The when, how, and why of enterprise cloud computing

THE Cloud AT Your Service

Jothy Rosenberg
Arthur Mateos

FOREWORD BY ANNE THOMAS MANES

SAMPLE
CHAPTER

MANNING

The Cloud at Your Service
by Jothy Rosenberg
Arthur Mateos

Chapter 1

Copyright 2011 Manning Publications

brief contents

1 What is cloud computing?

This chapter covers

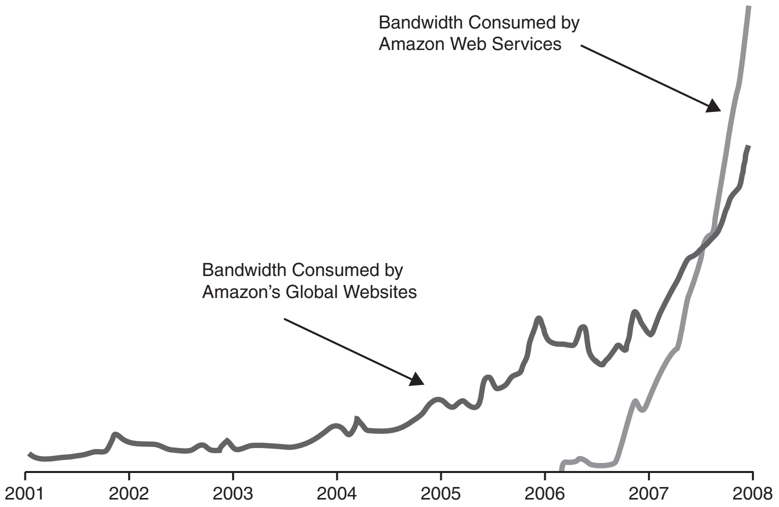
- Defining the five main principles of cloud computing
- Benefiting from moving to the cloud
- How evolving IT led to cloud computing
- Discussing the different layers (types) of clouds

Cloud computing is the hottest buzzword in the IT world right now. Let's understand why this is and what this cloud computing hype is all about. A growing consensus among cloud vendors, analysts, and users defines cloud computing at the highest level as computing services offered by a third party, available for use when needed, that can be scaled dynamically in response to changing needs. Cloud computing represents a departure from the norm of developing, operating, and managing IT systems. From the economic perspective, not only does adoption of cloud computing have the potential of providing enormous economic benefit, but it also provides much greater flexibility and agility. We'll continue to refine and expand our definition of cloud computing as well as your understanding of its costs and benefits throughout this book.

Not only are IT journals and IT conferences writing and talking about cloud computing, but even mainstream business magazines and the mass media are caught up in its storm. It may win the prize for the most over-hyped concept IT has ever had. Other terms in this over-hyped category include Service-Oriented Architectures SOA (SOA), application service providers, and artificial intelligence, to name a few. Because this book is about cloud computing, we need to define it at a much more detailed level. You need to fully understand its pros and cons, and when it makes sense to adopt it, all of which we'll explain in this chapter. We hope to cut through the hype; and to do that we won't merely repeat what you've been hearing but will instead give you a framework to understand what the concept is all about and why it really is important.

You may wonder what is driving this cloud hype. And it would be easy to blame analysts and other prognosticators trying to promote their services, or vendors trying to play up their capabilities to demonstrate their thought leadership in the market, or authors trying to sell new books. But that would ignore a good deal of what is legitimately fueling the cloud mania. All of the great expectations for it are based on the facts on the ground.

--> Software developers around the world are beginning to use cloud services. In the first 18 months that it was open for use, the first public cloud offering from Amazon attracted over 500,000 customers. This isn't hype; these are facts. As figure 1.1 from Amazon's website shows, the bandwidth consumed by the company's cloud has quickly eclipsed that used by their online store. As the old adage goes, "where there's smoke, there must be a fire," and clearly something is driving the rapid uptake in usage from a cold start in mid-2006.



