

A View of Cloud Computing

Bhagwan Singh⁽¹⁾, Sumeet Dhawan⁽²⁾, Anish Arora⁽³⁾, Abhishek Patail⁽⁴⁾

⁽¹⁾Asst. Prof, Ferozpur College of Engineering & Technology, Ferozpur
bhagwan87singh@gmail.com

⁽²⁾Asst. Prof, Ferozpur College of Engineering & Technology, Ferozpur
sumeet_dhawan87@yahoo.co.in

⁽³⁾Asst. Prof, Ferozpur College of Engineering & Technology, Ferozpur
anisharora87@gmail.com

⁽⁴⁾Asst. Prof, Ferozpur College of Engineering & Technology, Ferozpur
abhishekpatail88@gmail.com

ABSTRACT

In this work we strive to frame the full space of cloud-computing security issues, attempting to separate justified concerns from possible over-reactions. We examine contemporary and historical perspectives from industry, academia, government, and “black hats”. We argue that few cloud computing security issues are fundamentally new or fundamentally intractable; often what appears “new” is so only relative to “traditional” computing of the past several years. Looking back further to the time-sharing era, many of these problems already received attention. On the other hand, we argue that two facets are to some degree new and fundamental to cloud computing: the complexities of multi-party trust considerations, and the ensuing need for mutual audit ability.

KEYWORDS

Cloud Computing, Utility Computing, Internet Data enters, Distributed System Economics

Executive Summary

Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about over provisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or under provisioning for one that becomes wildly popular, thus missing potential customers and revenue. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1000 servers for one hour costs no more than using one server for 1000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT. Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data enters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The data enter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. People can be users or providers of SaaS, or users or providers of Utility Computing. We focus on SaaS Providers (Cloud Users) and Cloud Providers, which have received less attention than SaaS Users. From a hardware point of view, three aspects are new in Cloud Computing.

1. The illusion of infinite computing resources available on demand, thereby eliminating the need for Cloud Computing users to plan far ahead for provisioning.
2. The elimination of an up-front commitment by Cloud users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs.
3. The ability to pay for use of computing resources on a short-term basis as needed (e.g., processors by the hour and storage by the day) and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful.

We argue that the construction and operation of extremely large-scale, commodity-computer data enters at low-cost locations was the key necessary enabler of Cloud Computing, for they uncovered the factors of 5 to 7 decrease in cost of electricity, network bandwidth, operations, software, and hardware available at these very large economies. The RAD Lab's existence is due to the generous support of the founding members Google, Microsoft, and Sun Microsystems and of the affiliate

members Amazon Web Services, Cisco Systems, Face book, Hewlett-Packard, IBM, NEC, Network Appliance, Oracle, Siemens, and VMware; by matching funds from the State of California's MICRO program (grants 06-152, 07-010, 06 148, 07-012, 06-146, 07-009, 06-147, 07-013, 06-149, 06-150, and 07-008) and the University of California Industry/University Cooperative Research Program (UC Discovery) grant COM07-10240; and by the National Science Foundation (grant #CNS-0509559). . These factors, combined with statistical multiplexing to increase utilization compared a private cloud, meant that cloud computing could offer services below the costs of a medium-sized datacenter and yet still make a good profit. Any application needs a model of computation, a model of storage, and a model of communication. The statistical multiplexing necessary to achieve elasticity and the illusion of infinite capacity requires each of these resources to be virtualized to hide the implementation of how they are multiplexed and shared. Our view is that different utility computing offerings will be distinguished based on the level of abstraction presented to the programmer and the level of management of the resources. Amazon EC2 is at one end of the spectrum. An EC2 instance looks much like physical hardware, and users can control nearly the entire software stack, from the kernel upwards. This low level makes it inherently difficult for Amazon to offer automatic scalability and failover, because the semantics associated with replication and other state management issues are highly application-dependent. At the other extreme of the spectrum are application domain specific platforms such as Google AppEngine. AppEngine is targeted exclusively at traditional web applications, enforcing an application structure of clean separation between a stateless computation tier and a stateful storage tier. AppEngine's impressive automatic scaling and high-availability mechanisms and the proprietary Megastore data storage available to AppEngine applications, all rely on these constraints. Applications for Microsoft's Azure are written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment. Thus, Azure is intermediate between application frameworks like AppEngine and hardware virtual machines like EC2. When is Utility Computing preferable to running a Private Cloud? A first case is when demand for a service varies with time. Provisioning a data center for the peak load it must sustain a few days per month leads to underutilization at other times, for example. Instead, Cloud Computing lets an organization pay by the hour for computing resources, potentially leading to cost savings even if the hourly rate to rent a machine from a cloud provider is higher than the rate to own one. A second case is when demand is unknown in advance. For example, a web start-up will need to support a spike in demand when it becomes popular, followed potentially by a reduction once some of the visitors turn away. Finally, organizations that perform batch analytics can use the "cost associatively" of cloud computing to finish computations faster: using 1000 EC2 machines for 1 hour costs the same as using 1 machine for 1000 hours. The left-hand side multiplies the net revenue per user-hour by the number of user-hours, giving the expected profit from using Cloud Computing. The right-hand side performs the same calculation for a fixed-capacity data center by factoring in the average utilization, including nonpeak workloads, of the data center. Whichever side is greater

