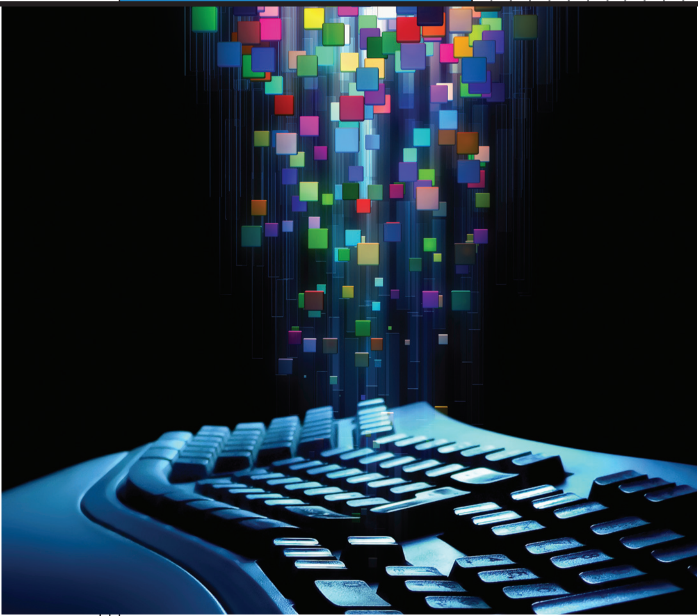

Everything as a Service: Powering the New Information Economy

Prith Banerjee, Cullen Bash, Rich Friedrich, Patrick Goldsack, Bernardo A. Huberman, John Manley, Chandrakant Patel, Partha Ranganathan, and Alistair Veitch, *Hewlett-Packard Laboratories*

The mobility/cloud ecosystem aims to deliver infrastructure, platforms, and software as a service, enabling more people to benefit from ubiquitous information access. The culmination of 10 years' research converges on-demand applications with the infrastructure—servers, storage, networks, and client devices—to support cloud computing.

A consumer in San Francisco hunts for a restaurant's address, a small-business woman in Paris checks textile prices in Bangalore, and a financial services executive in London studies global stock market trends. For each of these individuals, information at the moment of decision is critical to developing insights that can lead to the best outcome.

Few would debate that information is one of the 21st century's most valuable resources, and that an insatiable worldwide need for real-time information delivery exists. These realizations have fueled Hewlett-Packard's strategy to converge cloud computing and mobile personal information devices. The result of this decade-long effort is the *mobility/cloud ecosystem*, which promises to deliver personalized experiences through a scalable and secure information infrastructure.

In recent years, the information technology industry has begun to design cloud computing systems that deliver everything as a service—from resources to personal interactions. Morgan Stanley projects that the future mobile Internet will be 10 times larger than the desktop Internet, connecting more than 10 billion devices ranging from smartphones to wireless home appliances.[1] Information access will then be as ubiquitous as electricity. HP and large teams of customers and partners are attempting to transform economies of scale to significantly increase the population that can benefit from the information economy.

The research agenda to achieve this goal is both broad and deep, with the ultimate aim of delivering

- technologies for servers, storage, networking, and IT management that can deliver *infrastructure* as a service;
- technologies for a shared cloud infrastructure that provides enough enterprise-grade security, scalability, and quality of service to deliver *platforms* as a service; and
- novel consumer and enterprise services for the cloud to deliver *software* as a service.

Modern mobile operating systems, like Palm WebOS, combined with the ability to deliver secure, scalable computing resources economically on demand are bringing this vision to life. HP already provides millions of Internet users with digital photography, on-demand books and magazines, streaming music, and thousands of WebOS mobile phone applications. The next step is to build and expand a secure and scalable cloud infrastructure using

 Published by the IEEE Computer Society

Figure 1. The continuum of everything-as-a-service information economy. The mobility/cloud ecosystem aims to provide both an infrastructure and products to enable enterprise-grade security for the delivery of personalized services.

servers, storage, networks, and client devices such as printers, PCs, and mobile phones. This convergence of technology and services will be the basis for the next generation of connected devices, which will provide extremely high performance at a relatively low cost, thereby advancing the user experience and simplifying interactions to support personalized access to information and services anywhere, anytime.