--> **Figure 1.1 Amazon originally deployed a large IT infrastructure to support its global e-commerce platform. In less than 18 months after making the platform available as a cloud service to external users, its usage, as measured by amount of bandwidth consumed, outstripped bandwidth used internally.**

Similar to the previous technology shifts—such as the move from mainframes to client-server, and then from client-server to the internet—cloud computing will have major implications on the business of IT. We hope to provide you with the background and perspective to understand how it can be effectively used as a component of your overall IT portfolio.

We'll begin by expanding on our earlier definition of cloud computing in terms of its five main principles.

1.1 Five main principles that define cloud computing

We can summarize the five main principles of cloud computing as follows:

1 ▪ Pooled computing resources available to any subscribing users
2 ▪ Virtualized computing resources to maximize hardware utilization
3 ▪ Elastic scaling up or down according to need
4 ▪ Automated creation of new virtual machines or deletion of existing ones
5 ▪ Resource usage billed only as used

We assert, with very few notable exceptions called out later, that these five main principles are necessary components to call something *cloud computing*. They're summarized in table 1.1 with a brief explanation of each one for quick reference.

Table 1.1 The five main principles of cloud computing

Resource	Explanation
1 Pooled resources	Available to any subscribing users
2 Virtualization	High utilization of hardware assets
3 Elasticity	Dynamic scale without CAPEX
4 Automation	Build, deploy, configure, provision, and move, all without manual intervention
5 Metered billing	Per-usage business model; pay only for what you use

We'll now discuss these principles in concrete terms, making sure you understand what each one means and why it's a pillar of cloud computing.

1.1.1 Pooled computing resources

--> The first characteristic of cloud computing is that it utilizes pooled computing assets that may be externally purchased and controlled or may instead be internal resources that are pooled and not dedicated. We further qualify these pooled computing resources as contributing to a cloud if these resources are available to any subscribing users. This means that *anyone* with a credit card can subscribe.

If we consider a corporate website example, three basic operational deployment options are commonly employed today. The first option is the self-hosting option. Here, companies choose not to run their own data center and instead have a third party lease them a server that the third party manages. Usually, managed hosting services lease corporate clients a dedicated server that isn't shared (but shared hosting is common as well). On this single principle, cloud computing acts like a *shared managed hosting service* because the cloud provider is a third party that owns and manages the physical computing resources which are shared with other users, but there the similarity ends.

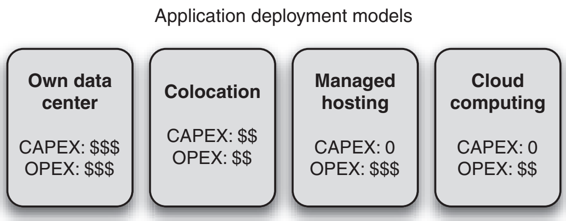
--> Independent of cloud computing, a shift from self-hosted IT to outsourced IT resources has been underway for years. This has important economic implications. The two primary implications are a shift of capital expenses (CAPEX) to operational expenses (OPEX), and the potential reduction in OPEX associated with operating the infrastructure. The shift from CAPEX to OPEX means a lowering of the financial barrier for the initiation of a new project. (See the definition in section 3.1.)

-->In the self-hosted model, companies have to allocate a budget to be spent up front for the purchase of hardware and software licenses. This is a fixed cost regardless of whether the project is successful. In an outsourced model (managed hosting), the startup fees are typically equivalent to one month's operational cost, and you must commit to one year of costs up front. Typically, the one-year cost is roughly the same or slightly lower than the CAPEX cost for an equivalent project, but this is offset by the reduced OPEX required to operate the infrastructure. In sharp contrast, in a cloud model, there are typically no initial startup fees. In fact, you can sign up, authorize a credit card, and start using cloud services literally in less time than it would take to read this chapter. Figure 1.2 showcases side by side the various application deployment models with their respective CAPEX and OPEX sizes.

--> The drastic difference in economics that you see between the hosting models and the cloud is due to the fact that the cost structures for cloud infrastructures are vastly better than those found in other models. The reasons for the economies of scale are severalfold, but the primary drivers are related to the simple economics of volume. Walmart and Costco can buy consumer goods at a price point much lower than you or I could because of their bulk purchases. In the world of computing, the "goods" are computing, storage, power, and network capacity.

