The Potential of Cloud Computing: Opportunities and Challenges

Armando Fox, University of California, Berkeley

Abstract

Cloud computing is not a flash in the pan: it has the potential to be as influential and transformative as the microprocessor. It opens genuinely new opportunities for more agile innovation, faster and higher quality research, and agile entrepreneurship, and exposes direct incentives and opportunities for more energy-efficient computing. While early success stories are inspiring, much work remains before the democratizing power of cloud computing is truly within the reach of every innovator, educator and researcher.

Introduction

> One of the overall design goals is to create a computing system which is capable of meeting almost all of the present and near-future requirements of a large computer utility. Such systems must run continuously and reliably 7 days a week, 24 hours a day in a way similar to telephone or power systems, and must be capable of meeting wide service demands ... [T]he importance of a multiple access system operated as a computer utility is that it allows a vast enlargement of the scope of computer-based activities, which should in turn stimulate a corresponding enrichment of many areas of our society.

Computer science moves so quickly and is so focused on its recent history that when we pause for reflection, we are often surprised at visionary ideas articulated long before technology would admit their practical implementation. The above vision of “utility computing” from 1965 comes from an overview of the pioneering and highly influential MULTICS computing system. [9]

More than forty years later, the vision appears close to becoming reality. In 2008, Amazon announced the availability of its Elastic Compute Cloud (EC2): Anyone with a credit card could use the servers in Amazon’s datacenters for 10 cents per server-hour with no minimum or maximum purchase and no contract.[1] As soon as one releases a server back into the pool, one stops being charged for it.

This is the essence of cloud computing: datacenter hardware and software available in a pay-as-you-go manner to the general public, with any given user enjoying the illusion of virtually infinite capacity available instantaneously on demand. Hence the term *utility computing* to describe the “product” being sold by a Cloud Computing provider. The applications running on this computing hardware, such as social networking or email, have long been referred to as *Software as a Service* or SaaS. Of course, by 2008 many companies were already operating their own “private clouds” in order to deliver such services, such as Google Search and eBay auctions. But EC2 represented the first truly low-cost offering of utility computing *unbundled* from any particular SaaS application, thereby providing a new deployment substrate. Unsurprisingly, SaaS providers are among Cloud Computing’s most enthusiastic customers, and other cloud computing providers with SaaS expertise, such as Google and Microsoft, now have their own public cloud offerings.

Skeptical observers were hard pressed to believe that Amazon could operate such a service at a profit, but as James Hamilton observed [14], the costs of bandwidth, storage and power for “warehouse scale” datacenters are 5 to 7 times cheaper per unit than for medium-sized datacenters, due to the buying power that comes with economy of scale (see Table 1). Indeed, a principal reason cloud computing did not emerge earlier is that the need to build warehouse-scale datacenters and develop the programming and operations expertise to run them is relatively recent, driven largely by the growth of consumer-facing Internet services. Amazon, with its retail-to-consumer operational expertise, had identified a profitable way to pass these savings along to individual users.

The cloud computing service model represents a radical departure from conventional IT. As an analogy, consider semiconductor chip design. At one time, leading hardware companies had to build and operate their own fabrication facilities, but the high cost of doing so (building such a facility costs over US$3B today) meant that only a handful

[1] Amazon has since added additional options and services, and reduced the base price to 8.5 cents per server-hour.

Table 1: Economies of scale in 2006 for medium-sized datacenter (≈1000 servers) vs. warehouse-scale datacenter (≈50,000 servers). [14]

Technology	Medium-sized DC	Warehouse-Scale DC	Ratio
Network	$95 per Mbit/sec/month	$13 per Mbit/sec/month	7.1
Storage	$2.20 per GByte/month	$0.40 per GByte/month	5.7
Administration	1 administrator per ≈140 Servers	1 administrator for >1000 Servers	7.1

of companies with extremely high sales volumes could justify it. Today, in contrast, semiconductor foundries that build chips for others, such as Taiwan Semiconductor Manufacturing Company (TSMC), enable "fab-less" companies such as nVIDIA to focus on innovative chip design without committing the capital, operational expenses, and risks associated with owning a state-of-the-art fabrication line. 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.