INFRASTRUCTURE AS A SERVICE

As Figure 1 shows, infrastructure as a service offers both storage and computing services. Typically, computing services offer virtual machine instances on which customers can install and run whatever software they choose. Storage services are more complicated because of the many ways that providers can offer services to applications, such as block devices, file systems, and databases, and all their variations.

Scalable computing services

Computing infrastructure is undergoing a revolution characterized by large-scale datacenters with millions of users accessing thousands of servers. Such datacenters obviously present unique challenges for server design, and the ability to scale server configurations is an important requirement. Datacenter infrastructure—including power and cooling—can be one of the largest capital and operating expenses for cloud companies, which strongly motivates a focus on the sweet spot of commodity pricing and energy efficiency.

In addition, software stack innovations, such as scalable storage, allow the exploration of novel approaches targeted at cloud datacenters. As part of our research on exascale datacenters, we have developed workloads and metrics to use in building and analyzing new system architectures optimized for the cloud. The "A Framework to Reduce Datacenter Cost" sidebar describes the energy savings possible by applying our techniques.

Figure 2 shows our microblade and megaserver architecture, which we have optimized for the cloud.[2] Microblades are modular cost-effective server blocks, which designers can use to build megaservers—large, powerful computing environments. Features include computing blades that use embedded/mobile processors and memory blades that use flash-based nonvolatile memory. This approach provides a common second-level memory pool and physical packaging optimized for power and cooling. Indeed, evaluation results show that microblades and megaservers could improve energy efficiency by a factor of four to six relative to current technology.[2]

Our cross-layer power management scheme also targets energy efficiency.[3] Most solutions for optimizing efficiency focus on individual system design optimizations, and the emergent behavior from such an uncoordinated collec-

A FRAMEWORK TO REDUCE DATACENTER COST

Cloud services have the potential to change existing business models and deliver a net positive impact by reducing the consumption of the global energy pool. However, to reach the price point at which it is feasible to have billions of users accessing millions of services, research must revisit the total cost of ownership (TCO) of the cloud's physical infrastructure. This is particularly true in growth economies, where the desired cost of Internet access is roughly a dollar per month. The cloud computing community has made some progress toward reducing the cost of access devices,[1] but it has largely failed to address the cost to access services. Without addressing the cost of datacenters—the foundation for services to the masses—scaling to billions of users will be impossible.

In previous work,[2] we found that a significant fraction of the datacenter TCO comes from the recurring energy consumed in datacenter operation and from the burdened capital expenditures associated with the physical infrastructure. We estimated that the burdened cost of power and cooling, including redundancy, is 25 to 30 percent of the TOC in typical enterprise datacenters.[2] These power and cooling infrastructure costs can match, or even exceed, the cost of the datacenter's IT hardware. Thus, including the cost of IT hardware, more than half of a typical datacenter's TCO is associated with design and management of the physical infrastructure. For cloud service providers, with thinner layers of software and licensing costs, the physical infrastructure could be responsible for as much as 75 percent of the TCO.

With this cost in mind, HP developed a framework for designing cloud datacenters that is based on the key sustainability principle of supply- and demand-side management.[3] On the supply side, the design should aim to minimize the energy required to extract, manufacture, mitigate, transport, operate, and reclaim components. Design and management should use local sources of available energy to minimize the destruction of available energy in transmission and distribution and exploit the energy in waste streams, such as exhaust heat from a turbine.

On the demand side, the design should aim to minimize energy consumption by provisioning resources according to the users' needs by implementing flexible building blocks, pervasive sensing, communications, knowledge discovery, and policy-based control. Figure A shows an energy comparison before and after applying our techniques. Results to date show a 41 percent reduction in the life-cycle energy footprint and a 48 percent reduction in TOC. If every datacenter worldwide applied these techniques, the savings over the next three years would exceed 650,000 terajoules (TJ)—more than the total energy that all of Peru produced in 2006.