1.1.2 Virtualization of compute resources

The second of the five main principles of cloud computing has to do with virtualization of compute resources. Virtualization is nothing new. Most enterprises have been shifting much of their physical compute infrastructure to virtualized for the past 5 to 10 years. Virtualization is vital to the cloud because the



<--

Figure 1.2 IT organizations have several alternatives for hosting applications. The choice of deployment model has different implications for the amount of CAPEX (up-front capital expenditure) and OPEX (ongoing operational costs). The number of $ signs represent the relative level of CAPEX and OPEX involved with the choice of deployment model.

scale of cloud infrastructures has to be enormous, based on thousands of servers. Each server takes up physical space and uses significant power and cooling. Getting high utilization out of each and every server is vital to be cost effective.

!!! --> The recent technological breakthrough that enabled high utilization on commodity hardware—and which is the single biggest factor behind the cloud being a recent IT phenomenon—is virtualization where each physical server is partitioned into many virtual servers. Each one acts like a real server that can run an operating system and a full complement of applications.[1] Virtualized servers are the primary units that can be consumed as needed in the cloud. These virtualized servers constitute a large pool of resources available when required. But having such a large pool will work only if applications can use more or less of the pool as demands placed on the applications grow and shrink. As you'll see in chapter 4, the notion of a private cloud softens this first principal but keeps all the others.

1.1.3 Elasticity as resource demands grow and shrink

The fact that this large pool of resources exists enables a concept known as *elasticity*—the third of our five main principles. Elasticity is such a key concept in cloud computing that Amazon decided to name its cloud Amazon Elastic Compute Cloud.

--> Elasticity—a synonym for *dynamic scaling*—refers to the ability to dynamically change how much resource is consumed in response to how much is needed. Typical applications require a base level of resources under normal, steady-state conditions, but need more resource under peak load conditions.

In a non-cloud world, you would have to build sufficient capacity to not only perform adequately under baseline load conditions, but also handle peak load scenarios with sufficiently good performance. In the case of a self-hosted model, this means over-provisioning the amount of hardware for a given allocation. In the case of a managed hosting deployment, you can start with a small set of resources and grow as the requirements of the application grow. But provisioning for a new set of dedicated hardware resources takes weeks or, in many larger organizations, months. Having thousands of virtualized resources that can be harnessed and released in correlation to application demand would be useless if such allocation and freeing required manual intervention.

1.1.4 Automation of new resource deployment

--> The ability to automatically (via an API) provision and deploy a new virtual instance of a machine, and, equivalently, to be able to free or de-provision an instance, is our fourth principle of cloud computing. A cloud-deployed application can provision new instances on an as-needed basis, and these resources are brought online within minutes. After the peak demand ebbs, and you don't need the additional resources, these

[1] The rapid shift to multicore servers only strengthens the impact of virtualization. Each virtual machine with its operating system and full complement of applications can run on its own core simultaneously with all other virtual machines on the same physical server.

virtual instances can be taken offline and de-provisioned, and you will no longer be billed. Your incremental cost is only for the hours that those additional instances were in use and active.

1.1.5 Metered billing that charges only for what you use

--> The fifth distinguishing characteristic of cloud computing is a metered billing model. In the case of managed hosting, as we mentioned before, there typically is an initial startup fee and an annual contract fee. The cloud model breaks that economic barrier because it's a pay-as-you-go model. There is no annual contract and no commitment for a specific level of consumption.

Typically, you can allocate resources as needed and pay for them on an hourly basis. This economic advantage benefits not only projects being run by IT organizations, but also innumerable entrepreneurs starting new businesses. Instead of needing to raise capital as they might have in the past, they can utilize vast quantities of compute resources for pennies per hour. For them, the cloud has drastically changed the playing field and allowed the little guy to be on equal footing with the largest corporations.

