

ACM Tech Pack on Cloud Computing

Doug Terry
Microsoft Research
Chairman, ACM Tech Pack Committee on Cloud Computing

Cloud computing promises to radically change the way computer applications and services are constructed, delivered, and managed. Although the term means different things to different people, and includes a bit of marketing hype and technical redefinition, the potential benefits are clear. Large datacenters permit resource sharing across hosted applications and lead to economies of scale at both the hardware and software level. Software services can obtain seemingly infinite scalability and incremental growth to meet customers' elastic demands. The pay-as-you-go model and rapid provisioning can result in more efficient resource utilization and reduced costs.

Realizing these benefits requires new techniques for managing shared data in the cloud, fault-tolerant computation, service composition, scheduling, metering and billing, protecting privacy, communication, and, more generally, sharing resources among applications under the control of diverse organizations. The research community is stepping up to meet these challenges, as are a number of high-tech companies. This collection of papers highlights some early efforts in what is sure to be a productive area of innovation for years to come.

The following is a list of topics and associated published papers that focus on cloud computing. Each topic starts with a set of questions that may be of interest to both researchers and practitioners. The listed papers do not necessarily answer all of these questions, but were selected because they provide insights and introduce new relevant technologies.

Basic paradigm

Cloud computing is a fundamental new paradigm in which computing is migrating from personal computers sitting on a person's desk (or lap) to large, centrally managed datacenters. How does cloud computing differ from Web services, Grid computing, and other previous models of distributed systems? What new functionality is available to application developers and service providers? How do such applications and services leverage pay-as-you-go pricing models to meet elastic demands?

Readings:

- [Cloud computing](#)

Brian Hayes. 2008. Cloud computing. *Commun. ACM* 51, 7 (July 2008), 9-11.
DOI=10.1145/1364782.1364786
<http://doi.acm.org/10.1145/1364782.1364786>

As software migrates from local PCs to distant Internet servers, users and developers alike go along for the ride.

Discusses the trend of moving software applications into the cloud and the challenges.

- [Cloud Computing: An Overview](#)

2009. Cloud Computing: An Overview. *Queue* 7, 5, Pages 2 (June 2009), 2 pages. DOI=10.1145/1538947.1554608
<http://doi.acm.org/10.1145/1538947.1554608>

A summary of important cloud-computing issues distilled from ACM CTO Roundtables

Presents some of the key topics discussed during the ACM Cloud Computing and Virtualization CTO Roundtables of 2008.

- [CTO Roundtable: Cloud Computing](#)

Mache Creeger. 2009. CTO Roundtable: Cloud Computing. *Queue* 7, 5, Pages 1 (June 2009), 2 pages. DOI=10.1145/1551644.1551646
<http://doi.acm.org/10.1145/1551644.1551646>

Our panel of experts discuss cloud computing and how companies can make the best use of it.

Provides solid advice from a panel of experts on how organizations can benefit from cloud computing.

- [Computing in the clouds](#)

Aaron Weiss. 2007. Computing in the clouds. *netWorker* 11, 4 (December 2007), 16-25. DOI=10.1145/1327512.1327513
<http://doi.acm.org/10.1145/1327512.1327513>

Powerful services and applications are being integrated and packaged on the Web in what the industry now calls "cloud computing"

Explores the many perspectives on cloud computing and debunks the notion that it is simply a rebranding of old computing models.

- [Emergence of the Academic Computing Clouds](#)

Kemal A. Delic and Martin Anthony Walker. 2008. Emergence of the Academic Computing Clouds. *Ubiquity* 2008, August, Article 1 (August 2008), 1 pages. DOI=10.1145/1414663.1414664
<http://doi.acm.org/10.1145/1414663.1414664>

Computational grids are very large-scale aggregates of communication and computation resources enabling new types of applications and bringing several benefits of economy-of-scale. The first computational grids were established in academic environments during the previous decade, and today are making inroads into the realm of corporate and enterprise computing.

Very recently, we observe the emergence of cloud computing as a new potential super structure for corporate, enterprise and academic computing. While cloud computing shares the same original vision of grid computing articulated in the 1990s by Foster, Kesselman and others, there are significant differences.

In this paper, we first briefly outline the architecture, technologies and standards of computational grids. We then point at some of notable examples of academic use of grids and sketch the future of research in grids. In the third section, we draw some architectural lines of cloud computing, hint at the design and technology choices and indicate some future challenges. In conclusion, we claim that academic computing clouds might appear soon, supporting the emergence of Science 2.0 activities, some of which we list shortly.

Discusses the emergence of cloud computing in support of experimental sciences addressing engineering, medical, and social problems.

Storage

A central challenge of cloud computing is providing scalable, secure, self-managing, and fault-tolerant data storage for long-running services. What data models are supported by existing cloud-based storage systems? What are the technical trade-offs between the key-value stores commonly provided and relational databases? How do application developers choose a particular storage system? How does one design cloud-based storage systems to ensure that a user's data survives for 100 years, even as companies come and go?

Readings:

- [The Google file system](#)

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. 2003. The Google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03)*. ACM, New York, NY, USA, 29-43. DOI=10.1145/945445.945450 <http://doi.acm.org/10.1145/945445.945450>

Describes the design and implementation of a scalable file system that supports many of Google's large, data-intensive applications and that influenced many subsequent systems.

- [Building a database on S3](#)

Matthias Brantner, Daniela Florescu, David Graf, Donald Kossmann, and Tim Kraska. 2008. Building a database on S3. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08)*. ACM, New York, NY, USA, 251-264. DOI=10.1145/1376616.1376645 <http://doi.acm.org/10.1145/1376616.1376645>

There has been a great deal of hype about Amazon's simple storage service (S3). S3 provides infinite scalability and high availability at low cost. Currently, S3 is used mostly to store multi-media documents (videos, photos, audio) which are shared by a community of people and rarely updated. The

purpose of this paper is to demonstrate the opportunities and limitations of using S3 as a storage system for general-purpose database applications which involve small objects and frequent updates. Read, write, and commit protocols are presented. Furthermore, the cost (\$), performance, and consistency properties of such a storage system are studied.