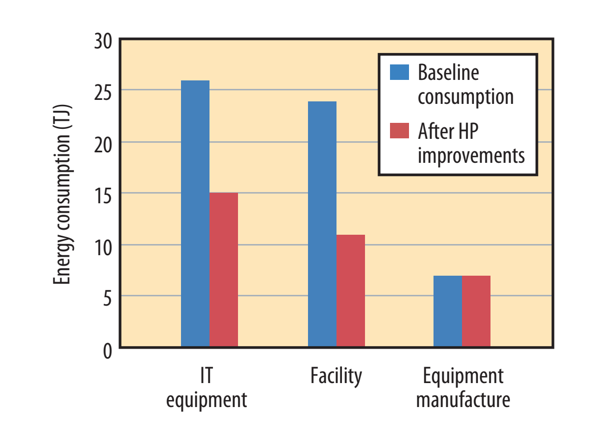

Figure A. Datacenter energy consumption over a three-year life cycle. If implemented globally on all datacenters, the proposed framework could save roughly 650,000 terajoules over three years.

In our framework, datacenter designers provision resources for sustainable cloud services according to the users' requirements in the service-level agreement. Cloud service designers then decompose these requirements into lower-level metrics that they can use to allocate IT microgrid power and cooling resources at the cloud infrastructure level. As an example, consider a cloud datacenter with a power microgrid using locally sourced wind, sun, biogas, and natural gas. For a given noncritical service, the center might queue up jobs for a time-variant supply source, such as wind and solar electricity. The same center could also execute jobs that require immediate servicing by using, say, biogas from dairy farm manure.[4] Indeed, with a microgrid of locally sourced power and cooling resources, even a net zero datacenter can supply cloud services. (A net zero datacenter is one that does not draw any power from the utility and runs at lowest cost using alternative power supplies.)

To achieve this dual vision of improved ecosystem sustainability and reduced service delivery cost, we have identified four points within the supply-demand framework that future research must consider:

- optimization of life-cycle design and cloud infrastructure size,
- monitoring of business services and correlation to performance and sustainability metrics,
- global workload scheduling according to sustainability and performance policies, and
- integration of IT demand management with resource supply constraints.

The first area is important to ensure that the infrastructure is designed for optimum performance throughout its life cycle, not just during peak operation; in the second area, the results of correlating services to performance and sustainability metrics could be provided to users to assist them in purchasing decisions.

Progress in all these areas will open up avenues for exploiting the unique attributes of the cloud infrastructure, particularly with respect to resource sharing, which in turn can reduce the overall cost of service delivery and provide a more sustainable solution.

References

1. A. Chopra, "$35 Computer Taps India's Huge Low-Income Market," *Christian Science Monitor*, 2010; www.csmonitor.com/World/Asia-South-Central/2010/0723/35-computer-taps-India-s-huge-low-income-market.
2. C. Patel and A. Shah, *Cost Model for Planning, Development, and Operation of a Datacenter*, tech. report HPL-2005-107R1, HP Laboratories, 2005.
3. C. Patel, "Sustainable Ecosystems: Enabled by Supply and Demand Management," *Proc. Int'l Conf. Distributed Computing and Networking* (ICDCN 11), LNCS 6522, Springer, 2011, pp. 12-28.
4. R. Sharma et al., "Design of Farm Waste-Driven Supply Side Infrastructure for Datacenters," *Proc. ASES Nat'l Solar Conf.*, American Solar Energy Soc., 2010, pp. 523-530.