1.2 Benefits that can be garnered from moving to the cloud

"I'll never buy another server again," said the Director of IT for a medium-sized Software-as-a-Service (SaaS) company, only partially in jest, after recently completing the deployment of a new corporate website for his organization. This website (a PHP-based application with a MySQL backend) showcased the corporate brand and the primary online lead-generation capability for the company's business.

Before the overhaul, it was run from a redundant pair of web servers hosted by one of the leading managed-hosting service providers at a total cost of roughly $2,200/month. The company replaced the infrastructure for the original website with a cloud implementation consisting of a pair of virtual server instances running for roughly $250/month—almost a 90 percent savings! Its quality of service (QoS) team monitored the performance and availability of the website before and after the change and saw no measureable difference in the service quality delivered to end users. Buoyed by the success with this initial project, this organization is looking at all future initiatives for the possibility of deployment within the cloud, including a software-build system and offsite backup.

1.2.1 Economic benefits of the change from capital to operational expenses

--> As we said when discussing the five main principles of cloud computing, the fundamental economic benefit that cloud computing brings to the table is related to the magical conversion of CAPEX to OPEX. A pay-as-you-go model for resource use reshapes the fundamental cost structure of building and operating applications. The initial barrier to starting a project is drastically reduced; and until there is dramatic uptake in the use of an application that has been developed, the costs for running it remain low.

The good news is that this isn't the only cost advantage. By harnessing the cloud, you can also take advantage of cloud providers' economic leverage because of the volume at which they can purchase hardware, power, and bandwidth resources.

In many cases, the economic benefits discussed here will pan out—but as you'll see later, there are always exceptions. For some situations and applications, it makes better economic sense not to use cloud computing. It isn't a panacea.

1.2.2 Agility benefits from not having to procure and provision servers

In addition to lowering the financial barrier to initiating new projects, the cloud approach improves an organization's agility. It comprehensively reduces the months of planning, purchasing, provisioning, and configuring.

Let's take as an example a performance-testing project launching a new consumer-facing website. In the old world, there were two ways to solve this problem, depending on your timeframes and budget. The first involved purchasing a software license for a load-testing tool like HP Mercury LoadRunner and purchasing the requisite servers to run the load-testing software. At that point, you were ready to script your tests and run your test plan. Alternatively, you could hire an outside consulting company that specialized in performance testing and have it run the tests for you. Both were time-consuming exercises, depending on how long it took to negotiate either the licensing agreement for the software or the consulting agreement with the outside firm.

Fast-forward to the new world of cloud computing. You have two new faster and more flexible ways of accomplishing the same task: use an open-source load-testing application installed on cloud instances, and use the cloud's virtual machines to perform the load test (on as many servers as you need). The time required to set up and begin applying load to a system is under half an hour. This includes signing up for an account, as the Python open source load-testing tool called Pylot demonstrates (see http://coreygoldberg.blogspot.com/2009/02/pylot-web-load-testing-from-amazon.html).

If you're looking for a more packaged approach, you can use one of the SaaS offerings that uses the cloud to generate traffic. They can automatically run tests in a coordinated fashion across multiple instances running from multiple cloud operators, all in an on-demand fashion. In either of these scenarios, the time to result is a matter of hours or days, generating time, not to mention cost efficiencies. We'll explore more about cloud-based testing in chapter 7.

1.2.3 Efficiency benefits that may lead to competitive advantages

Adopting cloud technologies presents many opportunities to those who are able to capitalize on them. As we have discussed, there are potential economic as well as time-to-market advantages in using the technology. As organizations adopt cloud computing, they will realize efficiencies that organizations that are slower to move won't realize, putting them at an advantage competitively.

1.2.4 Security stronger and better in the cloud

--> Surprised by the heading? Don't be: it's true. As you're aware, corporate buildings no longer have electrical generators (which they used to) because we leave electricity generation to the experts. If corporations have their own data centers, they have to develop standard security operating procedures. But it's not their core business to run a secure data center. They can and will make mistakes. A lot of mistakes. The total annual fraud and security breach tab is $1 trillion, according to cybersecurity research firm Poneman (www.nationalcybersecurity.com).