Shares experiences building a general-purpose database system on top of Amazon's simple storage service (S3), and provides insights not only into S3 but also into the issues faced by applications that want to manage structured data in the cloud.

- [Organizing and sharing distributed personal web-service data](#)

Roxana Geambasu, Cherie Cheung, Alexander Moshchuk, Steven D. Gribble, and Henry M. Levy. 2008. Organizing and sharing distributed personal web-service data. In *Proceeding of the 17th international conference on World Wide Web (WWW '08)*. ACM, New York, NY, USA, 755-764.

DOI=10.1145/1367497.1367599

<http://doi.acm.org/10.1145/1367497.1367599>

The migration from desktop applications to Web-based services is scattering personal data across a myriad of Web sites, such as Google, Flickr, YouTube, and Amazon S3. This dispersal poses new challenges for users, making it more difficult for them to: (1) organize, search, and archive their data, much of which is now hosted by Web sites; (2) create heterogeneous, multi-Web-service object collections and share them in a protected way; and (3) manipulate their data with standard applications or scripts.

In this paper, we show that a Web-service interface supporting standardized naming, protection, and object-access services can solve these problems and can greatly simplify the creation of a new generation of object-management services for the Web. We describe the implementation of Menagerie, a proof-of-concept prototype that provides these services for Web-based applications. At a high level, Menagerie creates an integrated file and object system from heterogeneous, personal Web-service objects dispersed across the Internet. We present several object-management applications we developed on Menagerie to show the practicality and benefits of our approach.

Presents the challenges of integrating, manipulating, protecting, and sharing personal data that is distributed across a number of Web-based services, and describes a prototype system to meet these challenges.

- [Cumulus: Filesystem backup to the cloud](#)

Michael Vrable, Stefan Savage, and Geoffrey M. Voelker. 2009. Cumulus: Filesystem backup to the cloud. *Trans. Storage* 5, 4, Article 14 (December 2009), 28 pages. DOI=10.1145/1629080.1629084

<http://doi.acm.org/10.1145/1629080.1629084>

Cumulus is a system for efficiently implementing filesystem backups over the Internet, specifically designed under a *thin cloud* assumption—that the remote datacenter storing the backups does not provide any special backup services, but only a least-common-denominator storage interface. Cumulus aggregates data from small files for storage and uses LFS-inspired segment cleaning to maintain storage efficiency. While Cumulus can use virtually any storage service, we show its efficiency is comparable to integrated approaches.

Evaluates thin-cloud vs. thick-cloud performance and cost trade-offs in the context of an application that uses cloud storage to back up files.

Data consistency and replication

Most current cloud-resident storage systems replicate data but have chosen to relax consistency in favor of increased performance (and availability). What consistency guarantees that lie somewhere between strong serializability and weak eventual consistency might appeal to cloud applications? How can they be provided for cloud-based services that serve a globally distributed user population?

Readings:

- [Eventually consistent](#)

Werner Vogels. 2009. Eventually consistent. *Commun. ACM* 52, 1 (January 2009), 40-44. DOI=10.1145/1435417.1435432
<http://doi.acm.org/10.1145/1435417.1435432>

Building reliable distributed systems at a worldwide scale demands trade-offs between consistency and availability.

Explains why giving up on strong consistency is necessary when replicating data within systems that operate on a global scale, and describes some alternative consistency models.

- [Dynamo: amazon's highly available key-value store](#)

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. 2007. Dynamo: amazon's highly available key-value store. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles (SOSP '07)*. ACM, New York, NY, USA, 205-220. DOI=10.1145/1294261.1294281
<http://doi.acm.org/10.1145/1294261.1294281>

Reliability at massive scale is one of the biggest challenges we face at Amazon.com, one of the largest e-commerce operations in the world; even the slightest outage has significant financial consequences and impacts customer trust. The Amazon.com platform, which provides services for many web sites worldwide, is implemented on top of an infrastructure of tens of thousands of servers and network components located in many datacenters around the world. At this scale, small and large components fail continuously

and the way persistent state is managed in the face of these failures drives the reliability and scalability of the software systems.

This paper presents the design and implementation of Dynamo, a highly available key-value storage system that some of Amazon's core services use to provide an "always-on" experience. To achieve this level of availability, Dynamo sacrifices consistency under certain failure scenarios. It makes extensive use of object versioning and application-assisted conflict resolution in a manner that provides a novel interface for developers to use.

Presents the design of a replicated, scalable system that provides key-value storage for many of Amazon's applications, sacrifices consistency, and relies on application involvement in resolving conflicting updates.

- [How replicated data management in the cloud can benefit from a data grid protocol: the Re:GRiDiT Approach](#)

Laura Cristiana Voicu and Heiko Schuldt. 2009. How replicated data management in the cloud can benefit from a data grid protocol: the Re:GRiDiT Approach. In *Proceeding of the first international workshop on Cloud data management* (CloudDB '09). ACM, New York, NY, USA, 45-48. DOI=10.1145/1651263.1651272 <http://doi.acm.org/10.1145/1651263.1651272>

Cloud computing has recently received considerable attention both in industry and academia. Due to the great success of the first generation of Cloud-based services, providers have to deal with larger and larger volumes of data. Quality of service agreements with customers require data to be replicated across data centers in order to guarantee a high degree of availability. In this context, Cloud Data Management has to address several challenges, especially when replicated data are concurrently updated at different sites or when the system workload and the resources requested by clients change dynamically. Mostly independent from recent developments in Cloud Data Management, Data Grids have undergone a transition from pure file management with read-only access to more powerful systems. In our recent work, we have developed the Re:GRiDiT protocol for managing data in the Grid which provides concurrent access to replicated data at different sites without any global component and supports the dynamic deployment of replicas. Since it is independent from the underlying Grid middleware, it can be seamlessly transferred to other environments like the Cloud. In this paper, we compare Data Management in the Grid and the Cloud, briefly introduce the Re:GRiDiT protocol and show its applicability for Cloud Data Management.

