ACM Tech Pack on Cloud Computing

Ajay Mohindra, IBM Research Division, Thomas J. Watson Research Center
Chair, ACM Tech Pack Committee on Cloud Computing

INTRODUCTION

(Updated November 2014)

Cloud computing promises to radically change the way that 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 can be read to learn more about 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.

Update: Please note that starting with the latest version of the Tech Pack, the editor has categorized the resources as follows:

[100] – Introductory Materials/Foundation
[200] – Intermediate (including practice/implementation)
[300] – Academic/Looking Ahead

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

A View of Cloud Computing [100]
Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (April 2010), 50-58. DOI=10.1145/1721654.1721672 http://doi.acm.org/10.1145/1721654.1721672 **Abstract:** Clearing the clouds away from the true potential and obstacles posed by this computing capability.
Significance: Presents the economic benefits of cloud computing as well as the top ten obstacles and opportunities.

Cloud computing [100]
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 **Abstract:** As software migrates from local PCs to distant Internet servers, users and developers alike go along for the ride.
Significance: Discusses the trend of moving software applications into the cloud and the challenges.

Cloud Computing: An Overview [100]
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 **Abstract:** A summary of important cloud-computing issues distilled from ACM CTO Roundtables.
Significance: Presents some of the key topics discussed during the ACM Cloud Computing and Virtualization CTO Roundtables of 2008.

Computing in the clouds [100]
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 **Abstract:** Powerful services and applications are being integrated and packaged on the Web in what the industry now calls "cloud computing."
Significance: Explores the many perspectives on cloud computing and debunks the notion that it is simply a rebranding of old computing models.

CTO Roundtable: Cloud Computing [100]
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 **Abstract:** Our panel of experts discuss cloud computing and how companies can make the best use of it.
Significance: Provides solid advice from a panel of experts on how organizations can benefit from cloud computing.

Emergence of the Academic Computing Clouds [100]
A. Delic and Martin Anthony Walker. 2008. Emergence of the Academic Computing Clouds. *Ubiquity* 2008, August, Article 1 (August 2008), 1 page. DOI=10.1145/1414663.1414664 http://doi.acm.org/10.1145/1414663.1414664 **Abstract:** 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.
Significance: Discusses the emergence of cloud computing in support of experimental sciences addressing engineering, medical, and social problems.

Security in the Cloud [100]
Gary Anthes. 2010. Security in the Cloud. *Commun. ACM* 53, 11 (November 2010), 16-18. DOI=10.1145/1839676.1839683 http://doi.acm.org/10.1145/1839676.1839683 **Abstract:** Cloud computing offers many advantages, but also involves security risks. Fortunately, researchers are devising some ingenious solutions.
Significance: Discusses recent research that could lead to better security in the cloud, including agents that monitor virtual machines and techniques for computation on encrypted data.

NIST Cloud Computing Reference Architecture [200]
Fang Liu; Jin Tong; Jian Mao, Robert B. Bohn, John V. Messina, Mark L. Badger, Dawn M. Leaf. 2011. NIST Cloud Computing Reference Architecture. *NIST Special Publication* 500-292. National Institute of Standards and Technology Special Publication 500-292
Abstract: The adoption of cloud computing into the Federal Government and its implementation depend upon a variety of technical and non-technical factors. A fundamental reference point, based on the NIST definition of Cloud Computing, is needed to describe an overall framework that can be used government-wide. This document presents the NIST Cloud Computing Reference Architecture (RA) and Taxonomy (Tax) that will accurately communicate the components and offerings of cloud computing.
Significance: Provides an overall framework for components and offerings in a cloud computing ecosystem.

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

Building a Database on S3 [100]
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 **Abstract:** 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.
Significance: 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.

Benchmarking Cloud Serving Systems with YCSB [200]
Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. 2010. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10). ACM, New York, NY, USA, 143-154. DOI=10.1145/1807128.1807152 http://doi.acm.org/10.1145/1807128.1807152 **Abstract:** While the use of MapReduce systems (such as Hadoop) for large scale data analysis has been widely recognized and studied, we have recently seen an explosion in the number of systems developed for cloud data serving. These newer systems address "cloud OLTP" applications, though they typically do not support ACID transactions. Examples of systems proposed for cloud serving use include BigTable, PNUTS, Cassandra, HBase, Azure, CouchDB, SimpleDB, Voldemort, and many others. Further, they are being applied to a diverse range of applications that differ considerably from traditional (e.g., TPC-C like) serving workloads. The number of emerging cloud serving systems and the wide range of proposed applications, coupled with a lack of apples-to-apples performance comparisons, makes it difficult to understand the tradeoffs between systems and the workloads for which they are suited. We present the "Yahoo! Cloud Serving Benchmark" (YCSB) framework, with the goal of facilitating performance comparisons of the new generation of cloud data serving systems. We define a core set of benchmarks and report results for four widely used systems: Cassandra, HBase, Yahoo!'s PNUTS, and a simple sharded MySQL implementation. We also hope to foster the development of additional cloud benchmark suites that represent other classes of applications by making our benchmark tool available via open source. In this regard, a key feature of the YCSB framework/tool is that it is extensible—it supports easy definition of new workloads, in addition to making it easy to benchmark new systems.
Significance: Proposes the first benchmark for evaluating the performance and scalability of a variety of cloud storage services.

Cumulus: an open source storage cloud for science [200]
John Bresnahan, Kate Keahey, David LaBissoniere, and Tim Freeman. 2011. Cumulus: an open source storage cloud for science. In *Proceedings of the 2nd international workshop on Scientific cloud computing* (ScienceCloud '11). ACM, New York, NY, USA, 25-32.

DOI=10.1145/1996109.1996115 http://doi.acm.org/10.1145/1996109.1996115 **Abstract:** Amazon's S3 protocol has emerged as the de facto interface for storage in the commercial data cloud. However, it is closed source and unavailable to the numerous science data centers all over the country. Just as Amazon's Simple Storage Service (S3) provides reliable data cloud access to commercial users, scientific data centers must provide their users with a similar level of service. Ideally scientific data centers could allow the use of the same clients and protocols that have proven effective to Amazon's users. But how well does the S3 REST interface compare with the data cloud transfer services used in today's computational centers? Does it have the features needed to support the scientific community? If not, can it be extended to include these features without loss of compatibility? Can it scale and distribute resources equally when presented with common scientific the usage patterns? We address these questions by presenting Cumulus, an open source implementation of the Amazon S3 REST API. It is packaged with the Nimbus IaaS toolkit and provides scalable and reliable access to scientific data. Its performance compares favorably with that of GridFTP and SCP, and we have added features necessary to support the econometrics important to the scientific community.
Significance: 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.

Cumulus: Filesystem Backup to the Cloud [200]
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 **Abstract:** 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.
Significance: 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.

Organizing and Sharing Distributed Personal Web-Service Data [200]
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
Abstract: 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.
Significance: 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.

RACS: A Case for Cloud Storage Diversity [200]
Hussam Abu-Libdeh, Lonnie Princehouse, and Hakim Weatherspoon. 2010. RACS: a case for cloud storage diversity. In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10). ACM, New York, NY, USA, 229-240. DOI=10.1145/1807128.1807165 http://doi.acm.org/10.1145/1807128.1807165 **Abstract:** The increasing popularity of cloud storage is leading organizations to consider moving data out of their own data centers and into the cloud. However, success for cloud storage providers can present a significant risk to customers; namely, it becomes very expensive to switch storage providers. In this paper, we make a case for applying RAID-like techniques used by disks and file systems, but at the cloud storage level. We argue that striping user data across multiple providers can allow customers to avoid vendor lock-in, reduce the cost of switching providers, and better tolerate provider outages or failures. We introduce RACS, a proxy that transparently spreads the storage load over many providers. We evaluate a prototype of our system and estimate the costs incurred and benefits reaped. Finally, we use trace-driven simulations to demonstrate how RACS can reduce the cost of switching storage vendors for a large organization such as the Internet Archive by seven-fold or more by varying erasure-coding parameters.
Significance: Presents techniques for avoiding lock-in and increasing data survivability by spreading data across cloud storage providers.