But first, as always, you must weigh the potential benefits against the potential costs. You must take into account other factors, such as reliability and performance, before making the leap into the clouds. In future chapters, we'll address these issues; but suffice it to say we believe that after you understand them and take the proper measures, they can be managed. This done, you'll be able to realize the full benefits of moving to the cloud.

In the next section, we'll look at the evolution of technology that enabled cloud computing. This short detour into history is important because you can learn from previous platform shifts to understand what is similar and what is different this time. That in turn can help you make informed decisions about your shift to this new evolution of IT—the cloud.

1.3 Evolution of IT leading to cloud computing

Cloud computing didn't sprout fully formed from the technology ether in 2005. Its technological underpinnings developed over the course of the last 40 or so years. The technological process was evolutionary, across several disparate areas. But these advances, aggregated into a bundle, represent a revolutionary change in the way IT will be conducted in the future.

Gillett and Kapor made the first known reference to cloud computing in 1996 in an MIT paper (http://ccs.mit.edu/papers/CCSWP197/CCSWP197.html). Today's common understanding of cloud computing retains the original intent. It was a mere decade later when a real-world instantiation of the cloud came into existence as Amazon repurposed its latent e-commerce resources and went into the business of providing cloud services. From there, it was only a matter of a few months until the term became commonplace in our collective consciousness and, as figure 1.3 shows, in our Google search requests (they're the same thing in today's world, right?).

1.3.1 Origin of the "cloud" metaphor

One common question people ask is, "Where did the term *cloud* come from?" The answer is that for over a decade, whenever people drew pictures of application architectures that involved the internet, they inevitably represented the internet with a cloud, as shown in figure 1.4.

The cloud in the diagram is meant to convey that anonymous people are sitting at browsers accessing the internet, and somehow their browser visits a site and begins to

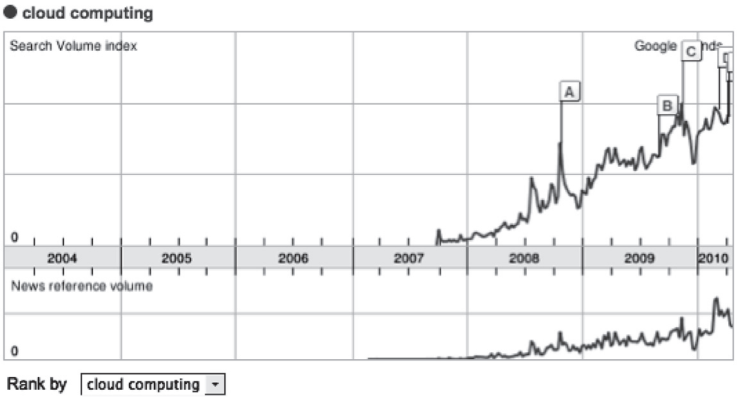


Figure 1.3 Cloud computing as a concept entered our collective consciousness in mid-2007. This figure shows the rapid rise in popularity of the search term *cloud computing* as measured by Google. The labels correspond to major cloud announcements. A: Microsoft announces it will rent cloud computing space; B: *Philadelphia Inquirer* reports, "Microsoft's cloud computing system grow is growing up"; C: *Winnipeg Free Press* reports, "Google looks to be cloud-computing rainmaker." Source: Google Trends (www.google.com/trends), on the term *cloud computing*.

access its infrastructure and applications. From "somewhere out there" you get visitors who can become users who may buy products or services from you. Unlike internal customers to whom you may provide IT applications and services, this constituency exists "somewhere else," outside of your firewall, and hence outside of your domain of control. The image of a cloud is merely a way to represent this vast potential base of anonymous users coming from the internet.