Compares cloud data management with previous work on data grids.

- [Middleware-based database replication: the gaps between theory and practice](#)

Emmanuel Cecchet, George Candea, and Anastasia Ailamaki. 2008. Middleware-based database replication: the gaps between theory and practice. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (SIGMOD '08). ACM, New York, NY, USA, 739-752. DOI=10.1145/1376616.1376691
<http://doi.acm.org/10.1145/1376616.1376691>

The need for high availability and performance in data management systems has been fueling a long running interest in database replication from both academia and industry. However, academic groups often attack replication problems in isolation, overlooking the need for completeness in their solutions, while commercial teams take a holistic approach that often misses opportunities for fundamental innovation. This has created over time a gap between academic research and industrial practice.

This paper aims to characterize the gap along three axes: performance, availability, and administration. We build on our own experience developing and deploying replication systems in commercial and academic settings, as well as on a large body of prior related work. We sift through representative examples from the last decade of open-source, academic, and commercial database replication systems and combine this material with case studies from real systems deployed at Fortune 500 customers. We propose two agendas, one for academic research and one for industrial R&D, which we believe can bridge the gap within 5-10 years. This way, we hope to both motivate and help researchers in making the theory and practice of middleware-based database replication more relevant to each other.

Describes examples of replicated systems from academic and commercial organizations and suggests ways to bridge the gap between them in terms of performance, availability, and administration.

Programming models

Cloud computing platforms offer computing on demand but differ in the flexibility and functionality that they provide to programmers. How should computational resources in the cloud be presented to application developers, as virtualized hardware or application-specific platforms or something in between? What programming tools are available and how are they used?

Readings:

- [MapReduce: simplified data processing on large clusters](#)

Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (January 2008), 107-113. DOI=10.1145/1327452.1327492
<http://doi.acm.org/10.1145/1327452.1327492>

MapReduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a *map* and a

reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct MapReduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.

Presents the design of and experience with a popular parallel programming model for processing large data sets with efficiency and high reliability on clusters of machines at Google.

- [MapReduce and parallel DBMSs: friends or foes?](#)

Michael Stonebraker, Daniel Abadi, David J. DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, and Alexander Rasin. 2010. MapReduce and parallel DBMSs: friends or foes?. *Commun. ACM* 53, 1 (January 2010), 64-71. DOI=10.1145/1629175.1629197 <http://doi.acm.org/10.1145/1629175.1629197>

MapReduce complements DBMSs since databases are not designed for extract-transform-load tasks, a MapReduce specialty.

Argues that MapReduce compliments, rather than competes with, parallel database management systems and provides insights into the types of application workloads best suited for each.

- [Distributed data-parallel computing using a high-level programming language](#)

Michael Isard and Yuan Yu. 2009. Distributed data-parallel computing using a high-level programming language. In *Proceedings of the 35th SIGMOD international conference on Management of data (SIGMOD '09)*, Carsten Binnig and Benoit Dageville (Eds.). ACM, New York, NY, USA, 987-994. DOI=10.1145/1559845.1559962 <http://doi.acm.org/10.1145/1559845.1559962>

The Dryad and DryadLINQ systems offer a new programming model for large scale data-parallel computing. They generalize previous execution environments such as SQL and MapReduce in three ways: by providing a general-purpose distributed execution engine for data-parallel applications; by adopting an expressive data model of strongly typed .NET objects; and by supporting general-purpose imperative and declarative operations on datasets within a traditional high-level programming language.

A DryadLINQ program is a sequential program composed of LINQ expressions performing arbitrary side-effect-free operations on datasets, and can be written and debugged using standard .NET development tools. The DryadLINQ system automatically and transparently translates the data-parallel portions of the program into a distributed execution plan which is passed to the Dryad execution platform. Dryad, which has been in continuous

operation for several years on production clusters made up of thousands of computers, ensures efficient, reliable execution of this plan on a large compute cluster.

This paper describes the programming model, provides a high-level overview of the design and implementation of the Dryad and DryadLINQ systems, and discusses the tradeoffs and connections to parallel and distributed databases.

Offers another programming model for large-scale data-parallel computing based on Microsoft's LINQ platform for SQL-like queries.

- [Boom analytics: exploring data-centric, declarative programming for the cloud](#)

Peter Alvaro, Tyson Condie, Neil Conway, Khaled Elmeleegy, Joseph M. Hellerstein, and Russell Sears. 2010. Boom analytics: exploring data-centric, declarative programming for the cloud. In *Proceedings of the 5th European conference on Computer systems (EuroSys '10)*. ACM, New York, NY, USA, 223-236. DOI=10.1145/1755913.1755937
<http://doi.acm.org/10.1145/1755913.1755937>

Building and debugging distributed software remains extremely difficult. We conjecture that by adopting a *data-centric* approach to system design and by employing *declarative* programming languages, a broad range of distributed software can be recast naturally in a data-parallel programming model. Our hope is that this model can significantly raise the level of abstraction for programmers, improving code simplicity, speed of development, ease of software evolution, and program correctness.

This paper presents our experience with an initial large-scale experiment in this direction. First, we used the Overlog language to implement a "Big Data" analytics stack that is API-compatible with Hadoop and HDFS and provides comparable performance. Second, we extended the system with complex distributed features not yet available in Hadoop, including high availability, scalability, and unique monitoring and debugging facilities. We present both quantitative and anecdotal results from our experience, providing some concrete evidence that both data-centric design and declarative languages can substantially simplify distributed systems programming.

Explores a declarative approach to writing data-parallel programs that run in a cloud environment.

Virtualization

Cloud computing currently relies heavily on virtualized CPU and storage resources to meet elastic demands. What is the role of virtualization in cloud-based services? Are current virtualization technologies sufficient?

