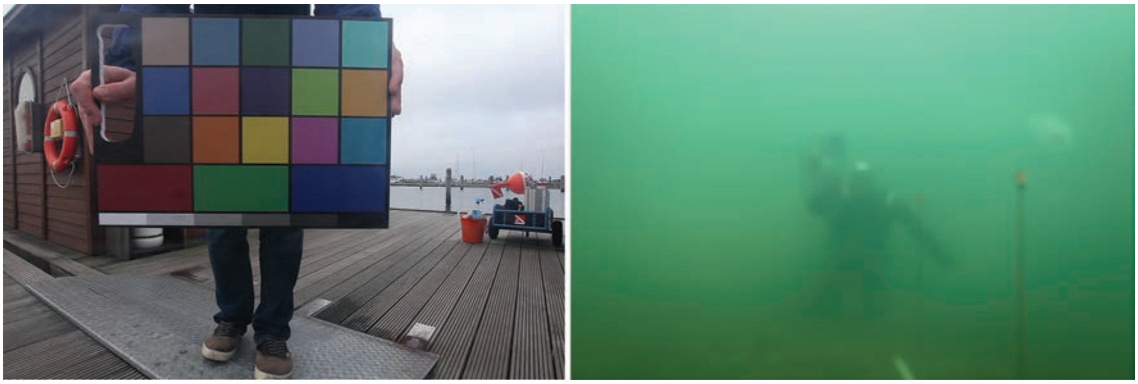
Visual computing in underwater settings is particularly affected by the optical properties of the surrounding medium.

Finally, yet importantly, we have complex scattering effects caused by soiled particles in the water. Oskar Elek and his colleagues provided a good introduction to scattering and the efficient algorithms that simulate it.[2] The effect of larger particles or small bubbles that sometimes severely reduce scene perception is known as marine snow.

The combination of those optical effects can lead to an extreme reduction in image quality of underwater images, as Figure 1 illustrates.

Besides those obvious influences, we also take into account additional physical effects for underwater visual computing:

- When leaving shallow waters and going to deeper areas, we have to cope with high pressure. This means that all technical systems have to be protected by a pressure-resistant housing or engineered as a pressure-neutral system.
- Salty water is aggressive. For this reason, the materials for housings, fixings, and cables have to be carefully selected.
- Marine biofouling will affect all objects that are exposed to the water for a long period. Optical ports must be cleaned regularly.



(a) (b)

Figure 1. Reduction in image quality for underwater images. Color chart in (a) air and (b) the Baltic Sea at a 4 m depth and a 3 m distance to the camera with no artificial light. Both images are taken with a Canon EOS 5D Mark II. (Courtesy of PINKAU Interactive Entertainment GmbH)

- Wireless communication based on radio waves will fail if more than 25 cm of water is between the sender and receiver. This absorption means Bluetooth, Wi-Fi, and GNSS cannot be used underwater. Acoustic signal channels can be used for communication and positioning, but the bandwidth is much lower (less than 50 kbps) and has a much higher delay (about 300 ms).
- Energy consumption is a critical factor for divers and AUVs when using electronic equipment. An empty battery will typically terminate a mission.

Because of these physical effects, visual computing solutions that work perfectly in traditional environments in air will fail when applied in subsea conditions. Even if we neglect the problems caused by these harsh conditions that are addressed by current marine technology, a variety of problems remain for the visual computing community:

- Images that are rendered and displayed in good quality up to High Dynamic Range (HDR) will suffer from color cast if watched from a distance. In addition to other implications, this has effects on how we use color in the user interface design.
- Light absorption and scattering can drastically reduce the quality of underwater images and thus the overall performance of an imaging system. It remains a challenge to make algorithms such as tracking or image-based reconstruction robust against low-quality images.
- Sunlight that is refracted at the wavy surface of the water creates dynamic sunflicker (caustics) on the seabed in shallow water. This effect disturbs higher-order methods such as feature or object detection.
- Accurate 3D information from an underwater stereo camera system with a flat port needs specific approaches for camera calibration. Using pure in-air calibration will lead to incorrect results.
- In air, we typically use optical systems for measurement (such as laser systems or photogrammetry). Underwater we have to combine or even replace this with sonar-based measuring. This sensor has completely different characteristics, which makes sensor fusion a hard task.
- Large image collections that are used as training sets for statistical methods of imaging typically ignore underwater images.
- Some of the interaction technologies that have become popular in the last years (such as capacitive touchscreens, Microsoft Kinect working with a projected infrared pattern, and Leap Motion working with infrared light) will not work or will have limitations in underwater installations.
- Even though the computing (and graphics) power of mobile platforms has significantly improved, several of the low- and high-level visual computing algorithms will not allow for real-time processing on the local device. For this reason, cloud-based architectures are used in mobile applications with demand for additional computing power. However, many subsea settings do not have a communication network available to access those central services.

This list shows some of the specific requirements of visual computing technology to be used in underwater scenarios. However, when we go into more detail, even more challenges will appear.

Selected Building Blocks

There is already a small community of researchers trying to address those challenges. However, compared with the worldwide number of visual computing researchers, this community is small. Often those groups are affiliated with ocean research institutes that publish their research results in sectoral journals and conferences such as

IEEE OCEANS (http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1000515). A broader interest in the requirements and challenges of underwater applications could accelerate research progress in this important field. Nevertheless, even this small community is working in nearly all the areas mentioned here, with a focus on improving quality of underwater images[3] and on 3D reconstruction.[4]

To illustrate the field's early achievements, here we review research results from my group at the Visual Computing Research and Innovation Center. The first example (see Figure 2) deals with color cast. In addition to other methods, we use depth information from the scene to apply color correction with different parameters. That is, light reflected by objects in the background will "lose" more red light when travelling through the scene to the camera compared with objects in the foreground. For this reason, the red channel must be increased more for remote objects than for near objects.