Cost Associativity and Elasticity

But Cloud Computing is not just about lower costs: the instantaneous and fine-grained availability of capacity on demand enables *fundamentally new* kinds of computation that were previously infeasible by exposing two new behaviors to users. In 2008, the National Archives released 17,481 pages of documents including First Lady Hillary Clinton's daily activities. Peter Harkins, a senior engineer at The Washington Post, immediately harnessed 200 computers in EC2 to produce a searchable corpus of these documents within nine hours, and made it publicly available on the World Wide Web less than a day later [1]. The server time cost less than $150. The ability to use 200 servers for nine hours for the same price as using one server for 1800 hours—an unprecedented new capability in IT—can be called *cost associativity*. Before cloud computing, the nearest approximation to this ability was usage-based pricing for network bandwidth, in which large-volume network users were charged for actual amount of data transferred independently over the time period during which the usage occurred.

That same year, programmers at Animoto developed an application to create music videos from a user's photo collection. When that application was made available to the over 200 million users of the Facebook social networking site [2], its popularity increased so quickly that the number of users doubled every 12 hours for three days, causing the application to grow from 50 servers to 3,500. After the peak subsided, demand fell to a level well below peak, and the unnecessary servers were released. This ability to both add and remove servers in minutes rather than days or weeks—again unprecedented in IT—is called *elasticity*.

We believe the real benefit of cloud computing's elasticity is *transference of risk.* As an example, real world estimates of steady-state server utilization in datacenters range from 5% to 20% [17, 19] because for many services the peak workload exceeds the average by factors of 2 to 10. With a conventional datacenter, the only option is to provision enough servers to handle the peak, thus wasting capacity and money at nonpeak times. The waste is even worse if the estimate of the peak is optimistic, since there will then be unused resources even at peak time. Conversely, if peak estimates are pessimistic and the datacenter is underprovisioned, some users will receive poor service and may abandon the site for good. Cloud computing can avoid all these scenarios because actual usage can closely track demand by harnessing and releasing servers in the pool as needed. Succinctly, the *risk* of making a poor provisioning decision is transferred from the service operator to the Cloud Computing provider.

Elasticity is even more important in handling *spikes* spikes and "data hotspots" resulting from unexpected events. During the terrorist attacks of September 11, 2001, viewer traffic to the CNN website increased by an order of magnitude in just 15 minutes [16]. When entertainer Michael Jackson died unexpectedly in 2009, the number of Web searches about Jackson spiked so suddenly, to nearly 10 times its average, that Google initially mistook the event for a malicious attack on its search service. The ability to deal with sudden surges is particularly relevant for mobile applications that *"respond in real time to information provided either by their users or by nonhuman sensors"* (Tim O'Reilly) [19]. Elasticity will be key for such services since they are now accessible to the more than 50% of the world's population equipped with the most ubiquitous Internet access device of all, the cell phone.

[2]In 2008. As of this writing, Facebook claims over 500 million users.

Opportunity & Challenge: Scaling Down

Before cloud computing, scaling up usually meant buying and installing new hardware, so it was considered a permanent change. Consequently, there was extensive research on scaling up without taking systems offline. But the idea of subsequently scaling down—and then possibly back up again later—just wasn't part of the landscape. Furthermore, since cloud computing involves borrowing machines from a shared pool that is constantly being upgraded, scale-up and scale-down will likely involve greater hardware heterogeneity than would be present in a conventional datacenter. Research on software that can gracefully scale down as well as up on short timescales is just beginning; SCADS (Scalable Consistency-Adjustable Data Storage) is one example [7].