Readings:

- [Virtualizing the Datacenter Without Compromising Server Performance](#)

Faouzi Kamoun. 2009. Virtualizing the Datacenter Without Compromising Server Performance. *Ubiquity* 2009, August, pages.

DOI=10.1145/1595422.1595424

<http://doi.acm.org/10.1145/1595422.1595424>

Virtualization has become a hot topic. Cloud computing is the latest and most prominent application of this time-honored idea, which is almost as old as the computing field itself. The term "cloud" seems to have originated with someone's drawing of the Internet as a puffy cloud hiding many servers and connections. A user can receive a service from the cloud without ever knowing which machine (or machines) rendered the service, where it was located, or how many redundant copies of its data there are. One of the big concerns about the cloud is that it may assign many computational processes to one machine, thereby making that machine a bottleneck and giving poor response time. Faouzi Kamoun addresses this concern head on, and assures us that in most cases the virtualization used in the cloud and elsewhere improves performance. He also addresses a misconception made prominent in a Dilbert cartoon, when the boss said he wanted to virtualize the servers to save electricity.

Provides an overview of server virtualization and issues to watch out for, good and bad.

- [Beyond Server Consolidation](#)

Werner Vogels. 2008. Beyond Server Consolidation. *Queue* 6, 1 (January 2008), 20-26. DOI=10.1145/1348583.1348590

<http://doi.acm.org/10.1145/1348583.1348590>

Virtualization technology was developed in the late 1960s to make more efficient use of hardware. Hardware was expensive, and there was not that much available. Processing was largely outsourced to the few places that did have computers. On a single IBM System/360, one could run in parallel several environments that maintained full isolation and gave each of its customers the illusion of owning the hardware.¹ Virtualization was time sharing implemented at a coarse-grained level, and isolation was the key achievement of the technology. It also provided the ability to manage resources efficiently, as they would be assigned to virtual machines such that deadlines could be met and a certain quality of service could be achieved.

Explains why virtualization not only increases hardware utilization through server consolidation but also provides benefits for application development and testing.

- [SnowFlock: rapid virtual machine cloning for cloud computing](#)

Horacio Andr s Lagar-Cavilla, Joseph Andrew Whitney, Adin Matthew Scannell, Philip Patchin, Stephen M. Rumble, Eyal de Lara, Michael Brudno, and Mahadev Satyanarayanan. 2009. SnowFlock: rapid virtual machine cloning for cloud computing. In *Proceedings of the 4th ACM European conference on Computer systems (EuroSys '09)*. ACM, New York, NY, USA, 1-12. DOI=10.1145/1519065.1519067
<http://doi.acm.org/10.1145/1519065.1519067>

Virtual Machine (VM) fork is a new cloud computing abstraction that instantaneously clones a VM into multiple replicas running on different hosts. All replicas share the same initial state, matching the intuitive semantics of stateful worker creation. VM fork thus enables the straightforward creation and efficient deployment of many tasks demanding swift instantiation of stateful workers in a cloud environment, e.g. excess load handling, opportunistic job placement, or parallel computing. Lack of instantaneous stateful cloning forces users of cloud computing into ad hoc practices to manage application state and cycle provisioning. We present SnowFlock, our implementation of the VM fork abstraction. To evaluate SnowFlock, we focus on the demanding scenario of services requiring on-the-fly creation of hundreds of parallel workers in order to solve computationally-intensive queries in seconds. These services are prominent in fields such as bioinformatics, finance, and rendering. SnowFlock provides sub-second VM cloning, scales to hundreds of workers, consumes few cloud I/O resources, and has negligible runtime overhead.

Describes how to quickly create copies of a virtual machine in the cloud for efficient task replication and deployment.

- [Virtual machine contracts for datacenter and cloud computing environments](#)

Jeanna Matthews, Tal Garfinkel, Christofer Hoff, and Jeff Wheeler. 2009. Virtual machine contracts for datacenter and cloud computing environments. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds (ACDC '09)*. ACM, New York, NY, USA, 25-30. DOI=10.1145/1555271.1555278
<http://doi.acm.org/10.1145/1555271.1555278>

Virtualization is an important enabling technology for many large private datacenters and cloud computing environments. Virtual machines often have complex expectations of their runtime environment such as access to a particular network segment or storage system. Similarly, the runtime environment may have complex expectations of a virtual machine's behavior such as compliance with network access control criteria or limits on the type and quantity of network traffic generated by the virtual machine. Today, these diverse requirements are too often specified, communicated and managed with non-portable, site specific, loosely coupled, and out-of-band processes. We propose Virtual Machine Contracts (VMCs), a platform independent way of automating the communication and management of such requirements. We describe how VMCs can be expressed through additions to the Open Virtual Machine Format (OVF) standard and how they can be

managed in a uniform way even across environments with heterogeneous elements for enforcement. We explore use cases for this approach and argue that it is an essential step towards automated control and management of virtual machines in large datacenters and cloud computing environments.

Proposes and explores explicit contracts between virtual machines and their runtime environment as a way of providing more automated control over resource requirements.

Provisioning and monitoring

Cloud datacenters consist of thousands of machines and disks that must be allocated (and later reallocated) to particular applications, with machines failing regularly and demand constantly changing. How do cloud providers monitor and provision services? How is machine learning being used to automatically detect and repair anomalies in cloud services?

Readings:

- [Automated control in cloud computing: challenges and opportunities](#)

Harold C. Lim, Shivnath Babu, Jeffrey S. Chase, and Sujay S. Parekh. 2009. Automated control in cloud computing: challenges and opportunities. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds (ACDC '09)*. ACM, New York, NY, USA, 13-18.
DOI=10.1145/1555271.1555275
<http://doi.acm.org/10.1145/1555271.1555275>

With advances in virtualization technology, virtual machine services offered by cloud utility providers are becoming increasingly powerful, anchoring the ecosystem of cloud services. Virtual computing services are attractive in part because they enable customers to acquire and release computing resources for guest applications adaptively in response to load surges and other dynamic behaviors. "Elastic" cloud computing APIs present a natural opportunity for feedback controllers to automate this adaptive resource provisioning, and many recent works have explored feedback control policies for a variety of network services under various assumptions.