represents the opportunity for higher profit. Table 1 below previews our ranked list of critical obstacles to growth of Cloud Computing in Section 7. The first three concern adoption, the next five affect growth, and the last two are policy and business obstacles. Each obstacle is

Paired with an opportunity, ranging from product development to research projects, which can overcome that obstacle. We predict Cloud Computing will grow, so developers should take it into account. All levels should aim at horizontal scalability of virtual machines over the efficiency on a single VM. In addition

1. Applications Software needs to both scale down rapidly as well as scale up, which is a new requirement. Such software also needs a pay-for-use licensing model to match needs of Cloud Computing.
2. Infrastructure Software needs to be aware that it is no longer running on bare metal but on VMs. Moreover, it needs to have billing built in from the beginning.
3. Hardware Systems should be designed at the scale of a container (at least a dozen racks), which will be is the minimum purchase size. Cost of operation will match performance and cost of purchase in importance, rewarding energy proportionality such as by putting idle portions of the memory, disk, and network into low power mode. Processors should work well with VMs, flash memory should be added to the memory hierarchy, and LAN switches and WAN routers must improve in bandwidth and cost

Cloud Computing

Cloud Computing is a new term for a long-held dream of computing as a utility [35], which has recently emerged as a commercial reality. Cloud Computing is likely to have the same impact on software that foundries have had on the following Table hardware industry. At one time, leading hardware companies required a captive semiconductor fabrication facility, and companies had to be large enough to afford to build and operate it economically. However, processing equipment doubled in price every technology generation. A semiconductor fabrication line costs over \$3B today, so only a handful of major "merchant" companies with very high chip volumes, such as Intel and Samsung, can still justify owning and operating their own fabrication lines. This motivated the rise of semiconductor foundries that build chips for others, such as Taiwan Semiconductor Manufacturing Company (TSMC). Foundries enable "fab-less" semiconductor chip companies whose value is in innovative chip design: A company such as nVidia can now be successful in the chip business without the capital, operational expenses, and risks associated with owning a state-of-the-art fabrication line. Conversely, companies with fabrication lines can time-multiplex their use among the products of many fab-less companies, to lower the risk of not having enough successful products to amortize operational costs. Similarly, the advantages of the economy of scale and statistical multiplexing may ultimately lead to a handful of Cloud Computing providers who can amortize the cost of their large datacenters over the products of many "datacenter-less" companies. Cloud Computing has been talked about [10], blogged about [13, 25], written about [15, 37, 38] and been featured in the

title of workshops, conferences, and even magazines. Nevertheless, confusion remains about exactly what it is and when it's useful, causing Oracle's CEO to vent his frustration

Table 1: Quick Preview of Top 10 Obstacles to and Opportunities for Growth of Cloud Computing.