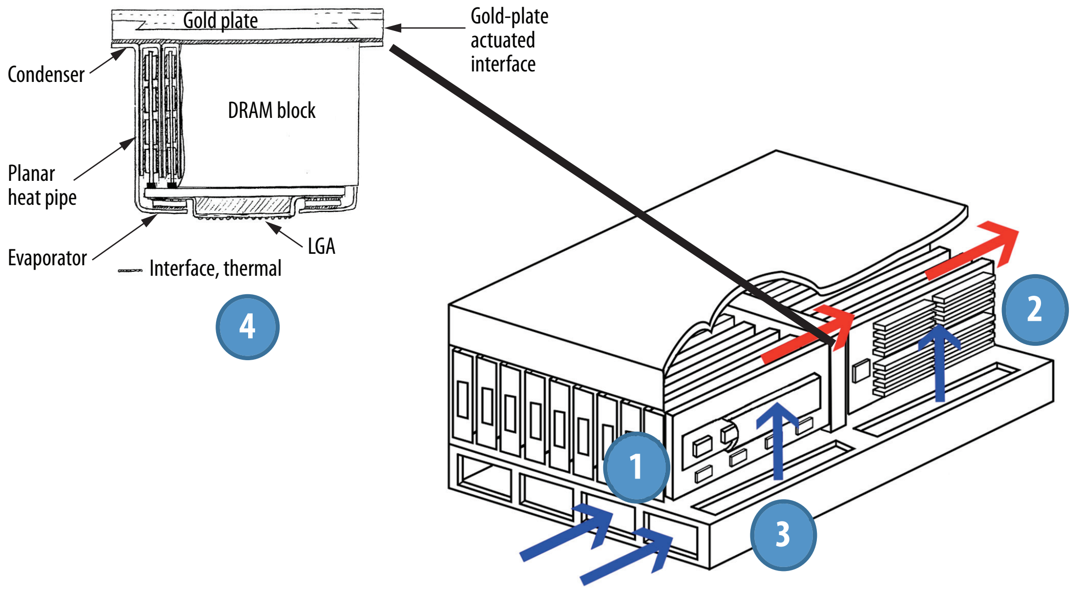

Figure 2. **Elements of a megaserver enclosure: (1) densely packaged microblades; (2) disaggregated memory blades that provide a second-level memory and novel cooling and packaging designs, such as (3) directed vertical airflow and heat removal using aggregated microblades; (4) cross-sectional view of the megaserver. Arrows indicate the direction of cooling (blue) and heat dissipation (red) airflow.**

tion is often neither stable nor correct. A more efficient approach is to create a flexible, extensible coordination framework, carefully designed to minimize the need for global information exchange and central arbitration.

Through a collaboration of computer scientists, thermal mechanical engineers, and control engineering experts, HP has developed a design that addresses this coordination need. The idea is to carefully connect and overload the abstractions in current implementations so that individual controllers can learn and react to the effect of other controllers in the same way they would respond to variations in workload demand. In addition, the design allows a formal mathematical analysis of stability and provides flexibility to dynamic changes in the controllers and system environments.

HP is continuing to look ahead to the next generation of *datacentric* workloads and novel supporting infrastructures. Emerging nonvolatile memories like memristors combined with advances in photonics and multicore processing offer intriguing opportunities for new system designs, such as nanostores,[4,5] that could offer significantly better performance and energy efficiency. These improvements in future system architectures will pave the way for applications previously not possible in the cloud, enabling a more sophisticated generation of insights across diverse data sources.

Scalable storage services

The core problems associated with building any cloud storage system are reliability, scalability, and cost-effectiveness. We believe that reliability (which encompasses both the durability of the data and its availability for access) is the primary property that users desire from such a system. Whenever a cloud service of any popularity becomes unavailable, or loses data, it quickly becomes front-page news because the service is vital to so many users and businesses. Compounding the problem is the number of possible failure scenarios; to meet scale and cost-effectiveness goals, such storage systems are built from clusters comprising commodity servers, disk drives, and networks that are spread over multiple datacenters. Failures can stem from individual disk and node failures, network infrastructure outages, power distribution outages, and even disaster scenarios that might render an entire datacenter unusable. To ensure overall system reliability, there must be enough data redundancy to enable recovery from all these failure scenarios.

The Consistent Available, Partition-Tolerant theorem, popularized by Eric Brewer,[6] characterizes three properties that distributed system developers must trade off: consistency (Will the read results be consistent with the most recent write?), availability (Can I access my data?), and partition tolerance (Can my system tolerate a sudden split into multiple parts?). The theorem essentially states that a distributed system can offer only two of the three properties.

Because availability is so important, many cloud systems optimize for this property, even with network partitions, and give up on consistency in various failure scenarios. A substantial part of our research has been on

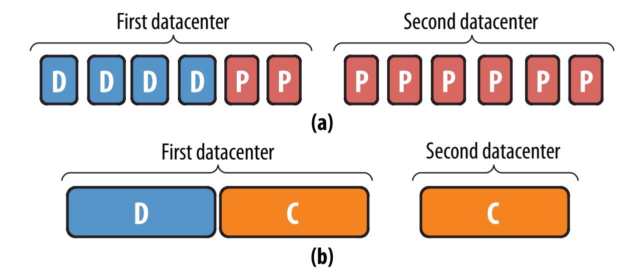

Figure 3. Erasure coding versus three-way replication. (a) Erasure coding splits an object into fragments, calculating redundant parity fragments (P). The system can then use a subset of these fragments to re-create the original data (D). (b) Three-way replication creates two copies (C) of the data and thus uses the same overall capacity but can tolerate far fewer failures.

defining and quantifying the ability of storage systems to maintain consistency.[7,8] A storage system must be able to provide different kinds of consistency under different operating conditions, yet developers of cloud-based applications have only a vague idea of the semantics that the underlying storage system can offer in various scenarios.