Research on cloud storage architecture and key technologies [200]
Wenying Zeng, Yuelong Zhao, Kairi Ou, and Wei Song. 2009. Research on cloud storage architecture and key technologies. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human* (ICIS '09). ACM, New York, NY, USA, 1044-1048. DOI=10.1145/1655925.1656114 http://doi.acm.org/10.1145/1655925.1656114 **Abstract:** This paper proposes a general architecture of cloud storage system, analyzes the functions of the components, and discusses the key technologies, etc. Cloud storage is a novel storage service mode which the service providers supply storage capacities and data storage services through the Internet to the clients; meanwhile, the clients needn't know the details and lowered structures and mechanisms. The proposed architecture of cloud storage is layered and cooperative, and the discussed key technologies involve deployment, storage virtualization, data organization, migration, security, etc. The operation mechanism including ecology chain, game theory, ant colony optimization, data life cycle management, maintenance and update, convergence and evolution mechanisms are analyzed too. So an overall and new viewpoint to cloud storage system is illustrated.
Significance: 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.

Resilient cloud data storage services [200]
Hemayamini Kurra, Youssif Al-Nashif, and Salim Hariri. 2013. Resilient cloud data storage services. In *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference* (CAC '13). ACM, New York, NY, USA. Article 17, 9 pages. DOI=10.1145/2494621.2494634 http://doi.acm.org/10.1145/2494621.2494634 **Abstract:** With the advance of cloud computing technologies, there is a huge demand for computing resources and storage. Many organizations prefer to outsource their storage and other resources. As the data reside on the third parties data centers, security is becoming a major concern. In this paper we propose a Resilient Cloud Storage (RCS) architecture that addresses the major security issues for cloud storage such as access control confidentiality, integrity, and secure communications. Our resilient approach is based on moving target defense and key hopping techniques. Data is partitioned into a random number of partitions where different keys are used to encrypt each partition. We also show that by using key hopping technique, we can reduce smaller key length that is normally used to improve performance without compromising the security. Our experimental results show that we can improve performance by 50% when we use a key of length 512 when compared with certificate technique that uses key length of 2048.
Significance: In this paper, the authors propose a Resilient Cloud Storage (RCS) architecture that addresses the major security issues for cloud storage such as access control confidentiality, integrity, and secure communications.

The Google file system [200]
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 **Abstract:** We have designed and implemented the Google File System, a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients. While sharing many of the same goals as previous distributed file systems, our design has been driven by observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system assumptions. This has led us to reexamine traditional choices and explore radically different design points. The file system has successfully met our storage needs. It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets. The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients. In this paper, we present file system interface extensions designed to support distributed applications, discuss many aspects of our design, and report measurements from both micro-benchmarks and real world use.
Significance: 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.

Windows Azure Storage: a highly available cloud storage service with strong consistency [200]
Brad Calder, Ju Wang, Aaron Ogus, Niranjan Nilakantan, Arild Skjolsvold, Sam McKelvie,

Yikang Xu, Shashwat Srivastav, Jiesheng Wu, Huseyin Simitci, Jaidev Haridas, Chakravarthy Uddaraju, Hemal Khatri, Andrew Edwards, Vaman Bedekar, Shane Mainali, Rafay Abbasi, Arpit Agarwal, Mian Fahim ul Haq, Muhammad Ikram ul Haq, Deepali Bhardwaj, Sowmya Dayanand, Anitha Adusumilli, Marvin McNett, Sriram Sankaran, Kavitha Manivannan, and Leonidas Rigas. 2011. Windows Azure Storage: a highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (SOSP '11). ACM, New York, NY, USA, 143-157. DOI=10.1145/2043556.2043571 http://doi.acm.org/10.1145/2043556.2043571 **Abstract:** Windows Azure Storage (WAS) is a cloud storage system that provides customers the ability to store seemingly limitless amounts of data for any duration of time. WAS customers have access to their data from anywhere at any time and only pay for what they use and store. In WAS, data is stored durably using both local and geographic replication to facilitate disaster recovery. Currently, WAS storage comes in the form of Blobs (files), Tables (structured storage), and Queues (message delivery). In this paper, we describe the WAS architecture, global namespace, and data model, as well as its resource provisioning, load balancing, and replication systems.
Significance: Provides details on the architecture of a commercial cloud storage service.

Consistency-based service level agreements for cloud storage [300]
Douglas B. Terry, Vijayan Prabhakaran, Ramakrishna Kotla, Mahesh Balakrishnan, Marcos K. Aguilera, and Hussam Abu-Libdeh. 2013. Consistency-based service level agreements for cloud storage. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles* (SOSP '13). ACM, New York, NY, USA, 309-324. DOI=10.1145/2517349.2522731 http://doi.acm.org/10.1145/2517349.2522731 **Abstract:** Choosing a cloud storage system and specific operations for reading and writing data requires developers to make decisions that trade off consistency for availability and performance. Applications may be locked into a choice that is not ideal for all clients and changing conditions. Pileus is a replicated key-value store that allows applications to declare their consistency and latency priorities via consistency-based service level agreements (SLAs). It dynamically selects which servers to access in order to deliver the best service given the current configuration and system conditions. In application-specific SLAs, developers can request both strong and eventual consistency as well as intermediate guarantees such as read-my-writes. Evaluations running on a worldwide test bed with geo-replicated data show that the system adapts to varying client-server latencies to provide service that matches or exceeds the best static consistency choice and server selection scheme.
Significance: Examines the tradeoffs of providing different application level consistency policies to developers.

DepSky: Dependable and Secure Storage in a Cloud-of-Clouds [300]
Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando André, and Paulo Sousa. 2013. DepSky: Dependable and Secure Storage in a Cloud-of-Clouds. *Trans. Storage* 9, 4, Article 12 (November 2013), 33 pages. DOI=10.1145/2535929 http://doi.acm.org/10.1145/2535929
Abstract: The increasing popularity of cloud storage services has lead companies that handle critical data to think about using these services for their storage needs. Medical record databases, large biomedical datasets, historical information about power systems and financial data are some examples of critical data that could be moved to the cloud. However, the reliability and security of data stored in the cloud still remain major concerns. In this work we present DepSky, a system that improves the availability, integrity, and confidentiality of information stored in the

cloud through the encryption, encoding, and replication of the data on diverse clouds that form a cloud-of-clouds. We deployed our system using four commercial clouds and used PlanetLab to run clients accessing the service from different countries. We observed that our protocols improved the perceived availability, and in most cases, the access latency, when compared with cloud providers individually. Moreover, the monetary costs of using DepSky in this scenario is at most twice the cost of using a single cloud, which is optimal and seems to be a reasonable cost, given the benefits.
Significance: Discusses the use of a cloud federation to handle sensitive workloads.

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 [100]
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 **Abstract:** Building reliable distributed systems at a worldwide scale demands trade-offs between consistency and availability.
Significance: 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.

Replicated data consistency explained through baseball [100]
Doug Terry. 2013. Replicated data consistency explained through baseball. *Commun. ACM* 56, 12 (December 2013), 82-89. DOI=10.1145/2500500 http://doi.acm.org/10.1145/2500500
Abstract: A broader class of consistency guarantees can, and perhaps should, be offered to clients that read shared data.
Significance: Discusses offering a selection of consistency guarantees to application developers.

Survey on NoSQL database [100]
Jing Han; PCN&CAD Center, *Beijing Univ. of Posts & Telecommun.*, Beijing, China; Haihong, E.; Guan Le; Jian Du, Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on Pervasive Computing and Applications. DOI: 10.1109/ICPCA.2011.6106531 http://dx.doi.org/10.1109/ICPCA.2011.6106531 **Abstract:** This paper describes the background, basic characteristics, data model of NoSQL. In addition, this paper classifies NoSQL databases according to the CAP theorem. Finally, the mainstream NoSQL databases are separately described in detail, and extract some properties to help enterprises to choose NoSQL.
Significance: Describes the background, basic characteristics, and data model of NoSQL.

Dynamo: Amazon's Highly Available Key-Value Store [200]
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
Abstract: 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.
Significance: 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.

Eventual consistency: How soon is eventual? An evaluation of Amazon S3's consistency behavior [200]
David Bermbach and Stefan Tai. 2011. Eventual consistency: How soon is eventual? An evaluation of Amazon S3's consistency behavior. In *Proceedings of the 6th Workshop on Middleware for Service Oriented Computing* (MW4SOC '11). ACM, New York, NY, USA, Article 1, 6 pages. DOI=10.1145/2093185.2093186
http://doi.acm.org/10.1145/2093185.2093186 **Abstract:** In this work the authors present a novel approach to benchmark staleness in distributed data stores and use the approach to evaluate Amazon's Simple Storage Service (S3).
Significance: In this work the authors present a novel approach to benchmark staleness in distributed data stores and use the approach to evaluate Amazon's Simple Storage Service (S3).