	Obstacle	Opportunity
1	Availability of Service Use Multiple Cloud Providers;	Use Elasticity to Prevent DDOS
2	Data Lock-In	Standardize APIs; Compatible SW to enable Surge Computing
3	Data Confidentiality and Audit ability	Deploy Encryption, VLANs, Firewalls; Geographical Data Storage
4	Data Transfer Bottlenecks	Feeding Disks; Data Backup/Archival; Higher BW Switches
5	Performance Unpredictability	Improved VM Support; Flash Memory; Gang Schedule VMs
6	Scalable Storage	Invent Scalable Store
7	Bugs in Large Distributed Systems	Invent Debugger that relies on Distributed VMs
8	Scaling Quickly	Invent Auto-Scaler that relies on ML; Snapshots for Conservation
9	Reputation Fate Sharing	Offer reputation-guarding services like those for email
10	Software Licensing	Pay-for-use licenses; Bulk use sales

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS), so we use that term. The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the public, we call it a Public Cloud; the service being sold is Utility Computing. Current examples of public Utility Computing include Amazon Web Services, Google AppEngine, and Microsoft Azure. We use the term Private Cloud to refer to internal datacenters of a business or other organization that are not made available to the public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not normally include Private Clouds. We'll generally use Cloud Computing, replacing it with one of the other terms only when clarity demands it. Figure 1 shows the roles of the people as users or providers of these layers of Cloud Computing, and we'll use those terms to help make our arguments clear. The advantages of SaaS to both end users and service providers are well understood. Service providers enjoy greatly simplified software installation and maintenance and centralized control over versioning; end users can access the service "anytime, anywhere", share data and collaborate more easily, and keep their data stored safely in the infrastructure. Cloud Computing does not change these arguments, but it does give more application providers the choice of deploying their product as SaaS without provisioning a datacenter: just as the emergence of semiconductor foundries gave chip companies the opportunity to design and sell chips without owning a fab, Cloud Computing allows deploying SaaS—and scaling on demand—without building or provisioning a datacenter. Analogously to how SaaS allows the user to offload some problems to the SaaS provider, the SaaS provider can now offload some of his problems to the Cloud Computing provider. From now on, we will focus on issues related to the potential SaaS Provider (Cloud User) and to the Cloud Providers, which have received less attention. We will eschew terminology such as "X as a service (XaaS)"; values of X we have seen in print include Infrastructure, Hardware, and Platform, but we were unable to agree even among ourselves what the precise differences among them might be.¹ (We are using Endnotes instead of footnotes. Go to page 20 at the end of paper to read the notes, which have more details.) Instead, we present a simple classification of Utility Computing services in Section 5 that focuses on the tradeoffs among programmer convenience, flexibility, and portability, from both the cloud provider's and the cloud user's point of view. From a hardware point of view, three aspects are new in Cloud Computing [22]:

1. The illusion of infinite computing resources available on demand, thereby eliminating the need for Cloud Computing users to plan far ahead for provisioning;
2. The elimination of an up-front commitment by Cloud users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs; and
3. The ability to pay for use of computing resources on a short-term basis as needed (e.g., processors by the hour and storage by the day) and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful.