At the other extreme, fine-grained pricing may enable even cheaper utility computing during demand troughs. California power companies have already introduced demand-based pricing models in which power is discounted during off-peak times. By analogy, Amazon EC2 has introduced a new mechanism whereby otherwise unused machines are made available at a discounted rate on a "best effort" basis: you may be forced to give up the machine on short notice if a priority customer willing to pay a premium needs the machine during a period of higher demand. This again leads to the relatively new situation of a cluster whose topology and size can change at any time, and whose cycles may be "reclaimed" on short notice for higher-priority applications; research on scheduling frameworks such as Mesos [15] is addressing how applications deployed on cloud computing applications can deal gracefully with such challenges.

Opportunity & Challenge: Greener computing

In traditional research proposals, energy costs are usually absorbed into general institutional overhead. With cloud computing, a customer using fewer machines is consuming less energy and paying less money. Although warehouse-scale datacenters are already being built in locations where cheaper power (e.g. hydroelectric) is available, cloud computing's pay-as-you-go model introduces a direct financial motivation for the cloud users to reduce their own energy usage.

Set against this opportunity for "greener" IT are several challenges. One is that servers should consume energy proportional to their activity; Barroso and others refer to this idea of "doing nothing well" as *energy-proportional* computing [8]. Unfortunately, today's servers consume nearly half as much energy when idle as when fully utilized. Design improvements at all levels, from the power supply to to energy-aware software, will be necessary to improve this situation.

A second challenge is that SaaS applications in particular are bound by stringent service-level agreements (SLAs) on application responsiveness. For example, a typical SLA might require that over any given 5-minute interval, 99.9% of all user requests are answered within 100ms. If we try to save energy by releasing unused servers, we must be prepared to reinstate those servers when demand rises again, and any delay in re-provisioning the servers could cause SLA violations. Since the financial penalties for violations can be severe, SaaS operators have been reluctant to pursue this strategy in the absence of accurate models relating performance to energy consumption and effective policies for releasing unused servers that are sufficiently agile to reinstate them quickly enough when needed.

Opportunity & Challenge: Better and Faster Research

Cost associativity means that "embarrassingly parallel" experiments—those requiring many trials or tasks that can be pursued independently—can be accelerated as far as available cloud resources will allow. An experiment that requires 100,000 trials taking 1 minute each would take over two months to complete on a single server, but cost associativity makes it possible to instead harness 1,000 cloud servers for two hours for the same cost. RAD Lab researchers working on datacenter-scale computing now routinely run experiments involving hundreds of servers to test out their ideas at realistic scales—something previously impossible for any university.

Tools like Google's MapReduce [10] (and its open-source equivalent Hadoop) give programmers a familiar data-parallel "building block" and encapsulate the complex software engineering needed to deal with the elasticity and heterogeneity of the cloud environment. But many problems cannot be easily expressed as MapReduce tasks, so other frameworks like Pig, Hive and Cascading have emerged to provide higher-level languages and abstractions for cloud programming. The Berkeley BOOM framework (boom.cs.berkeley.edu) aims to simplify the creation of new abstractions by applying principles from declarative networking. Progress on all these fronts is promising and vigorous, and some of these systems are in regular use in production environments, but the artifacts and the ecosystem comprising them are still a long way from "turnkey" systems that will allow domain-expert programmers to seamlessly combine these abstractions in their applications.

Table 2: Price of kilowatt-hours of electricity by region [3].