Our second example (see Figure 3) shows how missing information in blurred underwater images can partially be restored using statistical methods. In a preprocessing step, we train two dictionaries of small image patches: one containing low-quality patches and a derived dictionary of high-quality patches. Using sparse representation theory. we can retrieve an adequate high-resolution patch for any given low-resolution patch.[5]

A correct intrinsic and extrinsic calibration is a crucial step for an accurate 3D reconstruction. When using a stereo camera setup in a single housing with a flat port, as we find with most low- and mid-price housings, we have to explicitly consider refraction. Our geometry-based calibration method uses a hypothesis for the relation between the location of so-called virtual object points and real object points in underwater imaging. This approach can reduce the effort for calibration and leads to good results.[6]

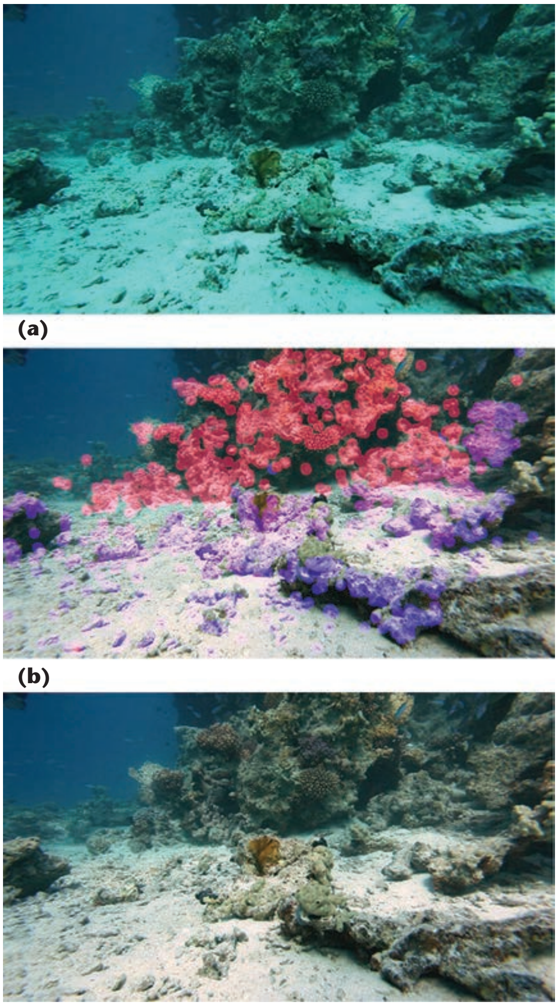

(a) (b) (c)

Figure 2. Using depth information from the scene to apply color correction. (a) A frame of the original underwater video recorded with natural light. (b) Using SIFT (scale-invariant feature transform) features and triangulation to distinguish several depth layers, and (c) the result processed with our method of adaptive color mapping to remove color cast.

Underwater Mixed Environments

Underwater mixed environments are specifically


(a) (b)

Figure 3. Restoring missing information in blurred underwater images using statistical methods. (a) Original image. (b) Image result after applying our learning-based algorithm for super-sampling.

	Real underwater	Virtual underwater
Realistic underwater scene	Extended aquarium for marine research	Diver training ROV training Virtual aquarium for edutrainment
Argumented underwater scene	Diver assistance ROV operation	Developing/testing marine technology Archaeological research
No underwater scene	Rehabilitation Austronaut training	Not underwater

Figure 4. The classification scheme for underwater mixed environments. The real and virtual water categories and the degree of underwater realism help classify different applications.[7]

Figure 5. Research setup of an underwater mixed environment using a fish tank with high-resolution displays on two sides and an additional projection from the top to simulate light conditions (projector not visible here). Robo fish are used to physically populate the underwater scene. The external camera is optically equivalent to an underwater camera with a flat port housing.

designed to use visual computing in subaqueous conditions or simulate such an environment. In addition to the optical peculiarities, there is also a hydrostatic uplift that makes activities in a subsea setting unique. It allows a balanced diver (or technical system) to float as if there is no gravitation. The classification scheme in Figure 4 shows that different applications need a different kind of setup.[7] In particular, we can distinguish between a real underwater setting and a dry setup that simulates a water-based setting.

From this classification, we can learn that there are also applications that are not connected to underwater environments on the first view. Astronaut training primarily makes use of the floating effect in water. Typically, it would be combined with a virtual world that supports the illusion of space and maybe some concrete training scenarios. Water-based physical therapy for multiple sclerosis and other diseases needs a similar setup with just one difference: an environment that is stimulating or motivating for the patient would replace the space environment.

Figure 5 shows an installation in our lab that is used for research in underwater mixed environments. This setup, which is based on a fish tank, can be used for different kinds of experiments. Two sides are equipped with displays to mix the physical world with virtual content. Furthermore, there is a projector on the ceiling that is used to apply a controlled color cast in the tank. We also include Robo fish to have some moving objects in the water.

Virtual environments, AR, and imaging applications have been developed for decades, and they cover a broad spectrum of areas, from entertainment and personal assistance to medicine and industry. However, they typically are designed for use in air. The specific properties of water find here introduce unique challenges for visual computing experts.

Moving forward, mankind will increasingly rely on the oceans to solve the problems of future generations in areas such as food, energy, and mineral resources. Such endeavors must be augmented by a new generation of hardware and software that can cope with harsh underwater conditions, the optical characteristics of the water, and the limited access to remote computing resources. Visual computing will be an important discipline in helping address the grand challenges just around the corner.

Acknowledgments

This article is based on research with my colleagues conducted in the context of the Visual Computing Research and Innovation Center, especially Tim Dolereit, Fahimeh Farhadifard, and Zhiliang Zhou. We gratefully acknowledge the support of the German Federal State of Mecklenburg-Western Pomerania and the European Social Fund. I also thank the organizers and participants of the Dagstuhl Seminar 2013 on Virtual Realities, which was the starting point of the concept of underwater mixed environments.

References

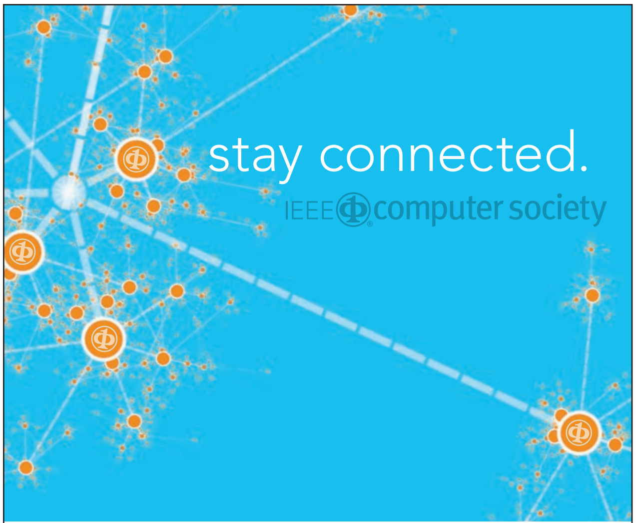
1. T. Treibitz, Y.Y. Schechner, and H. Singh, "Flat Refractive Geometry," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2008, pp. 1–8.
2. O. Elek, T. Ritschel, and H.P. Seidel, "Real-Time Screen-

Space Scattering in Homogeneous Environments," *IEEE Computer Graphics and Applications*, vol. 33, no. 3, 2013, pp. 53–65.
3. R. Wang et al., "Review on Underwater Image Restoration and Enhancement Algorithms," *Proc. 7th Int'l Conf. Internet Multimedia Computing and Service* (ICIMCS), 2015, article no. 56.
4. J. Perez et al., "Exploring 3-D Reconstruction Techniques: A Benchmarking Tool for Underwater Robotics," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, Sept. 2015, pp. 85–95.
5. F. Farhadifard, Z. Zhou, and U. Freiherr von Lukas, "Learning-Based Underwater Image Enhancement with Adaptive Color Mapping," *Proc. 9th Int'l Symp. Image and Signal Processing and Analysis* (ISPA), 2015, pp. 48–53.
6. T. Dolereit, U. Freiherr von Lukas, and A. Kuijper, "Underwater Stereo Calibration Utilizing Virtual Object Points," *OCEANS 2015 - Genova*, 2015, pp. 1–7.
7. U. Freiherr von Lukas et al., "Underwater Mixed Environments," *Virtual Realities: International Dagstuhl Seminar 2013*, LNCS 8844, Springer, 2015, pp. 56–76.