This paper addresses the challenge of building an effective controller as a customer add-on outside of the cloud utility service itself. Such external controllers must function within the constraints of the utility service APIs. It is important to consider techniques for effective feedback control using cloud APIs, as well as how to design those APIs to enable more effective control. As one example, we explore *proportional thresholding*, a policy enhancement for feedback controllers that enables stable control across a wide range of guest cluster sizes using the coarse-grained control offered by popular virtual compute cloud services.

Discusses the challenges of adaptive resource provisioning to meet elastic service demands and argues for placing control in the hands of cloud customers.

- [Quincy: fair scheduling for distributed computing clusters](#)

Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar, and Andrew Goldberg. 2009. Quincy: fair scheduling for distributed computing clusters. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles (SOSP '09)*. ACM, New York, NY, USA, 261-276.

DOI=10.1145/1629575.1629601

<http://doi.acm.org/10.1145/1629575.1629601>

This paper addresses the problem of scheduling concurrent jobs on clusters where application data is stored on the computing nodes. This setting, in which scheduling computations close to their data is crucial for performance, is increasingly common and arises in systems such as MapReduce, Hadoop, and Dryad as well as many grid-computing environments. We argue that data-intensive computation benefits from a fine-grain resource sharing model that differs from the coarser semi-static resource allocations implemented by most existing cluster computing architectures. The problem of scheduling with locality and fairness constraints has not previously been extensively studied under this resource-sharing model.

We introduce a powerful and flexible new framework for scheduling concurrent distributed jobs with fine-grain resource sharing. The scheduling problem is mapped to a graph datastructure, where edge weights and capacities encode the competing demands of data locality, fairness, and starvation-freedom, and a standard solver computes the optimal online schedule according to a global cost model. We evaluate our implementation of this framework, which we call Quincy, on a cluster of a few hundred computers using a varied workload of data-and CPU-intensive jobs. We evaluate Quincy against an existing queue-based algorithm and implement several policies for each scheduler, with and without fairness constraints. Quincy gets better fairness when fairness is requested, while substantially improving data locality. The volume of data transferred across the cluster is reduced by up to a factor of 3.9 in our experiments, leading to a throughput increase of up to 40%.

Describes a new approach for scheduling concurrent jobs in a computing cluster that places computations near their data while also taking fairness into account.

- [Detecting large-scale system problems by mining console logs](#)

Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael I. Jordan. 2009. Detecting large-scale system problems by mining console logs. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles (SOSP '09)*. ACM, New York, NY, USA, 117-132.

DOI=10.1145/1629575.1629587

<http://doi.acm.org/10.1145/1629575.1629587>

Surprisingly, console logs rarely help operators detect problems in large-scale datacenter services, for they often consist of the voluminous intermixing of messages from many software components written by independent developers. We propose a general methodology to mine this rich source of

information to automatically detect system runtime problems. We first parse console logs by combining source code analysis with information retrieval to create composite features. We then analyze these features using machine learning to detect operational problems. We show that our method enables analyses that are impossible with previous methods because of its superior ability to create sophisticated features. We also show how to distill the results of our analysis to an operator-friendly one-page decision tree showing the critical messages associated with the detected problems. We validate our approach using the Darkstar online game server and the Hadoop File System, where we detect numerous real problems with high accuracy and few false positives. In the Hadoop case, we are able to analyze 24 million lines of console logs in 3 minutes. Our methodology works on textual console logs of any size and requires no changes to the service software, no human input, and no knowledge of the software's internals.

Presents techniques for automatically processing textual server logs to detect system runtime problems in large datacenters.

Communications

Cloud datacenters consist of thousands of machines and disks that must be allocated (and later reallocated) to particular applications, with machines failing regularly and demand constantly changing. How do cloud providers monitor and provision services? How is machine learning being used to automatically detect and repair anomalies in cloud services?

Readings:

- [The cost of a cloud: research problems in data center networks](#)

Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. 2008. The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.* 39, 1 (December 2008), 68-73. DOI=10.1145/1496091.1496103 <http://doi.acm.org/10.1145/1496091.1496103>

The data centers used to create cloud services represent a significant investment in capital outlay and ongoing costs. Accordingly, we first examine the costs of cloud service data centers today. The cost breakdown reveals the importance of optimizing work completed per dollar invested. Unfortunately, the resources inside the data centers often operate at low utilization due to resource stranding and fragmentation. To attack this first problem, we propose (1) increasing network agility, and (2) providing appropriate incentives to shape resource consumption. Second, we note that cloud service providers are building out geo-distributed networks of data centers. Geo-diversity lowers latency to users and increases reliability in the presence of an outage taking out an entire site. However, without appropriate design and management, these geo-diverse data center networks can raise the cost of providing service. Moreover, leveraging geo-diversity requires services be designed to benefit from it. To attack this problem, we propose (1) joint optimization of network and data center resources, and (2) new systems and mechanisms for geo-distributing state.

Examines the cost of datacenters, shows that networking is a significant component of this cost, and proposes new approaches for cooperatively optimizing network and datacenter resources to improve agility.