Cents per kWh	Where	Possible Reasons Why
3.6	Idaho	Hydroelectric; not transmitted long distance
10.0	California	long-distance transmission; limited transmission lines in Bay Area; no coal-fired electricity allowed in California
18.0	Hawaii	Must ship fuel to generate electricity

More recently, the scientific and high-performance computing (HPC) community has become more interested in cloud computing. Compared to SaaS workloads that rely on request-level parallelism, HPC workloads rely on thread- or task-level parallelism and are therefore more communication-intensive and more sensitive to communication latency. HPC workloads are also more sensitive to memory bandwidth and inter-machine I/O, the latter of which is particularly vulnerable to the "performance noise" artifacts introduced by the pervasive use of virtualization in cloud environments [6]. They often rely on resource-scheduling approaches such as Gang Scheduling and assumptions about network topology connecting the servers, both of which make sense in a statically provisioned environment but not for cloud computing. Indeed, cloud providers try to obfuscate physical network topology, replacing it with virtualized attributes such as "availability zones," and the CPU/memory ratios of their larges available compute instances were until recently not cost-effective for many traditional HPC applications. Unsurprisingly, early benchmarks of existing HPC applications on public clouds did not give particularly encouraging results [11, 20].

However, the cloud providers are quickly responding to the potential HPC market, as illustrated by Amazon's recent introduction of "Cluster Compute Instances" in July 2010. These can be paid for on an hourly basis and consist of eight or more servers more appropriately provisioned for HPC: specific microprocessor architecture details are available to allow highly-tuned code, instances feature more RAM (32GB) than conventional cloud servers, and instances in an HPC cluster are placed on the same local subnet to minimize communication latency. Experiments conducted at the National Energey Research Scientific Computing lab (NERSC) at Lawrence Berkeley Laboratory measured an 8.5X performance improvement of several HPC codes when using this new instance type compared to conventional EC2 instances. Cloud computing installations operated by academic/industrial consortia, such as the Google/IBM/NSF CluE cluster running Hadoop [2], Yahoo's M45 cluster (http://labs.yahoo.com/Cloud_Computing), and OpenCirrus (opencirrus.org) are additional examples of cloud computing for scientific research.

So while both application and system software will need to continue developing and adapting to the new architectural constraints of cloud computing, cloud hardware vendors seem to be responding quickly to the needs of this emerging cloud computing user base. Indeed, as Ian Foster observes [12], even if the running time of a problem is slower on EC2 than a dedicated supercomputer, the total time-to-answer may be faster with cloud computing because traditional HPC facilites operate on a batch basis to maximize utilization of the equipment. This can lead to jobs waiting in long queues before they can be run, whereas a "virtual cluster" in the cloud can be provisioned in minutes and deprovisioned when no longer needed.

Longtime HPC veteran Dan Reed, now head of the Cloud Computing Futures research unit at Microsoft Research, also believes cloud computing is a "game changer" for HPC [18]. He points out that the challenge of designing cloud infrastructure has much in common with the challenge of designing HPC systems: both call for an integrated view of design, including processors, network fabric, storage architecture, and even power and cooling. Yet the cloud infrastructure market, because of its scale, can influence hardware design in a way that traditional HPC has been unable to do. While commodity clouds may not be able to replace very-high-end specialized supercomputers, a tremendous amount of science that is now done on desktops and small clusters is likely to benefit greatly from cloud computing.

Opportunity & Challenge: Big Data

Space limitations prevent a full exploration of all the challenges and opportunities of cloud computing in this abstract. We refer the reader to [6] for a more thorough discussion of additional challenges, including issues of regulatory compliance and inelastic software licensing models. However, of the "top 10" challenges we identify in that article, a highly relevant one for science and engineering research is the "gravity well" effect of big data. According to Wikipedia, the Large Hadron Collider could generate up to 15 petabytes (15×10^{15} bytes) of data per year, and researchers in astronomy, biology, and many other fields routinely deal with multi-terabyte datasets. A boon of cloud computing is the ability to pair tremendous amounts of computation on-demand with these large datasets; indeed, part of Amazon's motivation to host large public datasets for free [4] may be to attract users to purchase Cloud Computing cycles near this data.

The key word is "near". Transferring 10 terabytes from UC Berkeley to Amazon in Seattle, Washington, would take more than 45 days at 20 megabits/sec, a typical speed observed by Garfinkel in his measurements of long-haul bandwidth in and out of Amazon's S3 cloud storage service [13]. Current prices of $100 to $150 per TB transferred also make it expensive. There are therefore clear technical and financial implications around the relative placement of data and compute cycles for users whose work involves such large datasets.