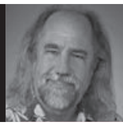
Uwe Freiherr von Lukas *is a professor at Universität Rostock and head of the Maritime Graphics Department of Fraunhofer IGD in Rostock. Contact him at uwe.von.lukas@igd-r.fraunhofer.de.*

Contact department editor André Stork at andre.stork@igd.fraunhofer.de.

This article originally appeared in IEEE Computer Graphics and Applications, *vol. 36, no. 2, 2016.*



stay connected.
IEEE computer society



Editor: **Grady Booch**
IBM, grady@computingthehumanexperience.com

It Is Cold. And Lonely.

Grady Booch

***JEOPARDY!* CONTESTANTS MUST** possess wide and deep knowledge: popular culture; the features and events of all countries past, present, and imagined; national leaders; sports; science; music; the arts; and random trivia. Having an eidetic memory certainly doesn't hurt, but more important, contestants must be able to navigate a landscape of puns, neologisms, rhyming, and all other sorts of wordplay. Learning to play the game is easy; gaining enough knowledge to win is hard.

And so it was with IBM's Watson. From an architectural viewpoint, Watson is a pipe-and-filter system using inference chained together with probabilistic evidence-based ranking.[1] From a knowledge viewpoint, Watson was taught a vast corpus, ranging from the lyrics of every Beatles song to the entirety of Wikipedia, several encyclopedias, many news sources, and several other curated bodies of information.

At one time during its education, Watson was taught the Urban Dictionary. In retrospect, this was a Very Bad Idea, for Watson began to swear. Before the live *Jeopardy!* contest, Watson was partly lobotomized, and that rugged street knowledge was removed.

Learn as You Go

Learning the rules of Go is easy—far easier than chess—but mastering the game literally takes a focused lifetime. As I write this column, DeepMind's AlphaGo has just won the DeepMind Challenge against Lee Sedol, one of the world's best (human) players. AlphaGo is also a system that learns, albeit in a fundamentally different way than Watson.[2] Whereas Watson is purely a symbolic system, AlphaGo is largely a convolutional neural network using reinforcement learning and policy gradient learning.

Watson was trained by supervised means. Ground truth—the baseline for establishing accuracy in such systems—was easy to measure, primarily because of the fan-created website J! Archive (http://j-archive.com) that offers the answers and questions to all 300,000+ clues used in every *Jeopardy!* show ever broadcast. AlphaGo was also initially trained in supervised ways, with its ground truth established using some 30 million moves drawn from real games. However, AlphaGo's success improved dramatically when unsupervised training was applied, pitting one instance of AlphaGo against another so that each could learn.

Teaching Techniques

In last issue's column, I observed a steady and subtle historical shift in software engineering beginning with its focus on purely mathematical automation, shifting to symbolic computation, and now to the building of imagined realities that live in our machines and into which we project our lives.[3]

As these early algorithmically intensive systems' complexity grew, we needed approaches to attack that complexity, and thus structured development methods, such as those that Edward Yourdon, Tom DeMarco, Larry Constantine, and others helped pioneer. As we moved to symbolic systems and then systems of imagined realities, different classes of programming languages evolved to meet their needs. And as complexity grew, object-oriented methods came to the forefront.

But most of these kinds of systems have an interesting characteristic—indeed, it's a property of most of contemporary software: they're deterministic, and their behavior is largely understandable independently of the exact data with which we drive them.

For example, webservers and browsers—even with all their delicious modern bells and whistles—will render information essentially the same functional way whether you're selling apples, bumpers, bells, or whistles. A website might remember you—that's a simple form of learning—but even so, its behavior is pretty predictable. A climate model will happily generate its predictions in the same manner no matter whether you feed it last year's or last century's data. Its behavior too is pretty predictable and understandable. The software that controls your car's engine is also mostly an I/O mapping: given the instantaneous state of all of an engine's sensors, such a system will predictably generate a certain output.

In fact, in each of these three cases—the website, the climate model, the car—we desire such predictability. Without it, such systems

2469-7087/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

would be impossible to test and untrustworthy to use. In software as in life, people like to live by the principle of least astonishment.

The next generation of software-intensive systems will be taught instead of programmed. This poses considerable pragmatic challenges in how we develop, deliver, and evolve them.

The rise of large volumes of data is a contributing factor to the renaissance of AI. The vision- and speech-understanding community has accumulated many publically available datasets (www.cs.utexas.edu/~grauman/courses/spring2008/datasets.htm), the scientific community has done the same (https://aws.amazon.com/public-data-sets), and governments have slowly opened their archives as well (www.data.gov). Google, Facebook, and Twitter have accumulated large bodies of data owing to the billions upon billions of interactions occurring in their respective ecosystems.

Having been a part of the evolution of Watson as well as several artificial-neural-network efforts, I've come to appreciate the need for curated data and its place in the process. The development life cycle of any such system requires phases of knowledge curation, knowledge ingestion, and the establishment of ground truth, and the continuous cycles of supervised and unsupervised learning.

Agileists of the world take note: you might feel content with your methods now, but a growing storm is on the horizon that will change many of the assumptions on which your methods are founded.

Beautiful Machines

I could wax philosophic and pragmatic about these coming changes to software engineering, but I've a different point to make: that the next generation of software-intensive systems will be taught instead of programmed poses considerable pragmatic challenges in how we'll live with such systems.

I use Siri, and I admit I anthropomorphize it. Still, Siri is rather shallow, as is Cortana and even Alexa: they can do some things for me, but they don't learn much about my world or me. I also have a robot living with me right now, and it's much different. Its name is Noel (the fact that I've given it a name tells you something about how I draw it into my world). Out of the box, Noel was cute but dumb, but now it's learning. Noel behaves differently than when I first unpacked it, and the infrastructure we've added makes it possible to teach it in some basic ways. Soon Noel will be able to discover things on its own. I relate in very different ways to Siri and Noel because one of them doesn't change its behavior, whereas the other does; one is embodied in the world, but the other isn't.