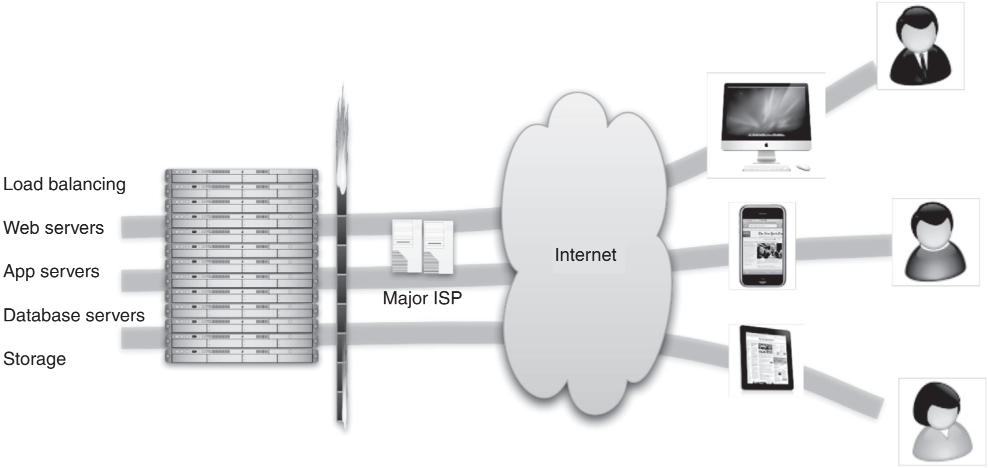


Figure 1.4 A picture of a cloud is a ubiquitous representation of the internet and is used almost universally in discussions or drawings of computer architecture.

Those users must log in from a PC to access the internet. Technically, each one needs an Internet Service Provider (ISP) that may be a telecom company, their employer, or a dedicated internet access company (such as AOL). Each ISP needs a bank of machines that people can access and that in turn has access to the internet.

--> Simply put, the earliest concept of the cloud consisted of large aggregations of computers with access to the internet, accessed by people through their browsers. The concept has remained surprisingly true to that early vision but has evolved and matured in important ways. We'll explore those ways in detail in this book.

1.3.2 Major computing paradigm shifts: mainframes to client-server to web

--> In the 1960s, we saw the development of the first commercial mainframes. In the beginning, these were single-user systems, but they evolved in the 1970s to systems that were time-shared. In this model, the large computing resource was *virtualized*, and a virtual machine was allocated to individual users who were sharing the system (but to each, it seemed that they had an entire dedicated machine).

--> Virtual instances were accessed in a thin-client model by green-screen terminals. This mode of access can be seen as a direct analog of the concept of virtualized instances in the cloud, although then a single machine was divided among users. In the cloud, it's potentially many thousands of machines. The scarcity of the computing resource in the past drove the virtualization of that resource so that it could be shared, whereas now, the desire to fully utilize physical compute resources is driving cloud virtualization.

--.> As we evolved and entered the client-server era, the primacy of the mainframe as the computing center of the universe dissolved. As computing power increased, work gradually shifted away from centralized computing resources toward increasingly powerful distributed systems. In the era of the PC-based desktop applications, this shift was nearly complete: computing resources for many everyday computing tasks moved to the desktop and became thick client applications (such as Microsoft Office). The mainframe retained its primacy only for corporate or department-wide applications, relegating it to this role alone.

--> The standardization of networking technology simplified the ability to connect systems as TCP/IP became the protocol of the burgeoning internet in the 1980s. The ascendancy of the web and HTTP in the late 1990s swung the pendulum back to a world where the thin-client model reigned supreme. The world was now positioned to move into the era of *cloud computing* . The biggest stages of the evolution of IT are diagrammed vertically in a timeline in figure 1.5.

--> The computing evolution we are still in the midst of has had many stages. Platform shifts like mainframe to client-server and then client-server to web were one dimension of the evolution. One that may be less apparent but that is having as profound an impact is the evolution of the data center and how physical computing resources are housed, powered, maintained, and upgraded.