In our abovementioned overview of cloud computing we proposed a service whereby big-data users could ship crates of hard drives overnight to a cloud provider, who would then physically incorporate these directly into the cloud infrastructure. This idea was motivated by the observations of the late Jim Gray, who had experience with this method and reported only 1 disk failure in about 400 attempts (and even this could have been mitigated with existing RAID-like techniques). Shortly after our paper was published, Amazon began offering such a service and continues to do so. As network cost-performance is improving more slowly than any other cloud computing technology (see Table 3), the FedEx disk option for large data transfers will get more attractive each year.

Table 3: We update Gray's costs of computing resources from 2003 to 2008, normalize to what $1 could buy in 2003 vs. 2008, and compare to the cost of paying per use of $1 worth of resources on AWS at 2008 prices.

	WAN bandwidth/mo.	**CPU hours (all cores)**	**disk storage**
Item in 2003	1 Mbps WAN link	2 GHz CPU, 2 GB DRAM	200 GB disk, 50 Mb/s transfer rate
Cost in 2003	$100/mo.	$2000	$200
$1 buys in 2003	1 GB	8 CPU hours	1 GB
Item in 2008	100 Mbps WAN link	2 GHz, 2 sockets, 4 cores/socket, 4 GB DRAM	1 TB disk, 115 MB/s sustained transfer
Cost in 2008	$3600/mo.	$1000	$100
$1 buys in 2008	2.7 GB	128 CPU hours	10 GB
cost/performance improvement	2.7x	16x	10x
Cost to rent $1 worth on AWS in 2008	**$0.27–$0.40** ($0.10–$0.15/GB × 3 GB)	**$2.56** (128× 2 VM's@$0.10 each)	**$1.20–$1.50** ($0.12–$0.15/GB-month × 10 GB)

Related to the "gravity well" effect of data is the possibility of cloud provider lock-in. Amazon's EC2 represents one end of a spectrum in that the utility computing service being sold consists of a "bare bones" server built around the Intel x86 processor architecture; cloud users must provide all the software themselves, with open source building blocks such as the Linux operating system being very popular starting points. Other offerings, however, such as Google AppEngine and Microsoft Azure, will provide proprietary value-added software functionality such as more sophisticated storage options, automatic capacity scaling for certain kinds of applications, or better facilities for batch job management. To the extent that a user's deployed software comes to rely on such proprietary features, it may be difficult for the user to migrate to a different cloud provider for technical or business reasons. One way to mitigate provider lock-in is to actively standardize the application programming interfaces to different cloud services. Providers could then differentiate their offerings through better implementations, but migration would result only in a possible loss of performance and not functionality. The Data Liberation Front, started by a group of Google engineers, is actively pursuing this effort.

Conclusion

The main technical obstacle holding back cloud computing remains the immaturity of software tools and infrastructure: When it comes to programming the cloud, we're still using the equivalent of assembly language. It's worth recalling that in 1995, researchers at Berkeley and elsewhere were arguing that networks of commodity workstations (NOWs) offered many potential advantages over high-performance symmetric multiprocessors [5], including better scalability, higher cost-effectiveness, and the potential for high availability through inexpensive redundancy. Although contemporary software lacked the ability to deal with important aspects of the NOW architecture such as the possibility of partial failure, the economic and technical arguments for NOW seemed so compelling that over the course of several years, academic research and commercial and open-source software developed tools and infrastructure for programming this idiosyncratic architecture at a much higher level of abstraction. As a result of their success, applications that once took engineer-years to develop and deploy on a NOW can today be prototyped by Berkeley undergraduates as an eight-week course project. Given this rapid evolution, there is reason to be optimistic that in a few years, computer-based scientific and engineering experiments that take weeks today will instead yield results in hours, and that the days of having to purchase and administer one's own computer cluster (and then wait in line to use it) will seem as archaic as text-only interfaces seem today.