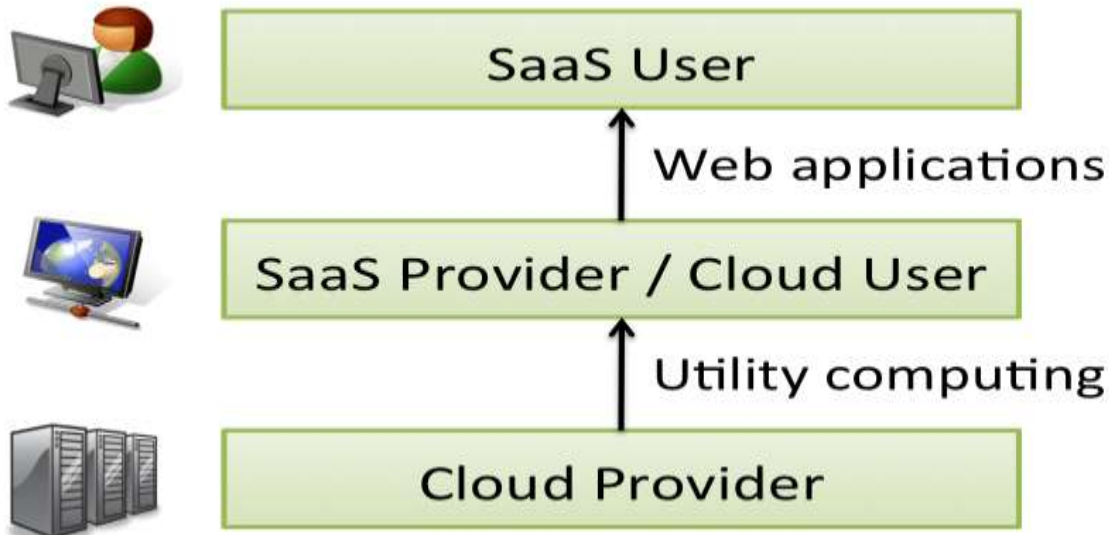


Figure 1: Users and Providers of Cloud Computing.

The benefits of SaaS to both SaaS users and SaaS providers are well documented, so we focus on Cloud Computing effects on Cloud Providers and SaaS Providers/Cloud users. The top level can be recursive, in that SaaS providers can also be a SaaS users. For example, a mishap provider of rental maps might be a user of the Craigslist and Google maps services.

We will argue that all three are important to the technical and economic changes made possible by Cloud Computing. Indeed, past efforts at utility computing failed, and we note that in each case one or two of these three critical characteristics were missing. For example, Intel Computing Services in 2000-2001 required negotiating a contract and longer-term use than per hour. As a successful example, Elastic Compute Cloud (EC2) from Amazon Web Services (AWS) sells 1.0-GHz x86 ISA "slices" for 10 cents per hour, and a new "slice", or instance, can be added in 2 to 5 minutes. Amazon's Scalable Storage Service (S3) charges \$0.12 to \$0.15 per gigabyte-month, with additional bandwidth charges of \$0.10 to \$0.15 per gigabyte to move data in to and out of AWS over the Internet. Amazon's bet is that by statistically multiplexing multiple instances onto a single physical box, that box can be simultaneously rented to many customers who will not in general interfere with each others' usage (see Section 7). While the attraction to Cloud Computing users (SaaS providers) is clear, who would become a Cloud Computing provider, and why? To begin with, realizing the economies of scale afforded by statistical multiplexing and bulk purchasing requires the construction of extremely large datacenters. Building, provisioning, and launching such a facility is a hundred-million-dollar undertaking. However, because of the phenomenal growth of Web services through the early 2000's, many large Internet companies, including Amazon,

eBay, Google, Microsoft and others, were already doing so. Equally important, these companies also had to develop scalable software infrastructure (such as MapReduce, the Google File System, BigTable, and Dynamo [16, 20, 14, 17]) and the operational expertise to armor their datacenters against potential physical and electronic attacks. Therefore, a necessary but not sufficient condition for a company to become a Cloud Computing provider is that it must have existing investments not only in very large datacenters, but also in large-scale software infrastructure and operational expertise required to run them. Given these conditions, a variety of factors might influence these companies to become Cloud Computing providers:

1. Make a lot of money. Although 10 cents per server-hour seems low, Table 2 summarizes James Hamilton's estimates [23] that very large datacenters (tens of thousands of computers) can purchase hardware, network bandwidth, and power for 1=5 to 1=7 the prices offered to a medium-sized (hundreds or thousands of computers) datacenter. Further, the fixed costs of software development and deployment can be amortized over many more machines. Others estimate the price advantage as a factor of 3 to 5 [37, 10]. Thus, a sufficiently large company

could leverage these economies of scale to offer a service well below the costs of a medium-sized company and still make a tidy profit