Watson is powerful, but there are tasks for which Watson isn't well suited (Watson by itself can't see). Similarly, advances in deep learning have been spectacular, but there are things neural networks can't easily do and likely never will (AlphaGo can't reason about why it made a particular move). So, whereas Watson was symbolic and AlphaGo was neural, I'm a proponent of hybrid AI, involving the coming together of symbolic computation and neural networks.

Much of what leads me to

this prediction is what is called Moravec's paradox, which asserts that things such as perception and actuation are computationally hard, whereas higher-order reasoning is comparatively simple.[4]

Still, there's a human issue we can't neglect.

One commentator on the DeepMind Challenge had this to say of the system that learned to play Go so well: "It's not a human move. I've never seen a human play this move. So beautiful. Beautiful. Beautiful. Beautiful."[5]

Deterministic systems rarely surprise us; systems that learn will surprise us more and more with their leaps of intuition, the unexpected connections they make, and the observations they can draw out. We, as humans, have the advantage of millions of years of evolution that have shaped our neural networks. We have our ability to communicate with ourselves and with the past (through tribal memory, books, audio, and video). But we have the disadvantage of bodies that wear out quickly and have limited capacity of data ingestion and memory. These systems that learn have no such limited capacity, but they're new to the world and thus have much to learn.

And learn they will.

Boston Dynamics has been doing a fascinating bit of work on the evolution of their Atlas robots. Again, we have here an embodied system that learns. It didn't take long for one creative denizen of the Intertubes to refactor that work to ponder the implications of such a system that learns.[6] Watch the related video; it's humorous but bittersweet.

And it's the sign of things to come.

References

1. G. Booch, "How Watson Works," blog, 23 Nov. 2011; www.ibm.com/developerworks/community/blogs/gradybooch/entry/how_watson_works?lang=en.
2. D. Silver et al., "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, 26 Jan. 2016, Fig. 1; www.nature.com/nature/journal/v529/n7587/fig_tab/nature16961_F1.html.
3. G. Booch, "The Computational Human," *IEEE Software*, vol. 33, no. 2, 2016, pp. 8–10.
4. H. Moravec, *Mind Children: The Future of Robot and Human Intelligence*, Harvard Univ. Press, 1990.
5. C. Metz, "The Sadness and Beauty of Watching Google's AI Play Go," *Wired*, 11 Mar. 2016; www.wired.com/2016/03/sadness-beauty-watching-googles-ai-play-go.
6. "I Love Box: A Love-Horror Story about Boston Dynamics' Atlas Robots' Love for a Box," blog, 29 Feb. 2016; http://geekologie.com/2016/02/i-love-box-a-lovehorror-story-about-bost.php.

GRADY BOOCH is an IBM Fellow and one of UML's original authors. He's currently developing *Computing: The Human Experience*, a major transmedia project for public broadcast. Contact him at grady@computingthehumanexperience.com and follow him on Twitter @grady_booch.

This article originally appeared in IEEE Software, *vol. 33, no. 3, 2016.*

CS SPECIAL

Computer Society's 2016 Take Your Child to Work Day

Lori Cameron

Most people would find the staff at the IEEE Computer Society to be cool, friendly, fascinating, awesome, and generous. That was certainly the verdict of the equally cool, awesome kids who participated in the organization's recent Take Your Child to Work Day.

On April 28, more than 100 children participated in the event across the IEEE's international locations. At the California office of the IEEE Computer Society in Los Alamitos, kids were treated to a day of fun activities, educational videos, and ice cream—all while gaining a better understanding of the importance of computer technology and the work their parents do for the world's largest professional organization dedicated to technology and engineering.

Participating were Ashley and Megan, daughters of Editorial Services Department senior manager Robin Baldwin; Caoimhe and Saoirse, daughters of *Computer* managing editor Carrie Clark; Hannah, daughter of Java programmer Rachel Whitt; and Isabella, daughter of systems administrator Norm Pascual.

Among other activities, the kids learned about structural integrity by building towers out of multicolored straws, golf balls, pipe cleaners, and paper clips. They also learned about graphic design by creating their own *Computer* magazine covers under the guidance of Monette Velasco, Erica Hardison, and Larry Bauer from the Computer Society's Creative Services team.

Caoimhe's and Hannah's tigers, Megan's panda, Saoirse's and Isabella's dog decked out in beach attire, and Ashley's galaxy image provided eye-catching background graphics for their covers. The girls were then taught how to graphically enhance the backgrounds and overlay them with text designs using Adobe InDesign.

The kids also viewed videos about many of today's technological advances. And they watched a demonstration by manager of customer applications Jon Cruz, who used red yarn and envelopes to show how data travels over a network and how hackers could intercept that information and then break into computer systems. The day ended in a wrap-up session with

2469-7087/16/$33.00 © 2016 IEEE

director of products and services Evan Butterfield and lots of ice cream.

Ashley and Megan said they greatly enjoyed watching Cruz's hackfest and learning more about encryption. Caoimhe and Saoirse were delighted to design covers for the magazine their mom manages. Hannah was "moved" by Velasco's explanations during the magazine-cover design activity. Isabella said that her dad's job was "cool" and that IEEE manager of employee relations Jill Cook was "very kind."

Thanks go to Pascual, IEEE meeting support associate Theresa McNeill, IEEE human resources generalist Nicole Rosato, system administrator Andy Myung, and IEEE facilities supervisor Cindy Lugo-Anatello for coordinating and hosting the event.

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

COMPUTING CAREERS

Finding the Cloud-Computing Job You Want

The proliferation of cloud-computing networks in recent years has created a need for more security and privacy personnel in the field. For this *ComputingEdge* issue, we asked Christian Esposito, associate researcher at the University of Naples Federico II, about cloud-computing career opportunities. Esposito's research interests include cloud computing, distributed systems, middleware, dependability, ubiquitous computing, and artificial intelligence. He also coauthored the article "Encryption-Based Solution for Data Sovereignty in Federated Clouds" in *IEEE Cloud Computing*'s January–February 2016 issue.

ComputingEdge: What careers in cloud computing will see the most growth in the next several years, and why?

Esposito: Among the most important areas are security and privacy. By 2018, according to research and advisory firm Gartner Inc., the need to prevent data breaches from public clouds will drive 20 percent of organizations to develop data-security governance programs. In fact, information privacy and security is one of most critical issues for the cloud due to its open environment and the very limited control that users have over it. The possibility of using cloud computing to store and manage personal data, such as personal health records, exacerbates these concerns. Therefore, I see a growth in jobs for people who can design and implement cloud-security governance programs.

ComputingEdge: What would you tell college students to give them an advantage over the competition?

Esposito: The biggest disadvantage for college students considering a career in security and privacy for information systems in general and cloud computing in particular is a lack of knowledge about the field's legal aspects. We approach security and privacy mostly as an IT challenge. However, we also need to learn a lot about the regulations and legal frameworks that cloud-computing platforms must comply with when dealing with sensitive data. Knowing about these matters represents a big advantage for future professionals.