Acknowledgments

This work was done at the UC Berkeley Reliable Adaptive Distributed Systems Laboratory (RAD Lab) and much of what is reported here builds on the original survey article written with Michael Armbrust, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Support was provided by Sun Microsystems, Google, Microsoft, Amazon Web Services, Cisco Systems, Cloudera, eBay, Facebook, Fujitsu, Hewlett-Packard, Intel, Network Appliance, SAP, VMWare and Yahoo! Inc., and matching funds from the State of California 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), the National Science Foundation (grant CNS-0509559), and the University of California Industry/University Cooperative Research Program (UC Discovery) grant COM07-10240.

References

[1] Washington Post Case Study: Amazon Web Services. http://aws.amazon.com/solutions/case-studies/washington-post.

[2] National science foundation awards millions to fourteen universities for cloud computing research, April 2009. http://www.nsf.gov/news/news_summ.jsp?cntn_id=114686.

[3] Energy Information Administration. State Electricity Prices, 2006. http://www.eia.doe.gov/neic/rankings/ stateelectricityprice.htm.

[4] Amazon AWS. Public Data Sets on AWS, 2008. http://aws.amazon.com/publicdatasets/.

[5] Thomas E. Anderson, David E. Culler, and David A. Patterson et al. The case for NOW (networks of workstations). *IEEE Micro*, 15(1):54–64, Feb 1995.

[6] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. *Communications of the ACM*, 53(4):50–59, Apr 2010.

[7] Michael Armbrust, Armando Fox, David A. Patterson, Nick Lanham, Beth Trushkowsky, Jesse Trutna, and Haruki Oh. SCADS: Scale-independent storage for social computing applications. In *CIDR 2009*, 2009.

[8] Luiz André Barroso and Urs Hölzle. The case for energy-proportional computing. *IEEE Computer*, December 2007.

[9] Fernando J. Corbató and Victor A. Vyssotsky. Introduction and overview of the multics system. In *AFIPS Conference Proceedings*, 1965. Available at http://www.lilli.com/fjcc1.html.

[10] Jeff Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *4th USENIX Syposium on Operating Systems Design and Implementation (OSDI '04)*, San Diego, CA, December 2004.

[11] Constantinos Evangelinos and Chris N. Hill. Cloud computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on amazon's ec2. In *First ACM Workshop on Cloud Computing and its Applications (CCA'08)*, Chicago, IL, October 2008.

[12] Ian Foster. What's faster, a supercomputer or EC2? http://ianfoster.typepad.com/blog/2009/08/whats-fastera-supercomputer-or-ec2.html.

[13] S Garfinkel. An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS . Technical Report TR-08-07, Harvard University, August 2007.

[14] James Hamilton. Internet-Scale Service Efficiency. In *Large-Scale Distributed Systems and Middleware (LADIS) Workshop*, September 2008.

[15] Benjamin Hindman, Andrew Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy H. Katz, Scott Shenker, and Ion Stoica. Mesos: A platform for fine-grained resource sharing in the data center. Technical Report UCB-EECS-2010-87, University of California, Berkeley, May 2010.

[16] William LeFebvre. CNN.com – facing a world crisis. Invited talk at USENIX Conference on Large Installation System Administration (LISA 2001), Dec 2001.

[17] Kash Rangan. The Cloud Wars: $100+ billion at stake. Technical report, Merrill Lynch, May 2008.

[18] Dan Reed. HPC and cloud computing: twins separated at birth. HPCWire interview with Dr. Dan Reed of Microsoft Cloud Computing Futures research team.

[19] Ludwig Siegele. Let It Rise: A Special Report on Corporate IT. *The Economist*, October 2008.

[20] Edward Walker. Benchmarking amazon EC2 for high performance scientific computing. *;login*, 33(5), Oct 2008.