- .2. Leverage existing investment. Adding Cloud Computing services on top of existing infrastructure provides a new revenue stream at (ideally) low incremental cost, helping to amortize the large investments of datacenters. Indeed, according to Werner Vogels, Amazon's CTO, many Amazon Web Services technologies were initially developed for Amazon's internal operations [32]

- .3. Defend a franchise. As conventional server and enterprise applications embrace Cloud Computing, vendors with an established franchise in those applications would be motivated to provide a cloud option of their own. For example, Microsoft Azure provides an immediate path for migrating existing customers of Microsoft enterprise applications to a cloud environment

.References

- [1] Cloudera, Hadoop training and support [online]. Available from: <http://www.cloudera.com/>.
- [2] TC3 Health Case Study: Amazon Web Services [online]. Available from: <http://aws.amazon.com/solutions/case-studies/tc3-health/>
- [3] Washington Post Case Study: Amazon Web Services [online]. Available from: <http://aws.amazon.com/solutions/case-studies/washington-post/>.
- [4] Amazon.com CEO Jeff Bezos on Animoto [online]. April 2008. Available from: <http://blog.animoto.com/2008/04/21/amazon-ceo-jeff-bezos-on-animoto/>
- [5] Black Friday traffic takes down Sears.com. Associated Press (November 2008).
- [6] ABRAMSON, D., BUYYA, R., AND GIDDY, J. A computational economy for grid computing and its implementation in the Nimrod-G resource broker. *Future Generation Computer Systems* 18, 8 (2002), 1061–1074.
- [7] ADMINISTRATION, E. I. State Electricity Prices, 2006 [online]. Available from: <http://www.eia.doe.gov/ncic/rankings/stateelectricityprice.htm>.
- [8] AMAZON AWS. Public Data Sets on AWS [online]. 2008. Available from: <http://aws.amazon.com/publicdatasets/>.
- [9] BARROSO, L. A., AND HOLZLE, U. The Case for Energy-Proportional Computing. *IEEE Computer* 40, 12 (December 2007)
- [10] BECHTOLSHEIM, A. Cloud Computing and Cloud Networking. talk at UC Berkeley, December 2008.
- [11] BIALECKI, A., CAFARELLA, M., CUTTING, D., AND O'MALLEY, O. Hadoop: a framework for running applications on large clusters built of commodity hardware. Wiki at <http://lucene.apache.org/hadoop>.
- [12] BRODKIN, J. Loss of customer data spurs closure of online storage service 'The Linkup'. *Network World* (August 2008).
- [13] CARR, N. Rough Type [online]. 2008. Available from: <http://www.roughtype.com>.
- [14] CHANG, F., DEAN, J., GHEMAWAT, S., HSIEH, W., WALLACH, D., BURROWS, M., CHANDRA, T., FIKES, A., AND GRUBER, R. Bigtable: A distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)* (2006).
- [15] CHENG, D. PaaS-onomics: A CIO's Guide to using Platform-as-a-Service to Lower Costs of Application Initiatives While Improving the Business Value of IT. Tech. rep., LongJump, 2008.
- [16] DEAN, J., AND GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. In *OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation (Berkeley, CA, USA, 2004)*, USENIX Association, pp. 10–10.
- [17] DECANDIA, G., HASTORUN, D., JAMPANI, M., KAKULAPATI, G., LAKSHMAN, A., PILCHIN, A., SIVASUBRAMANIAN, S., VOSSHALL, P., AND VOGELS, W. Dynamo: Amazon's highly available key-value store. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles (2007)*, ACM Press New York, NY, USA, pp. 205–220.
- [18] DEMERS, A. J., PETERSEN, K., SPREITZER, M. J., TERRY, D. B., THEIMER, M. M., AND WELCH, B. B. The bayou architecture: Support for data sharing among mobile users. In *Proceedings IEEE Workshop on Mobile Computing Systems & Applications (Santa Cruz, California, August-September 1994)*, pp. 2–7.
- [19] GARFINKEL, S. An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS . Tech. Rep. TR-08-07, Harvard niversity, August 2007
- [20] GHEMAWAT, S., GOBIOFF, H., AND LEUNG, S.-T. The google file system. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles (New York, NY, USA, 2003)*, ACM, pp. 29–43. Available from: http://portal.acm.org/ft_gateway.cfm?id=945450&type=pdf&coll=Portal&dl=GUIDE&CFID=19219697&CFTOKEN=50259492
- [21] GRAY, J. Distributed Computing Economics. *Queue* 6, 3 (2008), 63–68. Available from http://portal.acm.org/ft_gateway.cfm?id=1394131&type=digital%20edition&coll=Portal&dl=GUIDE&CFID=19219697&CFTOKEN=50259492
- [22] GRAY, J., AND PATTERSON, D. A conversation with Jim Gray. *ACM Queue* 1, 4 (2003), 8–17.
- [23] HAMILTON, J. Cost of Power in Large-Scale Data Centers [online]. November 2008. Available
- [24] HAMILTON, J. Internet-Scale Service Efficiency. In *Large-Scale Distributed Systems and Middleware (LADIS) Workshop (September 2008)*.