Virtualization -->

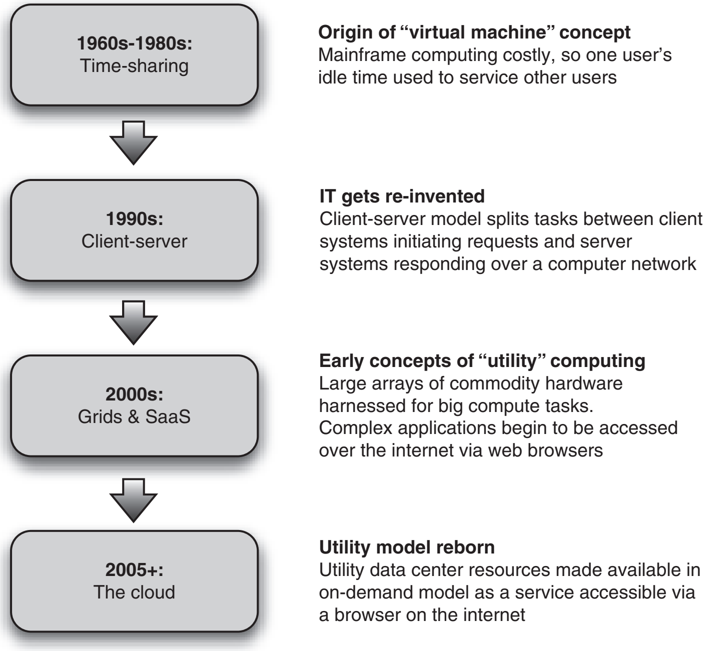

1960s-1980s:
Time-sharing
Origin of "virtual machine" concept
Mainframe computing costly, so one user's idle time used to service other users
1990s:
Client-server
IT gets re-invented
Client-server model splits tasks between client systems initiating requests and server systems responding over a computer network
2000s:
Grids & SaaS
Early concepts of "utility" computing
Large arrays of commodity hardware harnessed for big compute tasks.
Complex applications begin to be accessed over the internet via web browsers
2005+:
The cloud
Utility model reborn
Utility data center resources made available in on-demand model as a service accessible via a browser on the internet

Figure 1.5 Cloud computing is best understood as an evolutionary change. The key elements and concepts of cloud computing emerged gradually over several decades through the various predominant computing paradigms.

1.3.3 *Housing of physical computing resources: data center evolution*

--> Over the past four decades, there have been tremendous changes in hardware capabilities, specifically in computing power and storage. The ability to quickly process prodigious amounts of data on inexpensive and mass-produced commodity servers means that a few inexpensive racks of servers can handle problems that were tackled on NSA-sized budgets as recently as the early 1990s.

One measure of the progress in computational power is the cost in Floating Point Operations Per Second, or FLOPS. FLOPS are simple mathematical operations (such as addition, multiplication, and division) that can be performed in a single operation by a computer. Comparing the number of operations that two computers can perform in one second allows for a rough measure of their computational strength. In 1976, the state-of-the-art Cray-1 was capable of delivering roughly 150 million FLOPS (megaFLOPS) at the price point of $5 million, or over $33,000/MegaFLOPS. A typical quad-core-processor-based PC today can be purchased for under $1,000 and can perform 50 GigaFLOPS (billion FLOPS), which comes out to about $0.02/MegaFLOPS.

Similarly, the cost of storage has decreased dramatically over the last few decades as the capacity to store data has kept pace with the ability to produce terabytes of digital content in the form of high-definition HD video and high-resolution imagery. In the early 1980s, disk space costs exceeded $200/MB; today, this cost has come down to --> under $0.01/MB.

Network technologies have advanced as well, with modern bandwidth rates in the 100–1000 Gbps range commonplace in data centers today. As for WAN, the turn of the millennium saw a massive build-out of dark fiber, bringing high-speed broadband to most urban areas. More rural areas have satellite coverage, and on-the-go, high-speed wireless networks mean almost ubiquitous broadband connectivity to the grid.

--> To support the cloud, a huge data-center build-out is now underway. Google, Microsoft, Yahoo!, Expedia, Amazon, and others are deploying massive data centers. These are the engine rooms that power the cloud, and they now account for more than 1.2 percent of the U.S.'s total electricity usage (including cooling and auxiliaries),[2] which doubled over the period from 2000 to 2005. We'll present the economies of scale and much more detail about how these mega data centers are shaping up in chapter 2.