A Self-Organized, Fault-Tolerant and Scalable Replication Scheme for Cloud Storage [300]
Nicolas Bonvin, Thanasis G. Papaioannou, and Karl Aberer. 2010. A self-organized, fault-tolerant and scalable replication scheme for cloud storage. In *Proceedings of the 1st ACM symposium on Cloud computing*. ACM, New York, NY, USA, 205-216. DOI=10.1145/1807128.1807162 http://doi.acm.org/10.1145/1807128.1807162 **Abstract:** Failures of any type are common in current datacenters, partly due to the higher scales of the data stored. As data scales up, its availability becomes more complex, while different availability levels per application or per data item may be required. In this paper, we propose a self-managed key-value store that dynamically allocates the resources of a data cloud to several applications in a cost-efficient and fair way. Our approach offers and dynamically maintains multiple differentiated availability guarantees to each different application despite failures. We employ a virtual economy, where each data partition (i.e. a key range in a consistent-hashing space) acts as an individual optimizer and chooses whether to migrate, replicate or remove itself based on net benefit maximization regarding the utility offered by the partition and its storage and

maintenance cost. As proved by a game-theoretical model, no migrations or replications occur in the system at equilibrium, which is soon reached when the query load and the used storage are stable. Moreover, by means of extensive simulation experiments, we have proved that our approach dynamically finds the optimal resource allocation that balances the query processing overhead and satisfies the availability objectives in a cost-efficient way for different query rates and storage requirements. Finally, we have implemented a fully working prototype of our approach that clearly demonstrates its applicability in real settings.
Significance: Applies a distributed economic model for self-organization of data replicas in a cloud storage system that adapts to adding new storage, to node failures, and to dynamic clients.

Don't settle for eventual: scalable causal consistency for wide-area storage with COPS [300]
Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, and David G. Andersen. 2011. Don't settle for eventual: scalable causal consistency for wide-area storage with COPS. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (SOSP '11). ACM, New York, NY, USA, 401-416. DOI=10.1145/2043556.2043593 http://doi.acm.org/10.1145/2043556.2043593 **Abstract:** With the development of the Internet and cloud computing, there need databases to be able to store and process big data effectively, demand for high-performance when reading and writing, so the traditional relational database is facing many new challenges. Especially in large scale and high-concurrency applications, such as search engines and SNS, using the relational database to store and query dynamic user data has appeared to be inadequate. In this case, NoSQL database created. This paper describes the background, basic characteristics, data model of NoSQL. In addition, this paper classifies NoSQL databases according to the CAP theorem. Finally, the mainstream NoSQL databases are separately described in detail, and extract some properties to help enterprises to choose NoSQL.
Significance: Defines a consistency model based on causal consistency with convergent conflict handling.

How Replicated Data Management in the Cloud Can Benefit from a Data Grid Protocol: The Re:GRIDiT Approach [300]
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
Abstract: 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.
Significance: Compares cloud data management with previous work on data grids.

Middleware-based Database Replication: the Gaps between Theory and Practice [300]
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
Abstract: 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.
Significance: 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 and Parallel DBMSs: Friends or Foes? [100]
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 **Abstract:** MapReduce complements DBMSs since databases are not designed for extract-transform-load tasks, a MapReduce specialty.
Significance: 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.

MapReduce: Simplified Data Processing on Large Clusters [100]
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 **Abstract:** 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.
Significance: 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.

Parallel data processing with MapReduce: a survey [100]
Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. 2012. Parallel data processing with MapReduce: a survey. *SIGMOD Rec.* 40, 4 (January 2012), 11-20. DOI=10.1145/2094114.2094118 http://doi.acm.org/10.1145/2094114.2094118 **Abstract:** A prominent parallel data processing tool MapReduce is gaining significant momentum from both industry and academia as the volume of data to analyze grows rapidly. While MapReduce is used in many areas where massive data analysis is required, there are still debates on its performance, efficiency per node, and simple abstraction. This survey intends to assist the database and open source communities in understanding various technical aspects of the MapReduce framework. In this survey, we characterize the MapReduce framework and discuss its inherent pros and cons. We then introduce its optimization strategies reported in the recent literature. We also discuss the open issues and challenges raised on parallel data analysis with MapReduce.
Significance: Presents the pros & cons of MapReduce framework, & discuss the open issues and challenges raised on parallel data analysis with MapReduce.

Boom Analytics: Exploring Data-Centric, Declarative Programming for the Cloud [300]
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 **Abstract:** 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.
Significance: Explores a declarative approach to writing data-parallel programs that run in a cloud environment.

Distributed Data-Parallel Computing Using a High-Level Programming Language [300]
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 **Abstract:** 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.
Significance: Offers another programming model for large-scale data-parallel computing based on Microsoft's LINQ platform for SQL-like queries.

Distributed Shared Memory Programming in the Cloud [300]
Ahmad Anbar, Vikram K. Narayana, and Tarek El-Ghazawi. 2012. Distributed Shared Memory Programming in the Cloud. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (CCGRID '12). IEEE Computer Society, Washington, DC, USA, 707-708. DOI=10.1109/CCGrid.2012.48 http://dx.doi.org/10.1109/CCGrid.2012.48 **Abstract:** The paper explores the Adoption of a distributed shared memory (DSM) programming paradigm through the use of Partitioned Global Address Space (PGAS) languages.
Significance: Explores the adoption of a distributed shared memory (DSM) programming paradigm for writing applications for Cloud.

Stateful bulk processing for incremental analytics [300]
Dionysios Logothetis, Christopher Olston, Benjamin Reed, Kevin C. Webb, and Ken Yocum. 2010. Stateful bulk processing for incremental analytics. In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10). ACM, New York, NY, USA, 51-62. DOI=10.1145/1807128.1807138 http://doi.acm.org/10.1145/1807128.1807138 **Abstract:** This work addresses the need for stateful dataflow programs that can rapidly sift through huge, evolving data sets. These data-intensive applications perform complex multi-step computations over successive generations of data inflows, such as weekly web crawls, daily image/video uploads, log files, and growing social networks. While programmers may simply re-run the entire dataflow when new data arrives, this is grossly inefficient, increasing result latency and squandering hardware resources and energy. Alternatively, programmers may use prior results to incrementally incorporate the changes. However, current large-scale data processing tools, such as Map-Reduce or Dryad, limit how programmers incorporate and use state in data-parallel programs. Straightforward approaches to incorporating state can result in custom, fragile code and disappointing performance. This work presents a generalized architecture for continuous bulk processing (CBP) that raises the level of abstraction for building incremental applications. At its core is a flexible, groupwise processing operator that takes state as an explicit input. Unifying stateful programming with a data-parallel operator affords several fundamental opportunities for minimizing the movement of data in the underlying processing system. As case studies, we show how one can use a small set of flexible dataflow primitives to perform web analytics and mine large-scale, evolving graphs in an incremental fashion. Experiments with our prototype using real-world data indicate significant data movement and running time reductions relative to current practice. For example, incrementally computing PageRank using CBP can reduce data movement by 46% and cut running time in half.
Significance: Discusses the design and applications of a system for incremental analysis of large data that changes daily.

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

Beyond Server Consolidation [100]
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 **Abstract:** 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.
Significance: Explains why virtualization not only increases hardware utilization through server consolidation but also provides benefits for application development and testing.

Bringing Virtualization to the x86 Architecture with the Original VMware Workstation [100]
Edouard Bugnion, Scott Devine, Mendel Rosenblum, Jeremy Sugerman, and Edward Y. Wang.

2012. Bringing Virtualization to the x86 Architecture with the Original VMware Workstation. *ACM Trans. Comput. Syst.* 30, 4, Article 12 (November 2012), 51 pages. DOI=10.1145/2382553.2382554 http://doi.acm.org/10.1145/2382553.2382554 **Abstract:** This article describes the historical context, technical challenges, and main implementation techniques used by VMware Workstation to bring virtualization to the x86 architecture in 1999. Although virtual machine monitors (VMMs) had been around for decades, they were traditionally designed as part of monolithic, single-vendor architectures with explicit support for virtualization. In contrast, the x86 architecture lacked virtualization support, and the industry around it had disaggregated into an ecosystem, with different vendors controlling the computers, CPUs, peripherals, operating systems, and applications, none of them asking for virtualization. We chose to build our solution independently of these vendors. As a result, VMware Workstation had to deal with new challenges associated with (i) the lack of virtualization support in the x86 architecture, (ii) the daunting complexity of the architecture itself, (iii) the need to support a broad combination of peripherals, and (iv) the need to offer a simple user experience within existing environments. These new challenges led us to a novel combination of well-known virtualization techniques, techniques from other domains, and new techniques. VMware Workstation combined a hosted architecture with a VMM. The hosted architecture enabled a simple user experience and offered broad hardware compatibility. Rather than exposing I/O diversity to the virtual machines, VMware Workstation also relied on software emulation of I/O devices. The VMM combined a trap-and-emulate direct execution engine with a system-level dynamic binary translator to efficiently virtualize the x86 architecture and support most commodity operating systems. By relying on x86 hardware segmentation as a protection mechanism, the binary translator could execute translated code at near hardware speeds. The binary translator also relied on partial evaluation and adaptive retranslation to reduce the overall overheads of virtualization. Written with the benefit of hindsight, this article shares the key lessons we learned from building the original system and from its later evolution.
Significance: Describes the historical context, technical challenges, and main implementation techniques used by VMware Workstation to bring virtualization to the x86 architecture in 1999.