- [PortLand: a scalable fault-tolerant layer 2 data center network fabric](#)

Radhika Niranjana Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. 2009. PortLand: a scalable fault-tolerant layer 2 data center network fabric. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication (SIGCOMM '09)*. ACM, New York, NY, USA, 39-50. DOI=10.1145/1592568.1592575
<http://doi.acm.org/10.1145/1592568.1592575>

This paper considers the requirements for a scalable, easily manageable, fault-tolerant, and efficient data center network fabric. Trends in multi-core processors, end-host virtualization, and commodities of scale are pointing to future single-site data centers with millions of virtual end points. Existing layer 2 and layer 3 network protocols face some combination of limitations in such a setting: lack of scalability, difficult management, inflexible communication, or limited support for virtual machine migration. To some extent, these limitations may be inherent for Ethernet/IP style protocols when trying to support arbitrary topologies. We observe that data center networks are often managed as a single logical network fabric with a known baseline topology and growth model. We leverage this observation in the design and implementation of PortLand, a scalable, fault tolerant layer 2 routing and forwarding protocol for data center environments. Through our implementation and evaluation, we show that PortLand holds promise for supporting a "plug-and-play" large-scale, data center network.

Introduces a new routing and forwarding protocol designed for a more scalable, fault-tolerant, and manageable datacenter network.

- [Cloud control with distributed rate limiting](#)

Barath Raghavan, Kashi Vishwanath, Sriram Ramabhadran, Kenneth Yocum, and Alex C. Snoeren. 2007. Cloud control with distributed rate limiting. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '07)*. ACM, New York, NY, USA, 337-348. DOI=10.1145/1282380.1282419
<http://doi.acm.org/10.1145/1282380.1282419>

Today's cloud-based services integrate globally distributed resources into seamless computing platforms. Provisioning and accounting for the resource usage of these Internet-scale applications presents a challenging technical problem. This paper presents the design and implementation of distributed rate limiters, which work together to enforce a global rate limit across traffic aggregates at multiple sites, enabling the coordinated policing of a cloud-based service's network traffic. Our abstraction not only enforces a global limit, but also ensures that congestion-responsive transport-layer flows behave as if they traversed a single, shared limiter. We present two designs - one general purpose, and one optimized for TCP - that allow service operators

to explicitly trade off between communication costs and system accuracy, efficiency, and scalability. Both designs are capable of rate limiting thousands of flows with negligible overhead (less than 3% in the tested configuration). We demonstrate that our TCP-centric design is scalable to hundreds of nodes while robust to both loss and communication delay, making it practical for deployment in nationwide service providers.

Describes techniques for controlling network resources within the cloud by limiting the aggregate traffic between multiple sites.

- [Enhancing dynamic cloud-based services using network virtualization](#)

Fang Hao, T. V. Lakshman, Sarit Mukherjee, and Haoyu Song. 2010. Enhancing dynamic cloud-based services using network virtualization. *SIGCOMM Comput. Commun. Rev.* 40, 1 (January 2010), 67-74. DOI=10.1145/1672308.1672322 <http://doi.acm.org/10.1145/1672308.1672322>

It is envisaged that services and applications will migrate to a cloud-computing paradigm where thin-clients on user-devices access, over the network, applications hosted in data centers by application service providers. Examples are cloud-based gaming applications and cloud-supported virtual desktops. For good performance and efficiency, it is critical that these services are delivered from locations that are the best for the current (dynamically changing) set of users. To achieve this, we expect that services will be hosted on virtual machines in interconnected data centers and that these virtual machines will migrate dynamically to locations best-suited for the current user population. A basic network infrastructure need then is the ability to migrate virtual machines across multiple networks without losing service continuity. In this paper, we develop mechanisms to accomplish this using a network-virtualization architecture that relies on a set of distributed forwarding elements with centralized control (borrowing on several recent proposals in a similar vein). We describe a preliminary prototype system, built using Openflow components, that demonstrates the feasibility of this architecture in enabling seamless migration of virtual machines and in enhancing delivery of

cloud-based services.

Presents a virtualized network architecture that permits seamless migration of virtual machines within the cloud.

Privacy and Trust

Cloud computing is viewed as risky for various reasons, especially as cloud storage systems are increasingly used to store valuable business data and intensely private data, and even mix data from different individuals on the same servers. When all of a person's (or business's) data is stored in the cloud, what steps can be taken to ensure the privacy of that data and to reassure users that their data will not be inadvertently released to others? What explicit steps can cloud providers take to overcome fears of data leakage, outages, lack of long-term service viability, and an inability to get data out of the cloud once placed there?

Readings:

- [Controlling data in the cloud: outsourcing computation without outsourcing control](#)

Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina. 2009. Controlling data in the cloud: outsourcing computation without outsourcing control. In *Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW '09)*. ACM, New York, NY, USA, 85-90. DOI=10.1145/1655008.1655020
<http://doi.acm.org/10.1145/1655008.1655020>

Cloud computing is clearly one of today's most enticing technology areas due, at least in part, to its cost-efficiency and flexibility. However, despite the surge in activity and interest, there are significant, persistent concerns about cloud computing that are impeding momentum and will eventually compromise the vision of cloud computing as a new IT procurement model. In this paper, we characterize the problems and their impact on adoption. In addition, and equally importantly, we describe how the combination of existing research thrusts has the potential to alleviate many of the concerns impeding adoption. In particular, we argue that with continued research advances in trusted computing and computation-supporting encryption, life in the cloud can be advantageous from a business intelligence standpoint over the isolated alternative that is more common today.

Examines the concerns that are preventing corporations from placing sensitive information in the cloud and suggests research directions to address these concerns.

- [Taking account of privacy when designing cloud computing services](#)

Siani Pearson. 2009. Taking account of privacy when designing cloud computing services. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09)*. IEEE Computer Society, Washington, DC, USA, 44-52. DOI=10.1109/CLOUD.2009.5071532
<http://dx.doi.org/10.1109/CLOUD.2009.5071532>

Privacy is an important issue for cloud computing, both in terms of legal compliance and user trust, and needs to be considered at every phase of design. In this paper the privacy challenges that software engineers face when targeting the cloud as their production environment to offer services are assessed, and key design principles to address these are suggested.

Argues that privacy must be considered when designing all aspects of cloud services, for both legal compliance and user acceptance, discusses the inherent challenges, and offers constructive advice.