Data replication is the traditional approach to achieving reliability, but the need to repeat replication drives up the storage system's underlying cost. Our research over the past several years has focused on architecting and prototyping scalable storage systems for the cloud. Our goals for this research are ambitious: more than 1 exabyte across geographically distributed datacenters, 99.999 percent availability, zero data loss, and low cost.

With these goals in mind, we have developed a system that uses erasure-coded storage with eventual consistency semantics that encode data into $n = k + m$ fragments, allowing the system to use any k fragments to recover the object. Figure 3 illustrates erasure coding relative to three-way replication. Using the 4-of-12 erasure code in Figure 3a, the system can use any four of the 12 fragments to re-create the data. Spread across two datacenters, this encoding can tolerate the loss of any eight fragments—an entire datacenter—and two fragments in the second center. The three-way replication in Figure 3b can tolerate far fewer failures while using the same amount of overall capacity. Replication is far simpler to implement and reason about in building a distributed system.

Although erasure coding complicates system implementation, the benefits of greater data reliability at reduced cost outweigh implementation concerns. Moreover, system users can decide on a specific cost-capacity-reliability tradeoff by specifying a particular encoding. For example, one user could opt to store temporary, easily re-created data with minimal redundancy and cost; another could elect to widely disperse archival data to ensure complete reliability.

The downside is that erasure coding substantially complicates the protocols required for correct behavior, so one of our main challenges was to define efficient erasure-coded protocols.[9] We had to ensure correctness and availability given partial writes (the system updates only some fragments), ensure recovery after storage node crashes, and account for network partitions and multiple writers and readers at multiple sites.

To the best of our knowledge, our system is the first of its type. Our initial approach has focused on building a key-value storage scheme, in which a value comprises an arbitrary string of bytes that represent the object to be stored. Each value has a key, which is a relatively small byte string. Core operations for such a system are put(key, value), which associates a value with a key; get(key), which retrieves the associated value; enumerate(), which retrieves the keys used; and delete, which removes an object and its key from the system. These operations comprise the base set, which our future research will expand to duplicate the richness of application programming interfaces, including notions of users, security, and so on.

To test our key-value system prototype, we developed an implementation framework that enables code testing in both a simulated environment and on real hardware. We have used the framework to evaluate the system under a wide variety of failure conditions, including partitions, 50 percent random packet loss, disk failures, node failures, and rack failures. In all cases, the system has maintained availability. To ensure that the system behaves as expected, we have deployed instances across multiple continents.

PLATFORM AS A SERVICE

Offering a platform as a service means providing an enterprise-grade cloud computing infrastructure—a service that enforces quality-of-service guarantees over the security, isolation, reliability, and performance of the virtualized infrastructures that it generates and manages. Given a physical infrastructure of computing nodes, storage devices, and interconnecting networks, the fundamental requirement is to provide service providers with the illusion of a unique, secure infrastructure that supports many concurrent tenants while meeting all performance and security requirements. In addition, an infrastructure service must support a set of enterprise-grade properties, including privacy and security, quality of service and performance, flexibility, scalability, and resilience to failure.

To provide flexibility within these virtual infrastructures, a service running in the cloud must be able to scale its resource use up or down dynamically to cope with changing workloads. Service providers will want to limit such scaling for a variety of reasons, such as to maintain cost, eliminate a runaway service, or mitigate a service infected by a virus. Consequently, any infrastructure must have secure out-of-band mechanisms so that service providers can implement limits on flexing and other changes that automated infrastructure management must enforce.

The infrastructure must also allow cloud services to isolate security and performance and protect their data and sensitive information from other services, as well as provide performance guarantees regardless of what other services share an infrastructure. Finally, service management and core infrastructure management must be separate so that services cannot interfere with infrastructure management operations.

With this separation of infrastructure management and service management, there must be a clear separation of concerns between the parties to decide on policies around failure recovery and quality of service to eliminate unexpected service behavior. Consequently, the infrastructure service needs to provide mechanisms for the service provider to push policies into the infrastructure service, detailing aspects it wants to delegate to the service. Without this, enterprises and governments would not trust cloud infrastructure services.