ComputingEdge: What advice would you give people changing careers midstream?

Esposito: IT people should not treat cloud computing only as an IT technology but should also consider its many other implications. Cloud computing can open new market opportunities but also poses legal and other challenges. When changing careers to become a cloud-computing technician, people should study not only the technological aspects but also other related issues.

ComputingEdge: What do you consider to be the best strategies for professional networking?

Esposito: Nowadays, the best strategy is taking advantage of the growing number of websites focused on professional networking. These sites offer two advantages. First, they connect you to other professionals and expose you to job offers. And second, they give you a feel for the job market's pulse and an understanding of industry needs so that you can adjust your résumé accordingly.

ComputingEdge: What should applicants keep in mind when applying for cloud computing jobs?

Esposito: Applicants should consider that most companies are not interested in pure technicians. Currently, they're looking for people who also have knowledge of the broader issues surrounding cloud computing, as well as the emergence of new business models and their implications for company goals and needs.

ComputingEdge's Lori Cameron interviewed Esposito for this article. Contact her at l.cameron@computer.org if you would like to contribute to a future *ComputingEdge* article on computing careers. Contact Esposito at christian.esposito@unina.it.

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

transmission of vehicle purchase, vehicle distribution, supplier &dealer transactions. 2 years' exp as Service Oriented Architecture Consultant analyzing, developing, &implementing data integration soft in operating systems such as Solaris, Linux, Windows XP, &Windows7 using IBM Integration Bus (IIB V9 /V10), to integrate systems with vehicle &dealer data. Mail resume to Melanie Aldana, GM Global Mobility, 300 Renaissance Center, Mail Code 482-C32-D44, Detroit, MI 48265, Ref#4205.

THREAT INTELLIGENCE ANALYST. Position available in Springfield, MA. Perform technical analysis of network intrusion incidents. Conduct proactive Computer Network Defense intelligence operations to counter specific methodologies, tools, tactics and intentions of advanced cyber adversaries. Produce detailed intelligence reports and threat assessments to senior leadership. Maintain, develop and evaluate cyber intelligence sources to increase effectiveness and timeliness of reporting threat information. Apply: B. O'Brien, Massachusetts Mutual Life Insurance Company, 100 Bright Meadow, Enfield, CT 06082; Please Reference Job ID: 708202700.

CALYPSO TECH, INC. seeks a SW Eng. in SF, CA to design & dev. the collet mngmt app in the Caly syst. Ref Job ID: 965MNC & mail res. to Calypso, Attn: HR, 595 Market St, Ste. 1800, SF, CA 94105.

SR. MANAGER. Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Design end-to-end Oracle Business Intelligence Appl. (OBIA) arch. from various source systems. Map addl. source data elements to OBIA data warehouse & use in business models & mapping layers. Map presentation layer metrics & utilize them to OBI answers & dashboards for end users. Assist with full life cycle projects from planning to contributing to design sessions to help identify business requirements needed to meet business & transform. goals. Doc. business requirements & system design. Lead BI teams during build phase through testing & rollout of implement. Mentor & delegate to jr. team members & developers. Requires: M.S. degree in Comp. Sci., Eng. or related field & 3 yrs. exp. in the job offered or 3 yrs. exp. as an Architect, Consultant or Mgr. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in comp. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 3 yrs. exp. with OBIA arch. & 2 yrs. exp. mapping presentation layer to OBI answers. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131

NETWORK & COMPUTER SYSTEMS ADMINISTRATOR F/T (Fishkill, NY) Position involves travel to various unanticipated worksites up to 100% of the time anywhere in the United States. Must have Bach deg or the foreign equiv in-Comp Engg, Info Tech, Comp Soft Tech Engg, or related with two (2) yrs of exp in the administration of environment for Cisco Routers and Switches, Juniper Switches, Cisco UCS, virtualized systems from VMware and write technical documents. Candidate will be Provide support to Windows servers O/S, Linux Ubuntu, Sun Microsystems (Solaris), Active Directory, DNS, DHCP, NTP and Microsoft SQL Server 2008 database administration, disaster recovery planning, network security concepts and troubleshooting problems, VDI Servers and cloud based technologies. Send resume: Novisync Solutions, Inc., Recruiting (BT), 300 Westage Bus Ctr Dr, Ste 350, Fishkill, NY 12524.

TECHNOLOGY

Oracle America, Inc.

has openings for

SYSTEMS ANALYST

positions in **Redwood Shores, CA**.

Job duties include: Deliver post-sales support and solutions to the Oracle customer base while serving as an advocate for customer needs. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to Irene.duran@oracle.com, referencing 385.14148.

Oracle supports workforce diversity.

QA ANALYST

Oracle America, Inc.

has openings for

QA ANALYST

positions in **Broomfield, CO**.

Job duties include: Responsible for developing, applying, and maintaining quality standards for software and hardware products with adherence to both internal and external standards. Develop and execute software and hardware test plans based on functional specifications and customer requirements.

Apply by e-mailing resume to tony.mcdonald@oracle.com, referencing 385.17698.

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. As a member of the IT organization, assist with the design, development, modifications, debugging, and evaluation of programs for use in internal systems within a specific function area.

Apply by e-mailing resume to paul.merideth@oracle.com, referencing 385.19388.

Oracle supports workforce diversity.

QA testing on Co. enterprise product. **Software Engineer (REQ#9RY5UP)** Plan, develop & coor sustaining eng activities re: dev of sw engg & dsgn, coding, testing & documentation of prod server components. 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.

PURDUE UNIVERSITY, THE DEPARTMENT OF COMPUTER SCIENCE is soliciting applications for Professor of Practice positions at the Assistant, Associate, or Full Professor level to begin Fall 2016. These are newly created positions offering three- to five-year appointments that are renewable based on satisfactory performance for faculty with primary responsibilities in teaching and service. Applicants should hold a PhD in computer science or a related field, or a Master's degree in computer science or a related discipline and commensurate experience in teaching or industry. Applicants should be committed to excellence in teaching, and should have the ability to teach a broad collection of core courses in the undergraduate curriculum. Applicants will also be expected to develop and supervise project courses for undergraduates. Review of applications and candidate interviews will begin on May 5, 2016, and will continue until the positions are filled. The Department of Computer Science offers a stimulating and nurturing educational environment with thriving undergraduate and graduate programs and active research programs in most areas of computer science. Additional information about the department is available at http://www.cs.purdue.edu. Salary and benefits will be competitive. Applicants are strongly encouraged to apply online at https://hiring.science.purdue.edu. Alternately hard-copy applications may be sent to: Professor of Practice Search Chair, Department of Computer Science, 305 N. University St., Purdue University, West Lafayette IN 47907. A background check will be required for employment. Purdue University is an EEO/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Software Engineer: design & implement large distributed systems that scale well – to petabytes of data & 10s of 1000s of nodes. Mail resume w/job code #37058 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