Virtualization: blessing or curse? [100]
Evangelos Kotsovinos. 2011. Virtualization: blessing or curse? *Commun. ACM* 54, 1 (January 2011), 61-65. DOI=10.1145/1866739.1866754 http://doi.acm.org/10.1145/1866739.1866754
Abstract: Managing virtualization at a large scale is fraught with hidden challenges.
Significance: Cautions that virtualization, although beneficial, does not automatically solve all of the problems of managing large datacenters.

Virtualization: Issues, security threats, and solutions [100]
Michael Pearce, Sherali Zeadally, and Ray Hunt. 2013. Virtualization: Issues, security threats, and solutions. *ACM Comput. Surv.* 45, 2, Article 17 (March 2013), 39 pages. DOI=10.1145/2431211.2431216 http://doi.acm.org/10.1145/2431211.2431216 **Abstract:** Although system virtualization is not a new paradigm, the way in which it is used in modern system architectures provides a powerful platform for system building, the advantages of which have only been realized in recent years, as a result of the rapid deployment of commodity hardware and software systems. In principle, virtualization involves the use of an encapsulating software layer (Hypervisor or Virtual Machine Monitor) which surrounds or underlies an operating system and provides the same inputs, outputs, and behavior that would be expected

from an actual physical device. This abstraction means that an ideal Virtual Machine Monitor provides an environment to the software equivalent to the host system, but which is decoupled from the hardware state. Because a virtual machine is not dependent on the state of the physical hardware, multiple virtual machines may be installed on a single set of hardware. The decoupling of physical and logical states gives virtualization inherent security benefits. However, the design, implementation, and deployment of virtualization technology have also opened up novel threats and security issues which, while not particular to system virtualization, take on new forms in relation to it. Reverse engineering becomes easier due to introspection capabilities, as encryption keys, security algorithms, low-level protection, intrusion detection, or anti-debugging measures can become more easily compromised. Furthermore, associated technologies such as virtual routing and networking can create challenging issues for security, intrusion control, and associated forensic processes. We explain the security considerations and some associated methodologies by which security breaches can occur, and offer recommendations for how virtualized environments can best be protected. Finally, we offer a set of generalized recommendations that can be applied to achieve secure virtualized implementations.
Significance: Presents an overview of security methodologies by which security breaches can occur, and offer recommendations for how virtualized environments can best be protected.

CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines [200]
Timothy Wood, K. K. Ramakrishnan, Prashant Shenoy, and Jacobus van der Merwe. 2011. CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines. In *Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments* (VEE '11). ACM, New York, NY, USA, 121-132. DOI=10.1145/1952682.1952699 http://doi.acm.org/10.1145/1952682.1952699 **Abstract:** Virtual machine technology and the ease with which VMs can be migrated within the LAN, has changed the scope of resource management from allocating resources on a single server to manipulating pools of resources within a data center. We expect WAN migration of virtual machines to likewise transform the scope of provisioning compute resources from a single data center to multiple data centers spread across the country or around the world. In this paper we present the CloudNet architecture as a cloud framework consisting of cloud computing platforms linked with a VPN based network infrastructure to provide seamless and secure connectivity between enterprise and cloud data center sites. To realize our vision of efficiently pooling geographically distributed data center resources, CloudNet provides optimized support for live WAN migration of virtual machines. Specifically, we present a set of optimizations that minimize the cost of transferring storage and virtual machine memory during migrations over low bandwidth and high latency Internet links. We evaluate our system on an operational cloud platform distributed across the continental US. During simultaneous migrations of four VMs between data centers in Texas and Illinois, CloudNet's optimizations reduce memory migration time by 65% and lower bandwidth consumption for the storage and memory transfer by 19GB, a 50% reduction.
Significance: Argues that migrating virtual machines across datacenters is desirable for cloud bursting, site consolidation, and low-latency access to global services, and proposes techniques for efficient VM migration.

Virtualizing the Datacenter Without Compromising Server Performance [200]
Faouzi Kamoun. 2009. Virtualizing the Datacenter Without Compromising Server Performance.

Ubiquity 2009, August. DOI=10.1145/1595422.1595424 http://doi.acm.org/10.1145/1595422.1595424 **Abstract:** 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.
Significance: Provides an overview of server virtualization and issues to watch out for, good and bad.

NOVA: a microhypervisor-based secure virtualization architecture [300]
Udo Steinberg and Bernhard Kauer. 2010. NOVA: a microhypervisor-based secure virtualization architecture. In *Proceedings of the 5th European conference on Computer system*s (EuroSys '10). ACM, New York, NY, USA, 209-222. DOI=10.1145/1755913.1755935 http://doi.acm.org/10.1145/1755913.1755935 **Abstract:** The availability of virtualization features in modern CPUs has reinforced the trend of consolidating multiple guest operating systems on top of a hypervisor in order to improve platform-resource utilization and reduce the total cost of ownership. However, today's virtualization stacks are unduly large and therefore prone to attacks. If an adversary manages to compromise the hypervisor, subverting the security of all hosted operating systems is easy. We show how a thin and simple virtualization layer reduces the attack surface significantly and thereby increases the overall security of the system. We have designed and implemented a virtualization architecture that can host multiple unmodified guest operating systems. Its trusted computing base is at least an order of magnitude smaller than that of existing systems. Furthermore, on recent hardware, our implementation outperforms contemporary full virtualization environments.
Significance: Describe techniques to increase security of cloud systems through the virtualization layer.

The Turtles Project: Design and Implementation of Nested Virtualization [300]
Muli Ben-Yehuda, Michael D. Day, Zvi Dubitzky, Michael Factor, Nadav Har'El, Abel Gordon, Anthony Liguori, Orit Wasserman, and Ben-Ami Yassour. 2010. The turtles project: design and implementation of nested virtualization. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation* (OSDI'10). USENIX Association, Berkeley, CA, USA, 1-6. **Abstract:** In classical machine virtualization, a hypervisor runs multiple operating systems simultaneously, each on its own virtual machine. In nested virtualization, a hypervisor can run multiple other hypervisors with their associated virtual machines. As operating systems gain hypervisor functionality--Microsoft Windows 7 already runs Windows XP in a virtual machine--nested virtualization will become necessary in hypervisors that wish to host them. We present the design, implementation, analysis, and evaluation of high-performance nested virtualization on Intel x86-based systems. The Turtles project, which is part of the Linux/KVM

hypervisor, runs multiple unmodified hypervisors (e.g., KVM and VMware) and operating systems (e.g., Linux and Windows). Despite the lack of architectural support for nested virtualization in the x86 architecture, it can achieve performance that is within 6-8% of single-level (non-nested) virtualization for common workloads, through multi-dimensional paging for MMU virtualization and multi-level device assignment for I/O virtualization.
Significance: The paper describes nested virtualization where a hypervisor can run multiple other hypervisors with their associated virtual machines.

SnowFlock: rapid virtual machine cloning for cloud computing [300]
Horacio Andrés 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 **Abstract:** 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.
Significance: Describes how to quickly create copies of a virtual machine in the cloud for efficient task replication and deployment.