To meet these objectives, we took a *cells-as-a-service* approach,[10] introducing a class of virtualized infrastructure services in which the user declares the desired topology of the virtual machines, virtual block storage, and virtual networking. The infrastructure service is responsible for creating an instance of that topology, and the topology must satisfy a set of constraints, such as quality of service and permitted communication patterns. Both the topology and constraint descriptions can change dynamically in response to load or failure, and the cell will continually adapt to meet the changing requirements.

CELLS AS A SERVICE

The cells-as-a-service approach is built on the fundamental concept of a cell. Cells contain virtual machines (VMs), storage volumes, and subnets—all of which are declared as cell model elements. The model also describes how these components connect to create the desired virtual infrastructure. Each definition of a component or connection includes a set of relevant attributes. For example, VM elements include specifications for memory requirements, bus addresses, subnet connections, and behavior in the event of failure. An XML document contains the cell model description. The infrastructure designer can handle model changes by submitting an updated model document or by using an API that supports incremental model changes. The cell controller is responsible for securely interacting with the service provider and for monitoring the virtual infrastructure's status.

The underlying system secures the cell's boundary, or any separation of cell components. This boundary consists of both network connectivity between hosts and subnets of the cell and between cells, plus the ability to mount volumes owned by other service providers. By default, network traffic is only permitted between VMs on the same subnet, and volumes are only visible for connections and imaging within the same cell. Security may be relaxed in a controlled way by adding rules to the cell model to share volumes with other cells and open network connections from a subnet or perhaps only an individual VM to other subnets or VMs in the same cell. Network connections may also be opened between VMs and subnets in different cells provided both cells contain reciprocal rules.

Cell management is the job of the *system cell*, a privileged cell responsible for creating and deleting all the virtual components and managing their connectivity, enforcing all the connectivity policies defined within each cell, cell interaction, and enforcing any policy on recovery or scalability limits associated with a cell. No part of any other cell can communicate directly with the system cell apart from a locked-down and secure bastion component of each cell controller.

Web 2.0 and the cloud have given rise to a new class of services that cater to an increasingly connected population.

The system cell runs across all physical hosts, each of which must be running a hypervisor. The system cell contains host managers and core system services, such as resource management and storage management. A host manager runs on each physical host within the privileged VM (the host OS) and is responsible for managing and validating every action that occurs on that physical host.

Each host manager enforces the isolation of cells from each other and from the system cell by mediating access to the physical host's computing, network, and storage capabilities and by transforming abstract VM elements into configuration data appropriate to the underlying hypervisor. The host manager interacts with a storage manager to create and remove virtual block devices as the hosted VMs require.

We implement cell subnets as virtual overlay networks on a single shared physical network.[11] Implementation requires no special hardware; since we use a novel fully distributed virtualized router that facilitates single network hop communication between end points. Unlike traditional software routers, cells-as-a-service operates at the open systems interface network layer, allowing packets to be forwarded directly to their destination. To support the requirement to manage overall performance, the networking layer provides networking resource control to limit and prioritize the VM's bandwidth consumption.

SOFTWARE AS A SERVICE

Web 2.0 and the cloud have given rise to a new class of services that cater to an increasingly connected population. Combining sociology, economics, and computer science, HP aims to create models, methods, and technologies to harness the flow of collective attention, supporting a mobile society with context-aware and anticipative solutions. The ultimate goal is to build a fluid enterprise that captures collective intelligence for a variety of uses—from predicting the future to allocating resources. Some of our projects are ePrint, Rankr, i-Catcher, and Watercooler.

The *ePrint* platform aims to break down the barriers of distance and connectivity and empower people to use their mobile devices to send files they want to print. Users could be anyone, and the possible applications are limitless—for example, a mother and son can print drawings from an iPad or an executive can use his Palm Pre or BlackBerry smartphone to send a presentation to a Federal Express store to print and hold for pick up.

Rankr is a method for swiftly aggregating collective wisdom, such as views on political candidates. The method is so named because we created and deployed it as a webservice for deriving a rank ordering of multiple agendas, objects, suggestions, or websites that use pairwise comparisons. Unlike typical rank voting methods, voters do not need to compare and manually rank all the candidate items. Given the votes that others have already cast, Rankr automatically determines the most useful pair of candidates a user can evaluate to maximize the information gained while minimizing the number of votes required. Consequently, Rankr scales beyond traditional voting schemes while enabling the crowd-sourced ranking of many more items than a single user is likely to evaluate.