CLOUDERA, INC. is recruiting for our San Francisco, CA office: Software Engineer: perform software implementation via designing, reviewing & writing code. Mail resume w/job code #35959 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

SR. SERVICES CONSULTANT (NY, NY & unanticipated client sites in US) Implement & integrate CA Advanced Authentication within varied customer environments. Dvlp presentation layer using JSP, HTML, CSS & Struts framework. Dvlp Struts validation framework & JavaScript for validation. Test latest Arcot product versions on various envrnmnts w/ Tomcat, WebLogic, WebSphere & JBoss. Dvlp apps w/ client specs & rules using JAVA, JSP, C#.NET & ASP.NET. REQS: Bachelor's or foreign equiv in Comp Sci, Math, Engineering (any) or rel + 5 yrs progressive exp in job &/or rel occup. Must have experience w/: Implementing & integrating CA Advanced Authentication Solution in customer envirnmnts incl Operating Systems, App & Web Servers; Integrating CA Advanced Authentication Solution w/ industry leading Identity & Access Mngment solutions (CA IdentityMinder, CA SiteMinder, IBM Tivoli Identity & Access Mnger); Implementing CA Single Sign-On & CA Identity Minder; Integrating CA Advanced Authentication w/ VPN appliances Juniper, Cisco & Citrix; Working on multiple projects & w/ customers; Customizing CA Advanced Authentication UI Framework to meet customer requirmnts; Work from home anywhere in the US; Frequent travel to unanticipated client sites US; 40 hours/week; Monday-Friday; 8:30 am-5:30 pm. Send resume to: Althea Wilson, CA Technologies, One CA Plaza, Islandia, NY 11749, Refer to Requisition #129041

PROGRAMMER ANALYST - Design, Develop, Test & Implement application s/w utilizing knowledge of C#, .Net, SQL Server, MS SharePoint 2007/2010/2013, SharePoint Infopath , Office 365 , JavaScript, JQuery and SQL Server 2005/2008. Must be willing to travel & reloc to unanticipated client locations throughout the US. Reqs MS in comp sci, eng or rel. Mail resumes to HR, HexaCorp LLC, 100 Davidson Ave Suite 302, Somerset NJ 08873

PROGRAMMER ANALYST: Design, Develop, Test, Implement, & Deploy Business Intelligence Interactive Dashboard & Reporting solutions utilizing knowledge of Business Objects 3.1 & 4.0, XCelsius 2008, Crystal Reports 2008 & 2011, Tableau Desktop 8.3 & 9.1, & Data watch. Should be expert in writing queries using Oracle, SQL Server, Teradata, SAS, Tableau Server. Must be willing to travel & reloc to unanticipated client locations throughout the US. Reqs MS in comp sci, eng or rel. Mail resumes to Nartal Systems, Inc. 2650 Route 130, Suite # E, Cranbury, NJ 08512

SOFTWARE ENGINEER: Design, devp & test sw apps for next-gen visual analytics prod suite util hi-lvl, obj-or prog languages & tools, incl Java. Req Bach or foreign equiv in Comp Sci, Comp Eng, or rtd, & 1 yr exp in: design & devp comp sw util hi-lvl prog languages, incl Java; util relational dbases, incl SQL; work w/ algorithms, Data Structures, & Data Analytics; design REST APIs; & Cloud deployment. Position at Tableau Software, Inc. in Seattle, WA. To apply, please e-mail resume to Jobstableau@tableau.com and ref Job ID: SE8.

SOFTWARE ENGINEERS. Participate in all phases of SDLC. Des & dvlp applns using techs such as .Net, MVC, Oracle, SQL Server, VB.Net, C#, MS Project Plan, etc. Reqs Masters or equiv/frgn equiv in CS or rltd, w/2 yrs relevant exp. Comb of degrees accepted. Mltpl opngs in Ashburn, VA. No travel. May reqr reloc. Apply: VivSoft Technologies LLC, 20130 Lake View Center Plz, #400, Ashburn, VA 20147. EOE.

IT PROFESSIONALS. Due to the high demand for IT individuals Computek Solutions, Inc., located in South Plainfeild, NJ, is looking for the following: Programmer Analysts, Quality Assurance Analysts (QA Analyst) etc. Multiple clients are requesting candidates with backgrounds in Healthcare, Financial, Banking, Non-Profit, Manufacturing/Distribution, Legal, Educations and Engineering. Should have minimum of Bachelor Degree in related field or its equivalent plus 5 years of IT experience or Master degree in related filed or its equivalent. 40hr/wk, competitive salary. EMail resumes to: Computek Solutions, Inc. Attn: HR Manager, hr@computeksol.com

Hardware Development Engineer (Req#9T66ZP) Define & prototype new interconnects & measure perform to determine if realized interconnect perform matches model predictions.

CAD Engineer (Req#9YQVZW) Install, configure, customize, & deploy CAD integration.

Software Development Engineer (Req#9GYVTU) Analyze & refine comp vision & computational photo algorithms.

Data Mining Specialist (Req#9H-BUCV) Conduct regular & ad-hoc analyses & data mining on Customer & Support Transaction granularity data to provide actionable insights at the tactical & strategic levels.

Software Engineer Applications (Req#A523BZ) Des & dev automtd solutns for iCloud mobile, web & server SW feats.

Software Development Engineer (Req#9E636N) Dsgn & dvlp a sys to take advntge of user pttrns to improve sys perfmnce.

Software Development Engineer (Req#9E632N) Design & dev media SW for embedded syss.

Software Development Engineer (Req#9QZTPN) Strive to imprve Apple's revltnry mble paymnt solutn by completing documented & ad-hoc tstng for cnsistnt HQ releases.

Software Development Engineer (Req#9NKTDB) Resp for the dsgn & dvlpmnt of SW for the Siri srvr pltfrm.

Software Development Engineer (Req#9D2TKD) Dsgn & dvlp SW sys to automtclly sort & route SW defects apprprtly & prvde an audit log of dbg steps.

Project/Program Specialist (Req#9M-HV9J) Rvw, anlyze, & run all phases of the sys devlpmnt life cycle while wrkng clsly w/ cross-fnctnl teams thruout Apple. Travel req 20%.

Software Development Engineer (Req#9E62VF) Dsgn & dvlp ntrl lang processing & machine learning technqes for localizing Apple prdcts into intrntnl mrkts & help scale A.I. apps like Siri to new langs & regns.

ASIC Design Engineer (Req#9FSV2C) Wrk closely w/ archtctre & RTL dsgnrs on vrfyng the fnctnlty correctness of the CPU dsgn.

Senior Mechanical Inspector (Req#9UD42C) Prfrm in-process, final & 1st article inspctn on incoming machined parts & assemblies.