SnowFlock: Virtual Machine Cloning as a First-Class Cloud Primitive [300]
Horacio Andrés Lagar-Cavilla, Joseph Andrew Whitney, Adin Matthew Scannell, Philip Patchin, Stephen M. Rumble, Eyal de Lara, Michael Brudno, and Mahadev Satyanarayanan. 2011. SnowFlock: Virtual Machine Cloning as a First-Class Cloud Primitive. Trans. Computer Systems 29, 1, Article 2 (February 2011), 45 pages. DOI=10.1145/1925109.1925111 http://doi.acm.org/10.1145/1925109.1925111 **Abstract:** A basic building block of cloud computing is virtualization. Virtual machines (VMs) encapsulate a user's computing environment and efficiently isolate it from that of other users. VMs, however, are large entities, and no clear APIs exist yet to provide users with programmatic, fine-grained control on short time scales. We present SnowFlock, a paradigm and system for cloud computing that introduces VM cloning as a first-class cloud abstraction. VM cloning exploits the well-understood and effective semantics of UNIX fork. We demonstrate multiple usage models of VM cloning: users can incorporate the primitive in their code, can wrap around existing tool chains via scripting, can encapsulate the API within a parallel programming framework, or can use it to load-balance and self-scale clustered servers. VM cloning needs to be efficient to be usable. It must efficiently transmit VM state in order to avoid cloud I/O bottlenecks. We demonstrate how the semantics of cloning aid us in realizing its efficiency: state is propagated in parallel to multiple VM clones, and is

transmitted during runtime, allowing for optimizations that substantially reduce the I/O load. We show detailed micro benchmark results highlighting the efficiency of our optimizations, and macro benchmark numbers demonstrating the effectiveness of the different usage models of SnowFlock.
Significance: 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 [300]
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 **Abstract:** 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.
Significance: 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

CloudCmp: Comparing Public Cloud Providers [100]
Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. 2010. CloudCmp: comparing public cloud providers. In *Proceedings of the 10th annual conference on Internet measurement* (IMC '10). ACM, New York, NY, USA, 1-14. DOI=10.1145/1879141.1879143 http://doi.acm.org/10.1145/1879141.1879143 **Abstract:** While many public cloud providers offer pay-as-you-go computing, their varying approaches to infrastructure, virtualization, and software services lead to a problem of plenty. To help customers pick a cloud that fits their needs, we develop CloudCmp, a systematic comparator of the performance and cost of cloud

providers. CloudCmp measures the elastic computing, persistent storage, and networking services offered by a cloud along metrics that directly reflect their impact on the performance of customer applications. CloudCmp strives to ensure fairness, representativeness, and compliance of these measurements while limiting measurement cost. Applying CloudCmp to four cloud providers that together account for most of the cloud customers today, we find that their offered services vary widely in performance and costs, underscoring the need for thoughtful provider selection. From case studies on three representative cloud applications, we show that CloudCmp can guide customers in selecting the best-performing provider for their applications.
Significance: Systematically compares the major cloud system providers to better understand the performance-cost tradeoffs faced by customers and their services.

Cloud monitoring: A survey [100]
Giuseppe Aceto, Alessio Botta, Walter De Donato, and Antonio Pescapè. 2013. Cloud monitoring: A survey. *Comput. Netw.* 57, 9 (June 2013), 2093-2115. DOI=10.1016/j.comnet.2013.04.001 http://dx.doi.org/10.1016/j.comnet.2013.04.001 **Abstract:** Nowadays, Cloud Computing is widely used to deliver services over the Internet for both technical and economical reasons. The number of Cloud-based services has increased rapidly and strongly in the last years, and so is increased the complexity of the infrastructures behind these services. To properly operate and manage such complex infrastructures effective and efficient monitoring is constantly needed. Many works in literature have surveyed Cloud properties, features, underlying technologies (e.g. virtualization), security and privacy. However, to the best of our knowledge, these surveys lack a detailed analysis of monitoring for the Cloud. To fill this gap, in this paper we provide a survey on Cloud monitoring. We start analyzing motivations for Cloud monitoring, providing also definitions and background for the following contributions. Then, we carefully analyze and discuss the properties of a monitoring system for the Cloud, the issues arising from such properties and how such issues have been tackled in literature. We also describe current platforms, both commercial and open source, and services for Cloud monitoring, underlining how they relate with the properties and issues identified before. Finally, we identify open issues, main challenges and future directions in the field of Cloud monitoring.
Significance: Provides a survey on cloud monitoring services.

Challenges in Building Scalable Virtualized Datacenter Management [200]
Vijayaraghavan Soundararajan and Kinshuk Govil. 2010. Challenges in building scalable virtualized datacenter management. *SIGOPS Oper. Syst. Rev.* 44, 4 (December 2010), 95-102. DOI=10.1145/1899928.1899941 http://doi.acm.org/10.1145/1899928.1899941 **Abstract:** Virtualization drives higher resource utilization and makes provisioning new systems very easy and cheap. This combination has led to an ever-increasing number of virtual machines: the largest data centers will likely have more than 100K in few years, and many deployments will span multiple data centers. Virtual machines are also getting increasingly more capable, consisting of more vCPUs, more memory, and higher-bandwidth virtual I/O devices with a variety of capabilities like bandwidth throttling and traffic mirroring. To reduce the work for IT administrators managing these environments, VMware and other companies provide several monitoring, automation, and policy-driven tools. These tools require a lot of information about various aspects of each VM and other objects in the system, such as physical hosts, storage infrastructure, and networking. To support these tools and the hundreds of simultaneous users

who manage the environment, the management software needs to provide secure access to the data in real-time with some degree of consistency and backward compatibility, and very high availability under a variety of failures and planned maintenance. Such software must satisfy a continuum of designs: it must perform well at large-scale to accommodate the largest datacenters, but it must also accommodate smaller deployments by limiting its resource consumption and overhead according to demand. The need for high-performance, robust management tools that scale from a few hosts to cloud-scale poses interesting challenges for the management software. This paper presents some of the techniques we have employed to address these challenges.
Significance: Discusses the challenges of creating a scalable management infrastructure for systems with thousands of virtual machines in a virtualized datacenter and presents VMWare's vSphere architecture.

Characterizing Cloud Computing Hardware Reliability [200]
Kashi Venkatesh Vishwanath and Nachiappan Nagappan. 2010. Characterizing cloud computing hardware reliability. In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10). ACM, New York, NY, USA, 193-204. DOI=10.1145/1807128.1807161 http://doi.acm.org/10.1145/1807128.1807161 **Abstract:** Modern day datacenters host hundreds of thousands of servers that coordinate tasks in order to deliver highly available cloud computing services. These servers consist of multiple hard disks, memory modules, network cards, processors etc., each of which while carefully engineered are capable of failing. While the probability of seeing any such failure in the lifetime (typically 3-5 years in industry) of a server can be somewhat small, these numbers get magnified across all devices hosted in a datacenter. At such a large scale, hardware component failure is the norm rather than an exception. Hardware failure can lead to a degradation in performance to end-users and can result in losses to the business. A sound understanding of the numbers as well as the causes behind these failures helps improve operational experience by not only allowing us to be better equipped to tolerate failures but also to bring down the hardware cost through engineering, directly leading to a saving for the company. To the best of our knowledge, this paper is the first attempt to study server failures and hardware repairs for large datacenters. We present a detailed analysis of failure characteristics as well as a preliminary analysis on failure predictors. We hope that the results presented in this paper will serve as motivation to foster further research in this area.
Significance: Reports results from a study of server failures in datacenters and their repairs.

Automated Control in Cloud Computing: Challenges and Opportunities [300]
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 **Abstract:** 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.
Significance: Discusses the challenges of adaptive resource provisioning to meet elastic service demands and argues for placing control in the hands of cloud customers.

CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [300]
Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exper.* 41, 1 (January 2011), 23-50. DOI=10.1002/spe.995 http://dx.doi.org/10.1002/spe.995 **Abstract:** Cloud computing is a recent advancement wherein IT infrastructure and applications are provided as "services" to end-users under a usage-based payment model. It can leverage virtualized services even on the fly based on requirements (workload patterns and QoS) varying with time. The application services hosted under Cloud computing model have complex provisioning, composition, configuration, and deployment requirements. Evaluating the performance of Cloud provisioning policies, application workload models, and resources performance models in a repeatable manner under varying system and user configurations and requirements is difficult to achieve. To overcome this challenge, we propose CloudSim: an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited effort. Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). Moreover, it exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios. Several researchers from organizations, such as HP Labs in U.S.A., are using CloudSim in their investigation on Cloud resource provisioning and energy-efficient management of data center resources. The usefulness of CloudSim is demonstrated by a case study involving dynamic provisioning of application services in the hybrid federated clouds environment. The result of this case study proves that the federated Cloud computing model significantly improves the application QoS requirements under fluctuating resource and service demand patterns.
Significance: Describes a simulation toolkit to model and simulate cloud computing systems and application provisioning environments.

Detecting Large-Scale System Problems by Mining Console Logs [300]
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 **Abstract:**

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

Quincy: Fair Scheduling for Distributed Computing Clusters [300]
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 **Abstract:** 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 data structure, 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%.
Significance: Describes a new approach for scheduling concurrent jobs in a computing cluster that places computations near their data while also taking fairness into account.