- [25] HAMILTON, J. Perspectives [online]. 2008. Available from: <http://perspectives.mvdirona.com>.
- [26] HAMILTON, J. Cooperative Expendable Micro-Slice Servers (CEMS): Low Cost, Low Power Servers for Internet-Scale Services. In Conference on Innovative Data Systems Research (CIDR '09) (January 2009)
- [27] H"O LZLE, U. Private communication, January 2009.
- [28] HOSANAGAR, K., KRISHNAN, R., SMITH, M., AND CHUANG, J. Optimal pricing of content delivery network (CDN) services. In The 37th Annual Hawaii International Conference on System Sciences (2004), pp. 205–214.
- [29] JACKSON, T. We feel your pain, and we're sorry [online]. August 2008. Available from: <http://gmailblog.blogspot.com/2008/08/we-feel-your-pain-and-were-sorry.html>.
- [30] KISTLER, J. J., AND SATYANARAYANAN, M. Disconnected operation in the coda file system. In Thirteenth ACM Symposium on Operating Systems Principles (Asilomar Conference Center, Pacific Grove, U.S., 1991), vol. 25, ACM Press, pp. 213–225.
- [31] KREBS, B. Amazon: Hey Spammers, Get Off My Cloud! Washington Post (July 2008).
- [32] MCCALPIN, J. Memory bandwidth and machine balance in current high performance computers. IEEE Technical Committee on Computer Architecture Newsletter (1995), 19–25.
- [33] MCKEOWN, N., ANDERSON, T., BALAKRISHNAN, H., PARULKAR, G., PETERSON, L., REXFORD, J., SHENKER, S., , AND TURNER, J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review 38, 2 (April 2008).
- [34] NURMI, D., WOLSKI, R., GRZEGORCZYK, C., OBERTELLI, G., SOMAN, S., YOUSEFF, L., AND ZAGORODNOV, D. Eucalyptus: A Technical Report on an Elastic Utility Computing Archietcture Linking Your Programs to Useful Systems . Tech. Rep. 2008-10, University of California, Santa Barbara, October 2008.
- [35] PARKHILL, D. The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc., US, 1966.
- [36] PAXSON, V. private communication, December 2008.
- [37] RANGAN, K. The Cloud Wars: \$100+ billion at stake. Tech. rep., Merrill Lynch, May 2008.
- [38] SIEGELE, L. Let It Rise: A Special Report on Corporate IT. The Economist (October 2008)
- [39] STERN, A. Update From Amazon Regarding Friday's S3 Downtime. CenterNetworks (February 2008). Available from: <http://www.centernetworks.com/amazon-s3-downtime-update>.
- [40] STUER, G., VANMECHELEN, K., AND BROECKHOVE, J. A commodity market algorithm for pricing substitutable Grid resources. Future Generation Computer Systems 23, 5 (2007), 688–701.