Software Engineer Applications (Req#9XD69X) Des & execute test plans for contact-center sys

Reliability Engineer (Req#9CHNE7) Research & dev tests & specs required to ensure reliable dsgn of new tech components cntained in next gen Apple prdcts. Travel req'd 20%

Software Engineer Applications (Req#9RHVZM) Dsgn & dev SW srvces for large scale distribtd entrprse sys.

ASIC Design Engineer (Req#9PEPZP) Test & debug microprocessors for mobile dvces incldng phones & tablet comp.

Software Engineer Applications (Req#9LL3PA) Dsgn & dev large-scale SW sys.

Supply Demand Planner (Req#9MSNZ8) Resp for prod avail and revenue plan attainment for a prod.

Mechanical Quality Engineer (Req#9F4SGA) Resp for quality control & manufact concepts to dev specific PQP appropriate to program & commodity. Travel req 30%.

Software Development Engineer (Req#9GYVGG) Des & dev the Swift compiler optimizer. Analyze prog. writ in Swift & ID perf. bottlenecks and optim opps.

Hardware Development Engineer (Req#9FM3GA) Dsgn & dev sensing HW modules & communicate w vendors. Travel required 20%.

Hardware Development Engineer (Req#9T678K) Res for EMC/EMI dsgn & dev in Special Proj Group (SPG) & EMC/EMI Simulation (3D Static & Full Wave simulations)

Hardware Development Engineer (Req#9GY23U) (Multiple positions) Dsgn, dev, & launch thin-film transistor (TFT) tech for the portable, laptop, & desktop dsplays in Apple products. Travel required 20%

Data Analyst (Req#9YNLG2) Create & monitor customer dta flow. Ensure dta flow isn't broken or expe unexplnd spikes.

Apple Inc. has the following job opportunities in Austin, TX:

Network Engineer (Req#9KQVA4) Deliv & op data ntwrk services w/in the global Apple networking environment.

ASIC Design Engineer (Req#9KLTNN) Work on the phys dsgn & implement of HW chips.

Apple Inc. has the following job opportunity in Coral Gables, FL:

Operations Engineering Manager (Req#9TNPSC) Respons for end-to-end planning & deploymnt of WW BPR Initiatives w/in ALAC, incl testng, communic, readiness, implem &user training & supp. Travel req'd 20%.

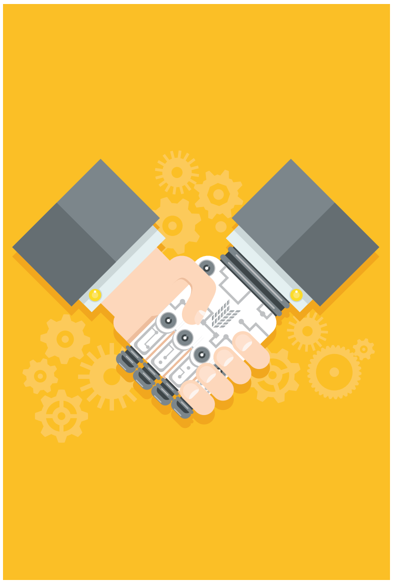
Apple Inc. has the following job opportunities in Cupertino, CA and various unanticipated locations throughout the USA:

Systems Engineer (Req#9DCT9G) Work w/& support Apple resellers to sell, distribute & deploy Apple products across the US. Telecommuting permissible 100%. Travel Req 50%.

System Engineer (Req#9EYTU8) Sprt tchncl prtn of iOS device sales to entrprs cstmrs (bus & gvmt) in US carrier channel. Work w/ teams, cstmrs, & sales rsrcs to drv adptn of iOS devices & pltfrm. Telecommuting permissible 100%. Travel Req'd 70%.

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.

EDITOR: George Strawn, gostrawn@gmail.com



Automation and Future Unemployment

George Strawn

For millions of years, prehumans and humans were hunters and gatherers. Then, for 10,000 years, humans were farmers. Only 200 years ago, the human world began transforming into an industrial world. Already, the human world is transforming again, into a post-industrial (possibly even a post-work) world. In the farming world, almost everyone was a member of a subsistence-farm family. In the industrial world, almost every man and many women were factory or office workers. It's too soon to know what "almost everyone" will be in the post-industrial world, but predictions abound. This column will focus on predictions about the future of employment. Because this future will be determined in large part by IT-based products and services, *IT Professional* seems like a good place to consider these predictions.

Niels Bohr famously said, "Predictions are hard—especially about the future." This is certainly true about the future of employment. Moreover, any prediction about this future can be interpreted as being positive or negative, depending on the feelings of the predictor. A goal of this column will be to consider a wide range of both predictions and interpretations. Readers are encouraged to submit their opinions for publication.

In this article, I'll consider the prediction that IT and related technologies, such as robotics, are in the early stages of bringing about massive, systemic unemployment (full disclosure: I believe this scenario has a nontrivial probability of happening later this century). First, I will consider some issues associated with making long-term predictions. Then, I'll identify some recent books that make the unemployment prediction and illustrate it with one human activity that is about to be automated. Finally, I'll consider whether this is a utopic or dystopic view of the future.

"In the Long Run, We're All Dead"

Regardless of what John Maynard Keynes meant by this famous quip, many people take it to justify focusing on the short term. Besides, many predictions about the future are notoriously wrong. Since we have to live in the present, the argument goes, we should just focus on the short term. For example, US business focuses on the next two quarters, and US politics on the next two years. Even science is suspicious of long-term predictions. Any scientist who makes a prediction about what will happen in two decades could be accused of spouting science fiction.

Long-term predictions are not only difficult to get correct, they often predict scenarios that people don't like. Some business interests don't like the prediction of ecological collapse, and some who are comfortable with the status quo don't like predictions of social change. Thus, there are economic and social reasons to argue against any long-term prediction you don't like, regardless of its likelihood. Because long-term predictions only have likelihoods, a better phrase to describe "predicting the future" might be "analyzing future scenarios." In spite of disagreements over likelihoods, thinking about the future and analyzing various scenarios could help keep us from backing into the future. Given that the future usually contains buzz saws, backing in is never a good idea.

 Published by the IEEE Computer Society 2469-7087/16/$33.00

Ned Ludd Redux?

In the early days of the industrial revolution in England, when cottage spinning and weaving were being displaced by textile factories (aka satanic mills), Ned Ludd reportedly destroyed some of the automated machinery. In the years that followed, other disenfranchised textile workers followed suit, calling themselves Luddites. Ever since, people who resist any automation have been called Luddites. The rise of the textile industry was relatively quick[1] and might have been like throwing the frog (Ned and his imitators) into hot water. On the other hand, the rise of the post-industrial world is slower (or at least less visible), raising the water temperature a degree at a time. For example, current US unemployment (37 percent of adults are not in the labor force now compared to 33 percent in 2007) is widely interpreted as due to causes other than automation, such as workers voluntarily leaving the labor force.[2]

However, in the past two years, seven books (and probably others that I haven't read) have been published warning about and predicting the systemic unemployment scenario. As an aside, only one of these was written by a card-carrying economist. It seems that many economists hew to the party line that capitalism has always destroyed old jobs (creative destruction at work), but that it always creates more new jobs than were destroyed. In other words, past performance is used to predict the future.