COMMUNICATIONS

High-speed, scalable, reliable networking is required for transferring data within the cloud and between the cloud and external clients. What networking protocols are suitable? How might applications take advantage of higher level communication protocols such as multicast, reliable message queues, and pub-sub systems?

READINGS

OpenFlow: A Radical New Idea in Networking [100]
Thomas A. Limoncelli. 2012. OpenFlow: A Radical New Idea in Networking. *Queue* 10, 6, Pages 40 (June 2012), 7 pages. DOI=10.1145/2246036.2305856 http://doi.acm.org/10.1145/2246036.2305856 **Abstract:** Computer networks have historically evolved box by box, with individual network elements occupying specific ecological niches as routers, switches, load balancers, NATs (network address translations), or firewalls. Software-defined networking proposes to overturn that ecology, turning the network as a whole into a platform and the individual network elements into programmable entities. The apps running on the network platform can optimize traffic flows to take the shortest path, just as the current distributed protocols do, but they can also optimize the network to maximize link utilization, create different reachability domains for different users, or make device mobility seamless.
Significance: An open standard that enables software-defined networking.

A Cost Comparison of Datacenter Network Architectures [300]
Lucian Popa, Sylvia Ratnasamy, Gianluca Iannaccone, Arvind Krishnamurthy, and Ion Stoica. 2010. A cost comparison of datacenter network architectures. In *Proceedings of the 6th International Conference* (Co-NEXT '10). ACM, New York, NY, USA, Article 16 , 12 pages. DOI=10.1145/1921168.1921189 http://doi.acm.org/10.1145/1921168.1921189 **Abstract:** There is a growing body of research exploring new network architectures for the data center. These proposals all seek to improve the scalability and cost-effectiveness of current data center networks, but adopt very different approaches to doing so. For example, some proposals build networks entirely out of switches while others do so using a combination of switches and servers. How do these different network architectures compare? For that matter, by what metrics should we even begin to compare these architectures? Understanding the tradeoffs between different approaches is important both for operators making deployment decisions and to guide future research. In this paper, we take a first step toward understanding the tradeoffs between different data center network architectures. We use high-level models of different classes of data center networks and compare them on cost using both current and predicted trends in cost and power consumption.
Significance: Uses a new methodology to estimate and compare the costs of several datacenter network architectures for similar performance targets.

Cloud Control with Distributed Rate Limiting [300]
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 **Abstract:** 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 tradeoff 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.
Significance: 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 [300]
Fang Hao, T. V. Lakshman, Sarit Mukherjee, and Haoyu Song. 2010. Enhancing dynamic cloud-based services using network virtualization. *SIGCOMM Computer Communication Review* 40, 1 (January 2010), 67-74. DOI=10.1145/1672308.1672322
http://doi.acm.org/10.1145/1672308.1672322 **Abstract:** 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.
Significance: Presents a virtualized network architecture that permits seamless migration of virtual machines within the cloud.

HyperFlow: A distributed control plane for OpenFlow [300]
Tootoonchian, Amin, and Yashar Ganjali. "HyperFlow: A distributed control plane for OpenFlow." In *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, pp. 3-3. USENIX Association, 2010. **Abstract:** OpenFlow assumes a logically centralized controller, which ideally can be physically distributed. However, current deployments rely on a single controller which has major drawbacks including lack of scalability. We present HyperFlow, a distributed event-based control plane for OpenFlow. HyperFlow is logically centralized but physically distributed: it provides scalability while keeping the benefits of network control centralization. By passively synchronizing network-wide views of OpenFlow

controllers, HyperFlow localizes decision making to individual controllers, thus minimizing the control plane response time to data plane requests. HyperFlow is resilient to network partitioning and component failures. It also enables interconnecting independently managed OpenFlow networks, an essential feature missing in current OpenFlow deployments. We have implemented HyperFlow as an application for NOX. Our implementation requires minimal changes to NOX, and allows reuse of existing NOX applications with minor modifications. Our preliminary evaluation shows that, assuming sufficient control bandwidth, to bound the window of inconsistency among controllers by a factor of the delay between the farthest controllers, the network changes must occur at a rate lower than 1000 events per second across the network.
Significance: Presents a distributed event-based control plane for OpenFlow.

PortLand: a Scalable Fault-Tolerant Layer 2 Data Center Network Fabric [300]
Radhika Niranjan 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 **Abstract:** 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.
Significance: Introduces a new routing and forwarding protocol designed for a more scalable, fault-tolerant, and manageable datacenter network.

The Cost of a Cloud: Research Problems in Data Center Networks [300]
Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. 2008. The cost of a cloud: research problems in data center networks. *SIGCOMM Computer Communications Review* 39, 1 (December 2008), 68-73. DOI=10.1145/1496091.1496103 http://doi.acm.org/10.1145/1496091.1496103 **Abstract:** 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.
Significance: 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.

VL2: a scalable and flexible data center network [300]
Albert Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. 2011. VL2: a scalable and flexible data center network. *Commun. ACM* 54, 3 (March 2011), 95-104. DOI=10.1145/1897852.1897877 http://doi.acm.org/10.1145/1897852.1897877 **Abstract:** To be agile and cost effective, data centers must allow dynamic resource allocation across large server pools. In particular, the data center network should provide a simple flat abstraction: it should be able to take any set of servers anywhere in the data center and give them the illusion that they are plugged into a physically separate, noninterfering Ethernet switch with as many ports as the service needs. To meet this goal, we present VL2, a practical network architecture that scales to support huge data centers with uniform high capacity between servers, performance isolation between services, and Ethernet layer-2 semantics. VL2 uses (1) flat addressing to allow service instances to be placed anywhere in the network, (2) Valiant Load Balancing to spread traffic uniformly across network paths, and (3) end system--based address resolution to scale to large server pools without introducing complexity to the network control plane. VL2's design is driven by detailed measurements of traffic and fault data from a large operational cloud service provider. VL2's implementation leverages proven network technologies, already available at low cost in high-speed hardware implementations, to build a scalable and reliable network architecture. As a result, VL2 networks can be deployed today, and we have built a working prototype. We evaluate the merits of the VL2 design using measurement, analysis, and experiments. Our VL2 prototype shuffles 2.7 TB of data among 75 servers in 395 sâ€”sustaining a rate that is 94% of the maximum possible.
Significance: Shows how to build a datacenter network that provides the illusion that all of the servers allocated to a given service are connected by a high capacity, dedicated Ethernet.

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') 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 [100]
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 **Abstract:** 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.
Significance: Examines the concerns that are preventing corporations from placing sensitive information in the cloud and suggests research directions to address these concerns.

Home is safer than the cloud!: privacy concerns for consumer cloud storage [100]
Iulia Ion, Niharika Sachdeva, Ponnurangam Kumaraguru, and Srdjan Capkun. 2011. Home is safer than the cloud!: privacy concerns for consumer cloud storage. In *Proceedings of the Seventh Symposium on Usable Privacy and Security* (SOUPS '11). ACM, New York, NY, USA, , Article 13 , 20 pages. DOI=10.1145/2078827.2078845 http://doi.acm.org/10.1145/2078827.2078845 **Abstract:** Several studies ranked security and privacy to be major areas of concern and impediments of cloud adoption for companies, but none have looked into end-users' attitudes and practices. Not much is known about consumers' privacy beliefs and expectations for cloud storage, such as web-mail, document and photo sharing platforms, or about users' awareness of contractual terms and conditions. We conducted 36 in-depth interviews in Switzerland and India (two countries with different privacy perceptions and expectations); and followed up with an online survey with 402 participants in both countries. We study users' privacy attitudes and beliefs regarding their use of cloud storage systems. Our results show that privacy requirements for consumer cloud storage differ from those of companies. Users are less concerned about some issues, such as guaranteed deletion of data, country of storage and storage outsourcing, but are uncertain about using cloud storage. Our results further show that end-users consider the Internet intrinsically insecure and prefer local storage for sensitive data over cloud storage. However, users desire better security and are ready to pay for services that provide strong privacy guarantees. Participants had misconceptions about the rights and guarantees their cloud storage providers offers. For example, users believed that their provider is liable in case of data loss, does not have the right to view and modify user data, and cannot disable user accounts. Finally, our results show that cultural differences greatly influence user attitudes and beliefs, such as their willingness to store sensitive data in the cloud and their acceptance that law enforcement agencies monitor user accounts. We believe that these observations can help in improving users privacy in cloud storage systems.
Significance: Presents a study of privacy attitudes and beliefs regarding use of cloud storage systems.