In recent years, social media has become ubiquitous, yet the content generated from these websites remains largely untapped. To address that need, we recently demonstrated a service that analyzes the allocation of attention within social media to predict real-world outcomes. We have already tested it by using the chatter from Twitter to forecast box-office revenues for movies and found that the rate at which tweets are created on particular topics can outperform market-based predictors.[12] We also demonstrated how further exploitation of sentiments extracted from Twitter can improve social media's forecasting power. Our methodology is general enough that users can apply it to any accessible social medium to predict trends in products and services, as well as in other areas.[12]

i-Catcher is a technology developed for increasing the attention devoted to content in any website. It relies on dynamic measurements of the rate at which the content's novelty and popularity change. The initial download rates are high enough that we can accurately predict the content's long-term popularity. Using two content-sharing portals, YouTube and Digg, we showed that by measuring the initial rate at which users view and vote on content, we can predict the long-term popularity of the submissions. In Digg, measuring access to given stories during the first two hours allowed us to forecast their popularity 30 days ahead with remarkable accuracy. We had to follow YouTube video downloads for 10 days to attain the same performance.[13]

Transferring expertise across an organization is difficult. Meanwhile, consumers have adopted distributed Internet services as a means of sharing and finding information. Web 2.0 services like Digg and Facebook let individuals discover resources from their social networks, and a new generation is entering the workforce expecting to collaborate the same way at work—more efficiently, rapidly, and at a lower cost. To meet that expectation, we developed *Watercooler*, an enterprise collaboration 2.0 technology that gives people better filters than their explicit (and binary) friend networks and thus allows users to explore the knowledge in distant parts of their organization. More than 130,000 users within HP are already benefiting from this technology.

Advances in information technology during the past 20 years have made cloud computing a reality. We have made tremendous progress, but many challenging problems remain. One is how to secure services, data, and the infrastructure from attack. Another is how to ensure the privacy of personal data. Flexible and dynamic resource allocation must occur in real time based on events or policy and on a scale that no one has yet implemented. In large, complex systems such as these, failed components are a common occurrence, and thus services must remain available to clients regardless of hardware or software failure or disruption. Performance must be predictable across a wide range of workload demands while maintaining acceptable economics in service delivery. New system architectures, programming models, development environments, and testing and debugging methodologies will be required for dynamically instantiated, distributed, self-managing, and ephemeral services.

The next generation of scientists and engineers must be prepared to create and deliver these advances. Without a continual push forward, the momentum of this new cloud computing and mobility convergence will stall.

Acknowledgments

The research described in this article is the work of more than 100 researchers in the Cloud and Security Lab, Intelligent Infrastructure Lab, Sustainable Ecosystems Group, and Social Computing Group. A complete listing of the researchers involved is available at www.hpl.hp.com.