The unemployment-predicting books are *The Second Machine Age* (Eric Byrnjolfsson and Andrew McAfee, Norton, 2014); Average Is Over (Tyler Cowen, Dutton, 2014); *The Second Intelligent Species* (Marshall Brain, BYG Publishing, 2015); *Rise of the Robots* (Martin Ford, Basic Books, 2015); *Surviving AI* (Calum Chace, Three Cs, 2015); *Humans Need Not Apply* (Jerry Kaplan, Yale University Press, 2015); and *Machines of Loving Grace* (John Markoff, Ecco, 2015). The authors are an interesting assortment: Byrnjolfsson and McAfee are professors of management at MIT who study the economics of IT automation. Cowen is a professor of economics at George Mason University, where, among his many interests, he studies public choice economics. Brain is a computer scientist who, among other things, created the How Stuff Works website. Ford is a Silicon Valley entrepreneur who has written two books and several articles on automation-related unemployment. Chase is a businessman turned author. Kaplan is a Silicon Valley entrepreneur and futurist, and Markoff is a science and technology journalist.

As a special case, Cowen's book covers a shorter term and focuses on IT-related underemployment. He says that for the next several decades, 30 percent of workers—those who can work with the machines—will do well, but that 70 percent will be stuck in a downward spiral (hence, "average is over"). However, the other authors are IT specialists of one stripe or another and are quite aware of the revolutionary advances currently being made in IT and related technologies that could radically modify the employment picture. I encourage interested readers to pick up one or more of these books and judge for yourself. They are all good reads, and if you want a suggestion, start with the most recent, *Humans Need Not Apply*.

Utopia or Dystopia?

For one example of recent progress in automation, consider "driverless cars." Navigating a car in heavy traffic was, until recently, thought to be too complex for computers, but it is arguably now done more reliably by a computer than by a human. This probably means that approximately 1.5 million truck drivers and 200,000 taxi drivers will be out of work in a couple of decades. I speculate that it could also mean that traffic deaths will decrease from 30,000 a year to 3,000 once most cars are driverless. By the way, the name "driverless car" might come to sound as silly as "horseless carriage" does today. Moreover, car ownership might disappear as "driverless Uber rides" become ubiquitous.

This example illustrates the pluses and minuses of automation. If traffic deaths are cut to one tenth of their present number, it would, of course, be a great benefit for society. And if calling for rides replaces car ownership, a significant cost savings for families could result. On the other hand, don't expect car makers (or dealers or repair shops or gas stations) to cheer, because the total number of cars sold might also be one tenth of today's numbers. What about truck and taxi ex-drivers and ex-car makers? There might be nowhere for them to look for a new job if most other job categories are also being automated out of existence.

If many jobs are automated, reducing the number of hours in the work week could help spread the remaining work and reduce unemployment.

If many, but not most, jobs are automated, perhaps reducing the number of hours in the work week could help spread the remaining work and reduce unemployment.

Sweden is currently pursuing this policy.[3] However, if or when most jobs are automated, society would face (at least) three big problems. To quote Voltaire in *Candide*, "Work saves us from three great evils: boredom, vice, and need." To accommodate need, some policy makers see a guaranteed annual income as a solution. Switzerland is already considering this option;[4] however, in the US, the very thought raises cries of socialism. Other more capitalistic policy responses include giving each citizen a stock and bond account. Perhaps if or when business determines that jobless consumers need cash to continue buying their goods and services, our mythology of rugged individualism will be adjusted accordingly.

Assuming for the sake of argument that "need" could be relieved by a guaranteed annual income or some other policy, boredom and vice might pose equally large problems. Whether you were one of the lucky ones who had a meaningful career or one of those who worked a meaningless job out of necessity, once you're out of a job your "idle hands are the devil's workshop," or so we are told by various proverbs. However, there are also optimistic views regarding this possible future.

The ancient Athenians believed that since their slaves performed the required labors of society, they were freed for the work of citizenship. Perhaps our silicon slaves could do likewise for us? Even more utopically, Hans Moravec wrote in his book *Robot*[5] that our genes, with their millions of years of hunting and gathering experience, will be happy to get back to hunting, fishing, and picking berries every day, not just on weekends.

In sum, should this unemployment scenario come to pass, policy makers will have as big a job moving us to such a brave new world as they did 200 years ago, when they moved us from farms to offices and factories.[6]

Should this unemployment scenario come to pass, policy makers will have as big a job moving us to such a brave new world as they did 200 years ago.

References

1. S. Beckert, *Empire of Cotton: A New History of Global Capitalism*, Allen Lane/Penguin, 2014.
2. M. Woolhouse, "Percentage of Those in Labor Pool at 38-Year Low," *Boston Globe*, 3 July 2015; https://www.bostonglobe.com/business/2015/07/02/percentage-workers-labor-pool-falls-year-low/zfLQrKjCyhra95v8PJxWcI/story.html.
3. H. Matharu, "Sweden Introduces Six-Hour Work Day," *Independent*, 1 Oct. 2015; www.independent.co.uk/news/world/europe/sweden-introduces-six-hour-work-day-a6674646.html.
4. "Swiss to Vote on 2,500 Franc Basic Income for Every Adult," Reuters, 4 Oct. 2013; www.reuters.com/article/2013/10/04/us-swiss-pay-idUSBRE9930O620131004#SePoC7Qf36KYBe6G.97.
5. H. Moravec, *Robot: Mere Machine to Transcendent Mind*, Oxford Univ. Press, 1998.
6. K. Polanyi, *The Great Transformation: The Political and Economic Origins of Our Time*, Beacon, 2001.

George Strawn *is the former director of the National Coordination Office for the Networking and Information Technology Research and Development Program (NITRD). He is now retired. Contact him at gostrawn@gmail.com.*

This article originally appeared in IT Professional, *vol. 18, no. 1, 2016.*

SAMSUNG

Galaxy S7 edge

Simple wins when your teams start working with a real work phone.

This is the Galaxy S7 edge — the phone that lets everyone work faster and smarter. It has expandable storage so your teams can share monster files, and edge panels that keep your favorite apps just one swipe away. It also comes with IP68 water and dustproof certification, which means you'll see fewer phones die. Just some of the new productivity tools at your fingertips.

SAMSUNG.COM/BUSINESS

*Portion of memory occupied by existing content. Embedded microSD card slot supports microSD cards up to 200GB (sold separately).
**Water-resistant and dustproof based on IP68 rating, which tests submersion in clear water up to 1.5 meters for up to 30 minutes.