Taking account of privacy when designing cloud computing services [100]
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
Abstract: 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.
Significance: 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.

Trusting the Cloud [100]
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 **Abstract:** 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.
Significance: Outlines cryptographic techniques for enforcing the integrity and consistency of data stored in the cloud.

A Client-Based Privacy Manager for Cloud Computing [300]
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 **Abstract:** 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.
Significance: Describes a privacy manager that allows clients to control their sensitive information in cooperation with cloud service providers.

A Cloud Trust Model in a Security Aware Cloud [300]
Hiroyuki Sato, Atsushi Kanai, and Shigeaki Tanimoto. 2010. A Cloud Trust Model in a Security Aware Cloud. In *Proceedings of the 2010 10th IEEE/IPSJ International Symposium on Applications and the Internet* (SAINT '10). IEEE Computer Society, Washington, DC, USA, 121-124. DOI=10.1109/SAINT.2010.13 http://dx.doi.org/10.1109/SAINT.2010.13 **Abstract:** Faced with today's innovative blow-up of cloud technologies, we are forced to rebuild services in terms of cloud. In the rebuilding, considering the facet of cloud as a social infrastructure, security is a critical problem of clouds. Most of insecurity against clouds can be summarized as social insecurity, which is classified into the multiple stakeholder problem, the open space security problem, and the mission critical data handling problem. As a solution of those problems, we propose a new cloud trust model. In our cloud trust model, in addition to conventional trust

models, we consider both internal trust that is the TPM of this model, and contracted trust that controls cloud service providers under contracts and related documents. We call the cloud platform that meets the cloud trust model as "Security Aware Cloud." In a security aware cloud, internal trust must be established as the firm base of trust. By implementing TPM of security such as Id management and key management on internal trust, we obtain a firm trust model. Moreover, by controlling levels of quality of service and security by contract, we can optimize ROI on service and security delegated to a cloud.
Significance: Proposed a new cloud trust model taking into account both internal trust, and contracted trust that controls cloud service providers under contracts and related documents.

Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds [300]
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 **Abstract:** 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.
Significance: Shows how customers in a cloud can perform side-channel attacks on virtual machines to extract private information from other customers.

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

Cloud SLAs: present and future [100]
Salman A. Baset. 2012. Cloud SLAs: present and future. *SIGOPS Oper. Syst. Rev.* 46, 2 (July 2012), 57-66. DOI=10.1145/2331576.2331586 http://doi.acm.org/10.1145/2331576.2331586
Abstract: The variability in the service level agreements (SLAs) of cloud providers prompted us to ask the question how do the SLAs compare and how should the SLAs be defined for future cloud services. We break down a cloud SLA into easy to understand components and use it to compare SLAs of public cloud providers. Our study indicates that none of the surveyed cloud

providers offer any performance guarantees for compute services and leave SLA violation detection to the customer. We then provide guidance on how SLAs should be defined for future cloud services.
Significance: Breaks down SLAs into easy to understand components to compare SLAs of public cloud providers.

Aggregation Patterns of Service Level Agreements [300]
Irfan Ul Haq and Erich Schikuta. 2010. Aggregation Patterns of Service Level Agreements. In *Proceedings of the 8th International Conference on Frontiers of Information Technology* (FIT '10). ACM, New York, NY, USA, Article 40, 6 pages. DOI=10.1145/1943628.1943668 http://doi.acm.org/10.1145/1943628.1943668 **Abstract:** IT-based Service Markets require an enabling infrastructure to support Service Value Chains and service choreographies resulting from service composition scenarios. This will result into novel business models where services compose together hierarchically in a producer-consumer manner to form service chains of added value. Service Level Agreements (SLAs) are defined at various levels in such a hierarchy to ensure the expected quality of service for different stakeholders. Automation of service composition directly implies the aggregation of their corresponding SLAs. Aggregation of hierarchical SLAs leads to cross-enterprise business networks such as Virtual Enterprise Organizations (VEO), Extended Enterprises and Value Networks. During the hierarchical aggregation of SLAs, certain SLA information pertaining to different stakeholders is meant to be restricted and can be only partially revealed to a subset of their business partners. Based on our notions of SLA Choreography and SLA-Views, we formally define an aggregation model to enable the automation of hierarchical aggregation of Service Level Agreements. The aggregation model leads to the discovery of various aggregation patterns in context with service composition and business collaboration.
Significance: Introduces hierarchical aggregation of SLAs for service value chains composed from different external services.

An SLA-based resource virtualization approach for on-demand service provision [300]
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 **Abstract:** 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.
Significance: Shows how to incorporate service level agreements when provisioning virtualized resources for cloud services.

Automatic Exploration of Datacenter Performance Regimes [300]
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
Abstract: 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.
Significance: Presents new techniques for developing accurate performance models that can aid in configuring system services and avoid violating service level agreements.

The SLA Evaluation Model for Cloud Computing [300]
Chenkang Wu, Yonghua Zhu, Shunhong Pan, "The SLA Evaluation Model for Cloud Computing," *International Conference on Computer, Networks and Communication Engineering* (ICCNCE 2013), May 2013 DOI: 10.2991/iccnce.2013.83
http://dx.doi.org/10.2991/iccnce.2013.83 **Abstract:** Cloud Computing is an emerging technology to deliver the infrastructure, network, software, and even the development environment as a service in the philosophy of pay-as-you-use. To guarantee all participants' rights in the cloud, SLA is suggested to be a necessary contract which can protect users and service providers from violation. However, SLA is an early-proposed method that not suitable for new characteristics of cloud computing. This paper present an optimized SLA model for cloud computing and a corresponding evaluation method, which can determine a more fair and reliable evaluation result of cloud service. With this model, users can simplify their selection process, while providers can use the evaluation result for further resource allocation strategy.
Significance: Discusses an SLA model for cloud computing.

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

Cutting the Electric Bill for Internet-Scale Systems [300]
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 **Abstract:** 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 U.S., 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.
Significance: 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 [300]
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 **Abstract:** 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.
Significance: Describes an architecture that reduces energy consumption in a datacenter through on-line monitoring and migration of virtual machines.

Performance and Power Management for Cloud Infrastructures [300]
Hien Nguyen Van, Frédéric Dang Tran, and Jean-Marc Menaud. 2010. Performance and Power Management for Cloud Infrastructures. In *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing* (CLOUD '10). IEEE Computer Society, Washington, DC, USA, 329-336. DOI=10.1109/CLOUD.2010.25 http://dx.doi.org/10.1109/CLOUD.2010.25
Abstract: A key issue for Cloud Computing data-centers is to maximize their profits by minimizing power consumption and SLA violations of hosted applications. In this paper, we propose a resource management framework combining a utility-based dynamic Virtual Machine

provisioning manager and a dynamic VM placement manager. Both problems are modeled as constraint satisfaction problems. The VM provisioning process aims at maximizing a global utility capturing both the performance of the hosted applications with regard to their SLAs and the energy-related operational cost of the cloud computing infrastructure. We show several experiments how our system can be controlled through high level handles to make different trade-off between application performance and energy consumption or to arbitrate resource allocations in case of contention.
Significance: Proposes a resource management framework combining a utility-based dynamic Virtual Machine provisioning manager and a dynamic VM placement manager.

Power Provisioning for a Warehouse-Sized Computer [300]
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 **Abstract:** 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.
Significance: Presents results from a study of the power consumption of large clusters of servers and suggests opportunities for significant energy savings.

Virtual Machine Power Metering and Provisioning [300]
Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. 2010. Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10). ACM, New York, NY, USA, 39-50. DOI=10.1145/1807128.1807136 http://doi.acm.org/10.1145/1807128.1807136 **Abstract:** Virtualization is often used in cloud computing platforms for its several advantages in efficiently managing resources. However, virtualization raises certain additional challenges, and one of them is lack of power metering for

virtual machines (VMs). Power management requirements in modern data centers have led to most new servers providing power usage measurement in hardware and alternate solutions exist for older servers using circuit and outlet level measurements. However, VM power cannot be measured purely in hardware. We present a solution for VM power metering, named Joulemeter. We build power models to infer power consumption from resource usage at runtime and identify the challenges that arise when applying such models for VM power metering. We show how existing instrumentation in server hardware and hypervisors can be used to build the required power models on real platforms with low error. Our approach is designed to operate with extremely low runtime overhead while providing practically useful accuracy. We illustrate the use of the proposed metering capability for VM power capping, a technique to reduce power provisioning costs in data centers. Experiments are performed on server traces from several thousand production servers, hosting Microsoft's real-world applications such as Windows Live Messenger. The results show that not only does VM power metering allows virtualized data centers to achieve the same savings that non-virtualized data centers achieved through physical server power capping, but also that it enables further savings in provisioning costs with virtualization.
Significance: Extends schemes for power monitoring of physical server machines to virtual machines that share the same hardware.