- [Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds](#)

Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. 2009. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security (CCS '09)*. ACM, New York, NY, USA, 199-212. DOI=10.1145/1653662.1653687 <http://doi.acm.org/10.1145/1653662.1653687>

Third-party cloud computing represents the promise of outsourcing as applied to computation. Services, such as Microsoft's Azure and Amazon's EC2, allow users to instantiate virtual machines (VMs) on demand and thus purchase precisely the capacity they require when they require it. In turn, the use of virtualization allows third-party cloud providers to maximize the utilization of their sunk capital costs by multiplexing many customer VMs across a shared physical infrastructure. However, in this paper, we show that this approach can also introduce new vulnerabilities. Using the Amazon EC2 service as a case study, we show that it is possible to map the internal cloud infrastructure, identify where a particular target VM is likely to reside, and then instantiate new VMs until one is placed co-resident with the target. We explore how such placement can then be used to mount cross-VM side-channel attacks to extract information from a target VM on the same machine.

Shows how customers in a cloud can perform side-channel attacks on virtual machines to extract private information from other customers.

- [A client-based privacy manager for cloud computing](#)

Miranda Mowbray and Siani Pearson. 2009. A client-based privacy manager for cloud computing. In *Proceedings of the Fourth International ICST Conference on COMmunication System softWARE and middlewaRE (COMSWARE '09)*. ACM, New York, NY, USA, , Article 5 , 8 pages. DOI=10.1145/1621890.1621897 <http://doi.acm.org/10.1145/1621890.1621897>

A significant barrier to the adoption of cloud services is that users fear data leakage and loss of privacy if their sensitive data is processed in the cloud. In this paper, we describe a client-based privacy manager that helps reduce this risk, and that provides additional privacy-related benefits. We assess its usage within a variety of cloud computing scenarios. We have built a proof-of-concept demo that shows how privacy may be protected via reducing the amount of sensitive information sent to the cloud.

Describes a privacy manager that allows clients to control their sensitive information in cooperation with cloud service providers.

- [Trusting the cloud](#)

Christian Cachin, Idit Keidar, and Alexander Shraer. 2009. Trusting the cloud. *SIGACT News* 40, 2 (June 2009), 81-86. DOI=10.1145/1556154.1556173
<http://doi.acm.org/10.1145/1556154.1556173>

More and more users store data in "clouds" that are accessed remotely over the Internet. We survey well-known cryptographic tools for providing integrity and consistency for data stored in clouds and discuss recent research in cryptography and distributed computing addressing these problems.

Outlines cryptographic techniques for enforcing the integrity and consistency of data stored in the cloud.

Service level agreements

The service level guarantees from cloud services are imprecisely specified, often only in the minds of the users. Are best effort guarantees good enough? As cloud-based services mature, how should they provide more specific service level agreements and what sorts of guarantees will be desired by their clients?

Readings:

- [An SLA-based resource virtualization approach for on-demand service provision](#)

Attila Kertesz, Gabor Kecskemeti, and Ivona Brandic. 2009. An SLA-based resource virtualization approach for on-demand service provision. In *Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing (VTDC '09)*. ACM, New York, NY, USA, 27-34. DOI=10.1145/1555336.1555341
<http://doi.acm.org/10.1145/1555336.1555341>

Cloud computing is a newly emerged research infrastructure that builds on the latest achievements of diverse research areas, such as Grid computing, Service-oriented computing, business processes and virtualization. In this paper we present an architecture for SLA-based resource virtualization that provides an extensive solution for executing user applications in Clouds. This work represents the first attempt to combine SLA-based resource negotiations with virtualized resources in terms of on-demand service provision resulting in a holistic virtualization approach. The architecture description focuses on three topics: agreement negotiation, service brokering and deployment using virtualization. The contribution is also demonstrated with a real-world case study.

Shows how to incorporate service level agreements when provisioning virtualized resources for cloud services.

- [Automatic exploration of datacenter performance regimes](#)

Peter Bodik, Rean Griffith, Charles Sutton, Armando Fox, Michael I. Jordan, and David A. Patterson. 2009. Automatic exploration of datacenter performance regimes. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds (ACDC '09)*. ACM, New York, NY, USA, 1-6. DOI=10.1145/1555271.1555273
<http://doi.acm.org/10.1145/1555271.1555273>

Horizontally scalable Internet services present an opportunity to use automatic resource allocation strategies for system management in the datacenter. In most of the previous work, a controller employs a performance model of the system to make decisions about the optimal allocation of resources. However, these models are usually trained offline or on a small-scale deployment and will not accurately capture the performance of the controlled application. To achieve accurate control of the web application, the models need to be trained directly on the production system and adapted to changes in workload and performance of the application. In this paper we propose to train the performance model using an *exploration policy* that quickly collects data from different performance regimes of the application. The goal of our approach for managing the exploration process is to strike a balance between not violating the performance SLAs and the need to collect sufficient data to train an accurate performance model, which requires pushing the system close to its capacity. We show that by using our exploration policy, we can train a performance model of a Web 2.0 application in less than an hour and then immediately use the model in a resource allocation controller.

Presents new techniques for developing accurate performance models that can aid in configuring system services and avoid violating service level agreements.

Power management

A sizeable percentage of power consumed in the U.S. goes into datacenters. How can datacenters intelligently manage resources to save power? What can be done to reduce the energy demands of cloud-based services?

Readings:

- [Power provisioning for a warehouse-sized computer](#)

Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th annual international symposium on Computer architecture (ISCA '07)*. ACM, New York, NY, USA, 13-23. DOI=10.1145/1250662.1250665
<http://doi.acm.org/10.1145/1250662.1250665>

Large-scale Internet services require a computing infrastructure that can be appropriately described as a warehouse-sized computing system. The cost of building datacenter facilities capable of delivering a given power capacity to such a computer can rival the recurring energy consumption costs themselves. Therefore, there are strong economic incentives to operate

facilities as close as possible to maximum capacity, so that the non-recurring facility costs can be best amortized. That is difficult to achieve in practice because of uncertainties in equipment power ratings and because power consumption tends to vary significantly with the actual computing activity. Effective power provisioning strategies are needed to determine how much computing equipment can be safely and efficiently hosted within a given power budget.