References

1. Morgan Stanley, *The Mobile Internet Report*, 15 Dec. 2009. pp. 84-125.
2. K. Lim et al., "Server Designs for Warehouse-Computing Environments," *IEEE Micro*, vol. 29, no. 1, 2009, pp. 41-49.
3. R. Raghavendra et al., "No 'Power' Struggles: Coordinated Multi-level Power Management for the Datacenter," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 08), ACM Press, 2008, pp. 48-49.
4. N. Jouppi and Y. Xie, "Emerging Technologies and Their Impact on System Design," *Proc. 14th Int'l Symp. Low Power Electronics and Design* (ISPLED 09), ACM Press, 2009, pp. 427-428.
5. P. Ranganathan, "From Microprocessors to Nanostores: Rethinking Data-Centric Systems," *Computer*, Jan. 2011, pp. 39-48.
6. S. Gilbert and N. Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services," *ACM SIGACT News*, vol. 33, no. 2, 2002, pp. 51-59.
7. A. Aiyer et al., "Consistability: Describing Usually Consistent Systems," *Proc. 4th Workshop on Hot Topics in System Dependability* (HotDep 08), Usenix, 2008; www.usenix.org/event/hotdep08/tech/full_papers/aiyer/aiyer.pdf.
8. E. Anderson et al., "What Consistency Does Your Key-Value Store Actually Provide?" *Proc. 6th Workshop Hot Topics in System Dependability* (HotDep 10), Usenix, 2010; www.usenix.org/event/hotdep10/tech/full_papers/Anderson.pdf.
9. E. Anderson et al., "Efficient Eventual Consistency in Pahoehoe: An Erasure-Coded Key-Blob Archive," *Proc. Int'l Conf. Dependable Systems and Networks* (DSN 10), IEEE CS Press, 2010, pp. 181-190.
10. A. Coles et al., *Cells-as-a-Service: A Cloud Infrastructure Service*, tech. report HPL-2011-7, HP Laboratories, Jan. 2011.
11. A. Edwards, A. Fischer, and A. Lain, "Diverter: A New Approach to Networking within Virtualized Infrastructures," *Proc. 1st ACM Workshop Research on Enterprise Networking* (WREN 09), ACM Press, 2009, pp. 103-110.
12. S. Asur, and B. Huberman, "Predicting the Future with Social Media," *Proc. Int'l Conf. Web Intelligence and Intelligent Agent Technology*, ACM Press, 2010, pp. 492-499.
13. G. Szabo and B. Huberman, "Predicting the Popularity of Online Content," *Comm. ACM*, Aug. 2010, pp. 80-88.

Prith Banerjee *is HP's senior vice president of research and director of Hewlett-Packard Laboratories. In these roles, he assists in charting technical strategies for the company and manages all seven HP Laboratories worldwide. Banerjee received a PhD in electrical engineering from the University of Illinois. He is a Fellow of the American Association for the Advancement of Science, the ACM, and IEEE. Contact him at prith.banerjee@hp.com.*

Cullen Bash *is a Distinguished Scientist at HP. His research interests include thermomechanical architectures of future systems and fluid mechanics and heat-transfer processes in datacenters. Bash received an MS in mechanical engineering from the University of California, San Diego. Contact him at cullen.bash@hp.com.*

Rich Friedrich *is the director of Strategic Innovations at HP, where he is responsible for applying open innovation to amplify and accelerate research results and technology transfer. His research interests include personalized user experiences, secure cyberinfrastructure, and the visualization of large, complex data sets. Friedrich received a BS in electrical engineering and computer science from Northwestern University. He is a senior member of IEEE. Contact him at rich.friedrich@hp.com.*

Patrick Goldsack *is a Distinguished Scientist in HP's Cloud and Security Laboratory. His primary research interests are in large-scale distributed computation and computer languages. Goldsack received an MS in mathematics and in electrical engineering from the University of Oxford. Contact him at patrick.goldsack@hp.com.*

Bernardo A. Huberman *is an HP Senior Fellow and director of the Social Computing Laboratory and a consulting professor in the Department of Applied Physics at Stanford University. His research interests include the interaction between social behavior and information technology, particularly crowd sourcing and the economics of attention. Huberman received a PhD in physics from the University of Pennsylvania. Contact him at bernardo.huberman@hp.com.*

John Manley *is director of HP's Cloud Computing Research Group. His research interests include very large-scale systems, such as knowledge-base management systems; the telecommunications management network; and utility/cloud computing. Manley received a PhD in molecular quantum mechanics from the University of Bristol. Contact him at jmanley@hp.com.*

Chandrakant Patel *is an HP Senior Fellow and director of the Sustainable Ecosystems Research Group. His research interests include smart datacenters, in which the datacenter is the computer, and the demand-driven dynamic provisioning of computing, power, and cooling resources. Patel received an MS in mechanical engineering from San Jose State University. He is an IEEE Fellow. Contact him at chandrakant.patel@hp.com.*

Partha Ranganathan *is a Distinguished Technologist at HP and principal investigator for the Data-Centric Datacenter Project. His research interests include systems architecture and energy efficiency. Ranganathan received a PhD in electrical and computer engineering from Rice University. He is a senior member of IEEE and the ACM. Contact him at partha.ranganathan@hp.com.*

Alistair Veitch *is director of HP's Intelligent Storage Group. His research interests include storage systems management and design, including demonstrations of completely self-managing storage systems. Veitch received a PhD in computer science from the University of British Columbia. He is a member of the ACM and Usenix. Contact him at alistair.veitch@hp.com.*