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

A survey of mobile cloud computing: architecture, applications, and approaches [100]
Hoang T. Dinh, Chonho Lee, Dusit Niyato and Ping Wang. 2013. A survey of mobile cloud computing: architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.*, 13, Issue 18, pages 1587-1611, 25 December 2013. DOI=10.1002/wcm.1203 http://dx.doi.org/10.1002/wcm.1203 **Abstract:** Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobile cloud computing (MCC) has been introduced to be a potential technology for mobile services. MCC integrates the cloud computing into the mobile environment and overcomes obstacles related to the performance (e.g., battery life, storage, and bandwidth), environment (e.g., heterogeneity, scalability, and availability), and security (e.g., reliability and privacy) discussed in mobile computing. This paper gives a survey of MCC, which helps general readers have an overview of the MCC including the definition, architecture, and applications. The issues, existing solutions, and approaches are presented. In addition, the future research directions of MCC are discussed.
Significance: Presents a survey of Mobile cloud computing including the definition, architecture, and applications.

A Survey of Research on Mobile Cloud Computing [100]
Le Guan, Xu Ke, Meina Song, and Junde Song. 2011. A Survey of Research on Mobile Cloud

Computing. In *Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science* (ICIS '11). IEEE Computer Society, Washington, DC, USA, 387-392. DOI=10.1109/ICIS.2011.67 http://dx.doi.org/10.1109/ICIS.2011.67 **Abstract:** The rapid development of mobile computing and cloud computing trigger novel computing paradigm----- Mobile Cloud Computing. This paper review current research effort towards Mobile Computing. First, we present several challenges for the design of Mobile Cloud Computing service. Second, a concept model has been proposed to analyze related research work. Third, we survey recent Mobile Cloud Computing architecture, application partition & offloading, and context-aware service.
Significance: Reviews current research effort including challenges and applications of mobile computing.

Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing [200]
Xinwen Zhang, Anugeetha Kunjithapatham, Sangoh Jeong, and Simon Gibbs. 2011. Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing. *Mob. Netw. Appl.* 16, 3 (June 2011), 270-284. DOI=10.1007/s11036-011-0305-7 http://dx.doi.org/10.1007/s11036-011-0305-7 **Abstract:** We propose a new elastic application model that enables seamless and transparent use of cloud resources to augment the capability of resource-constrained mobile devices. The salient features of this model include the partition of a single application into multiple components called weblets, and a dynamic adaptation of weblet execution configuration. While a weblet can be platform independent (e.g., Java or .Net bytecode or Python script) or platform dependent (native code), its execution location is transparent—it can be run on a mobile device or migrated to the cloud, i.e., run on one or more nodes offered by an IaaS provider. Thus, an elastic application can augment the capabilities of a mobile device including computation power, storage, and network bandwidth, with the light of dynamic execution configuration according to device's status including CPU load, memory, battery level, network connection quality, and user preferences. This paper presents the motivation behind developing elastic applications and their architecture including typical elasticity patterns and cost models that are applied to determine the elasticity patterns. We implement a reference architecture and develop a set of elastic applications to validate the augmentation capabilities for smartphone devices. We demonstrate promising results of the proposed application model using data collected from one of our example elastic applications. We first analyze NVM technology scaling trends, and then propose a cloud service cache architecture that resides on the mobile device's NVM (pocket cloudlet). This architecture utilizes both individual user and community access models to maximize its hit rate, and subsequently reduce overall service latency and energy consumption. As a showcase we present the design, implementation and evaluation of PocketSearch, a search and advertisement pocket cloudlet. We perform mobile search characterization to guide the design of PocketSearch and evaluate it with 200 million mobile queries from the search logs of m.bing.com. We show that PocketSearch can serve, on average, 66% of the web search queries submitted by an individual user without having to use the slow 3G link, leading to 16x service access speedup. Finally, based on experience with PocketSearch we provide additional insight and guidelines on how future pocket cloudlets should be organized, from both an architectural and an operating system perspective.
Significance: Presents a technique for dynamically partitioning an application between a mobile

device and the cloud using weblets that are configured and replicated based on elasticity patterns and cost considerations.

Using RESTful Web-Services and Cloud Computing to Create Next Generation Mobile Applications [200]
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 **Abstract:** 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.
Significance: Explores architectures for mobile applications that access cloud-based services.

WhereStore: location-based data storage for mobile devices interacting with the cloud [200]
Patrick Stuedi, Iqbal Mohomed, and Doug Terry. 2010. WhereStore: location-based data storage for mobile devices interacting with the cloud. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond* (MCS '10). ACM, New York, NY, USA, Article 1, 8 pages. DOI=10.1145/1810931.1810932 http://doi.acm.org/10.1145/1810931.1810932 **Abstract:** In recent years, two major trends have changed the way mobile phones are used: smartphones have become a platform for applications, and 3G connectivity has turned them into ubiquitous Internet clients. Increasingly, applications on smartphones (such as document sharing, media players and map browsers) interact with the cloud as a backend for data storage and computation. We observe that, for many mobile applications, the specific data that is accessed depends on the current location of the user. For example, a restaurant recommendation application is often used to get information about nearby restaurants. In this paper, we present WhereStore, a location-based data store for smart-phones interacting with the cloud. It uses filtered replication along with each device's location history to distribute items between smartphones and the cloud. We discuss the challenges of designing such a system, relevant applications, and a specific design and prototype implementation.
Significance: Shows how to replicate data between the cloud and mobile devices in a location-aware manner.

Elastic Mobility: Stretching Interaction [300]
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 **Abstract:** 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.
Significance: Reflects on how cloud computing will augment applications on mobile devices, and vice versa, particularly for context-aware interaction.

Energy efficiency of mobile clients in cloud computing [300]
Antti P. Miettinen and Jukka K. Nurminen. 2010. Energy efficiency of mobile clients in cloud computing. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* (HotCloud'10). USENIX Association, Berkeley, CA, USA, 4-4. **Abstract:** Energy efficiency is a fundamental consideration for mobile devices. Cloud computing has the potential to save mobile client energy but the savings from offloading the computation need to exceed the energy cost of the additional communication. In this paper we provide an analysis of the critical factors affecting the energy consumption of mobile clients in cloud computing. Further, we present our measurements about the central characteristics of contemporary mobile handheld devices that define the basic balance between local and remote computing. We also describe a concrete example, which demonstrates energy savings. We show that the trade-offs are highly sensitive to the exact characteristics of the workload, data communication patterns and technologies used, and discuss the implications for the design and engineering of energy efficient mobile cloud computing solutions.
Significance: Presents an analysis of the critical factors affecting the energy consumption of mobile client in cloud computing.

Pocket cloudlets [300]
Emmanouil Koukoumidis, Dimitrios Lymberopoulos, Karin Strauss, Jie Liu, and Doug Burger. 2011. Pocket cloudlets. In *Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems* (ASPLOS '11). ACM, New York, NY, USA, 171-184. DOI=10.1145/1950365.1950387 http://doi.acm.org/10.1145/1950365.1950387 **Abstract:** Cloud services accessed through mobile devices suffer from high network access latencies and are constrained by energy budgets dictated by the devices' batteries. Radio and battery technologies will improve over time, but are still expected to be the bottlenecks in future systems. Non-volatile memories (NVM), however, may continue experiencing significant and steady improvements in density for at least ten more years. In this paper, we propose to leverage the abundance in memory capacity of mobile devices to mitigate latency and energy issues when accessing cloud services. We first analyze NVM technology scaling trends, and then propose a cloud service cache architecture that resides on the mobile device's NVM (pocket cloudlet). This architecture utilizes both individual user and community access models to maximize its hit rate, and subsequently reduce overall service

latency and energy consumption. As a showcase we present the design, implementation and evaluation of PocketSearch, a search and advertisement pocket cloudlet. We perform mobile search characterization to guide the design of PocketSearch and evaluate it with 200 million mobile queries from the search logs of m.bing.com. We show that PocketSearch can serve, on average, 66% of the web search queries submitted by an individual user without having to use the slow 3G link, leading to 16x service access speedup. Finally, based on experience with PocketSearch we provide additional insight and guidelines on how future pocket cloudlets should be organized, from both an architectural and an operating system perspective.
Significance: Evaluates how large non-volatile memories on devices can effectively cache cloud data to alleviate communication and power bottlenecks.