In this paper we present the aggregate power usage characteristics of large collections of servers (up to 15 thousand) for different classes of applications over a period of approximately six months. Those observations allow us to evaluate opportunities for maximizing the use of the deployed power capacity of datacenters, and assess the risks of over-subscribing it. We find that even in well-tuned applications there is a noticeable gap (7 - 16%) between achieved and theoretical aggregate peak power usage at the cluster level (thousands of servers). The gap grows to almost 40% in whole datacenters. This headroom can be used to deploy additional compute equipment within the same power budget with minimal risk of exceeding it. We use our modeling framework to estimate the potential of power management schemes to reduce peak power and energy usage. We find that the opportunities for power and energy savings are significant, but greater at the cluster-level (thousands of servers) than at the rack-level (tens). Finally we argue that systems need to be power efficient across the activity range, and not only at peak performance levels.

Presents results from a study of the power consumption of large clusters of servers and suggests opportunities for significant energy savings.

- [Cutting the electric bill for internet-scale systems](#)

Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. 2009. Cutting the electric bill for internet-scale systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication (SIGCOMM '09)*. ACM, New York, NY, USA, 123-134.
DOI=10.1145/1592568.1592584
<http://doi.acm.org/10.1145/1592568.1592584>

Energy expenses are becoming an increasingly important fraction of data center operating costs. At the same time, the energy expense per unit of computation can vary significantly between two different locations. In this paper, we characterize the variation due to fluctuating electricity prices and argue that existing distributed systems should be able to exploit this variation for significant economic gains. Electricity prices exhibit both temporal and geographic variation, due to regional demand differences, transmission inefficiencies, and generation diversity. Starting with historical electricity prices, for twenty nine locations in the US, and network traffic data collected on Akamai's CDN, we use simulation to quantify the possible economic gains for a realistic workload. Our results imply that existing systems may be able to save millions of dollars a year in electricity costs, by being cognizant of locational computation cost differences.

Observes that electricity prices vary temporally and geographically, and presents a technique to reduce energy costs by exploiting this property.

- [GreenCloud: a new architecture for green data center](#)

Liang Liu, Hao Wang, Xue Liu, Xing Jin, Wen Bo He, Qing Bo Wang, and Ying Chen. 2009. GreenCloud: a new architecture for green data center. In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session (ICAC-INDST '09)*. ACM, New York, NY, USA, 29-38. DOI=10.1145/1555312.1555319
<http://doi.acm.org/10.1145/1555312.1555319>

Nowadays, power consumption of data centers has huge impacts on environments. Researchers are seeking to find effective solutions to make data centers reduce power consumption while keep the desired quality of service or service level objectives. Virtual Machine (VM) technology has been widely applied in data center environments due to its seminal features, including reliability, flexibility, and the ease of management. We present the *GreenCloud* architecture, which aims to reduce data center power consumption, while guarantee the performance from users' perspective. *GreenCloud* architecture enables comprehensive online-monitoring, live virtual machine migration, and VM placement optimization. To verify the efficiency and effectiveness of the proposed architecture, we take an online real-time game, Tremulous, as a VM application. Evaluation results show that we can save up to 27% of the energy when applying GreenCloud architecture.

Describes an architecture that reduces energy consumption in a datacenter through on-line monitoring and migration of virtual machines.

Mobile clients

Increasingly, the clients of cloud-based services are not desktop PCs but rather mobile devices, such as cell phones and portable media players. How do mobile devices at the edge of the network interact with cloud-based services to effectively manage data and computation on behalf of users? How does a user's location factor into the design of cloud-based services?

Readings:

- [Elastic mobility: stretching interaction](#)

Lucia Terrenghi, Thomas Lang, and Bernhard Lehner. 2009. Elastic mobility: stretching interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, , Article 46 , 4 pages. DOI=10.1145/1613858.1613916
<http://doi.acm.org/10.1145/1613858.1613916>

Based on a consideration of usage and technological computing trends, we reflect on the implications of cloud computing on mobile interaction with applications, data and devices. We argue that by extending the interaction capabilities of the mobile device by connecting it to external peripherals, new mobile contexts of personal (and social) computing can emerge, thus creating

novel contexts of mobile interaction. In such a scenario, mobile devices can act as context-adaptive information filters. We then present Focus, our work in progress on a context-adaptive UI, which we can demonstrate at the MobileHCI demo session as a clickable dummy on a mobile device.

Reflects on how cloud computing will augment applications on mobile devices, and vice versa, particularly for context-aware interaction.

- [Using RESTful web-services and cloud computing to create next generation mobile applications](#)

Jason H. Christensen. 2009. Using RESTful web-services and cloud computing to create next generation mobile applications. In *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications (OOPSLA '09)*. ACM, New York, NY, USA, 627-634. DOI=10.1145/1639950.1639958
<http://doi.acm.org/10.1145/1639950.1639958>

In this paper we will examine the architectural considerations of creating next generation mobile applications using Cloud Computing and RESTful Web Services. With the advent of multimodal smart mobile devices like the iPhone, connected applications can be created that far exceed traditional mobile device capabilities. Combining the context that can be ascertained from the sensors on the smart mobile device with the ability to offload processing capabilities, storage, and security to cloud computing over any one of the available network modes via RESTful web-services, has allowed us to enter a powerful new era of mobile consumer computing. To best leverage this we need to consider the capabilities and constraints of these architectures. Some of these are traditional trade-offs from distributed computing such as a web-services request frequency vs. payload size. Others are completely new - for instance, determining which network type we are on for bandwidth considerations, federated identity limitations on mobile platforms, and application approval.

Explores architectures for mobile applications that access cloud-based services.

The ACM Digital Library is published by the Association for Computing Machinery.

Copyright © 2010 ACM, Inc.