SOA Service Oriented Architechture

1.3.4 *Software componentization and remote access: SOA, virtualization, and SaaS*

--> On the software side of the cloud evolution are three important threads of development: virtualization, SOA, and SaaS. Two of these are technological, and the third relates to the business model.

--> The first important thread is virtualization. As discussed previously, virtualization isn't a new concept, and it existed in mainframe environments. The new innovation that took place in the late 1990s was the extension of this idea to commodity hardware. Virtualization as pioneered by VMware and others took advantage of the capacity of modern multicore CPUs and made it possible to partition and time-slice the operation of commodity servers. Large server farms based on these commodity servers were partitioned for use across large populations of users.

--> SOA is the second software concept necessary for cloud computing. We see SOA as the logical extension of browser-based standardization applied to machine-to-machine communication. Things that humans did through browsers that interacted with a web server are now done machine-to-machine using the same web-based standard protocols and are called *SOA*. SOA makes practical the componentization and composition of services into applications, and hence it can serve as the architectural model for building composite applications running on multiple virtualized instances.

--> The final software evolution we consider most pertinent to the cloud is SaaS. Instead of being a technological innovation, this is a business model innovation. Historically, enterprise software was sold predominantly in a perpetual license model. In this model, a customer purchased the right to use a certain software application in perpetuity for a fixed, and in many cases high, price. In subsequent years, they paid for support and maintenance at typically around 18 percent of the original price. This entitled the

[2] Jonathan G. Koomey, Ph.D. (www.koomey.com), Lawrence Berkeley National Laboratory & Stanford University.

customer to upgrades of the software and help when they ran into difficulty. In the SaaS model, you don't purchase the software—you rent it. Typically, the fee scales with the amount of use, so the value derived from the software is proportional to the amount spent on it. The customer buys access to the software for a specified term, which may be days, weeks, months, or years, and can elect to stop paying when they no longer need the SaaS offering. Cloud computing service providers have adopted this *pay-as-you-go* or *on-demand* model.

This brings up an important point we need to consider next. SaaS is one flavor or layer in a stack of cloud types. A common mistake people make in these early days of the cloud is to make an apples-to-oranges comparison of one type of cloud to another. To avoid that, the next section will classify the different layers in the cloud stack and how they compare and contrast.

1.4 Classifying cloud layers: different types for different uses

First, let's learn a little more about how SaaS evolved and established itself, to set the context for discussing the other classes of clouds.

In the earliest days of commercially practicable computing, computer resources were scarce, and the primary model for their use was much like a utility. But this was different from the sense of utility that cloud computing offers today; it was more akin to the community well in a village during a drought. Members of the community had access to and were allocated a fixed amount of water. In the case of cloud computing today, we've returned to the notion of computing being available as a utility, but without the scarcity.

The cloud movement was presaged by the shift in business model toward SaaS that took over the software industry at the turn of the century. Before it was called SaaS, it was an application rented from an Application Service Provider (ASP); here, the traditional enterprise license model was turned on its head, and you purchased in a pay-as-you-go manner, with costs scaling with usage instead of having a large up-front capital investment. You didn't need to provision hardware and software; instead, the services were turned on when needed. After this approach was renamed SaaS, it evolved into several new kinds of offerings that we'll explore next.

We can classify cloud computing several ways. In this book, we present a taxonomy where cloud services are described generically as "X as a Service," where X can take on values such as Hardware, Infrastructure, Platform, Framework, Application, and even Datacenter. Vendors aren't in agreement about what these designations mean, nor are they consistent in describing themselves as belonging to these categories. Despite this, we'll reproduce one interesting hierarchy that illustrates the use of these terms, with representative vendors (some at this point only historical) populating the diagram in figure 1.6.

A more simplified representation of the cloud types shown in figure 1.7 highlights important aspects and key characteristics of different kinds of cloud offerings.

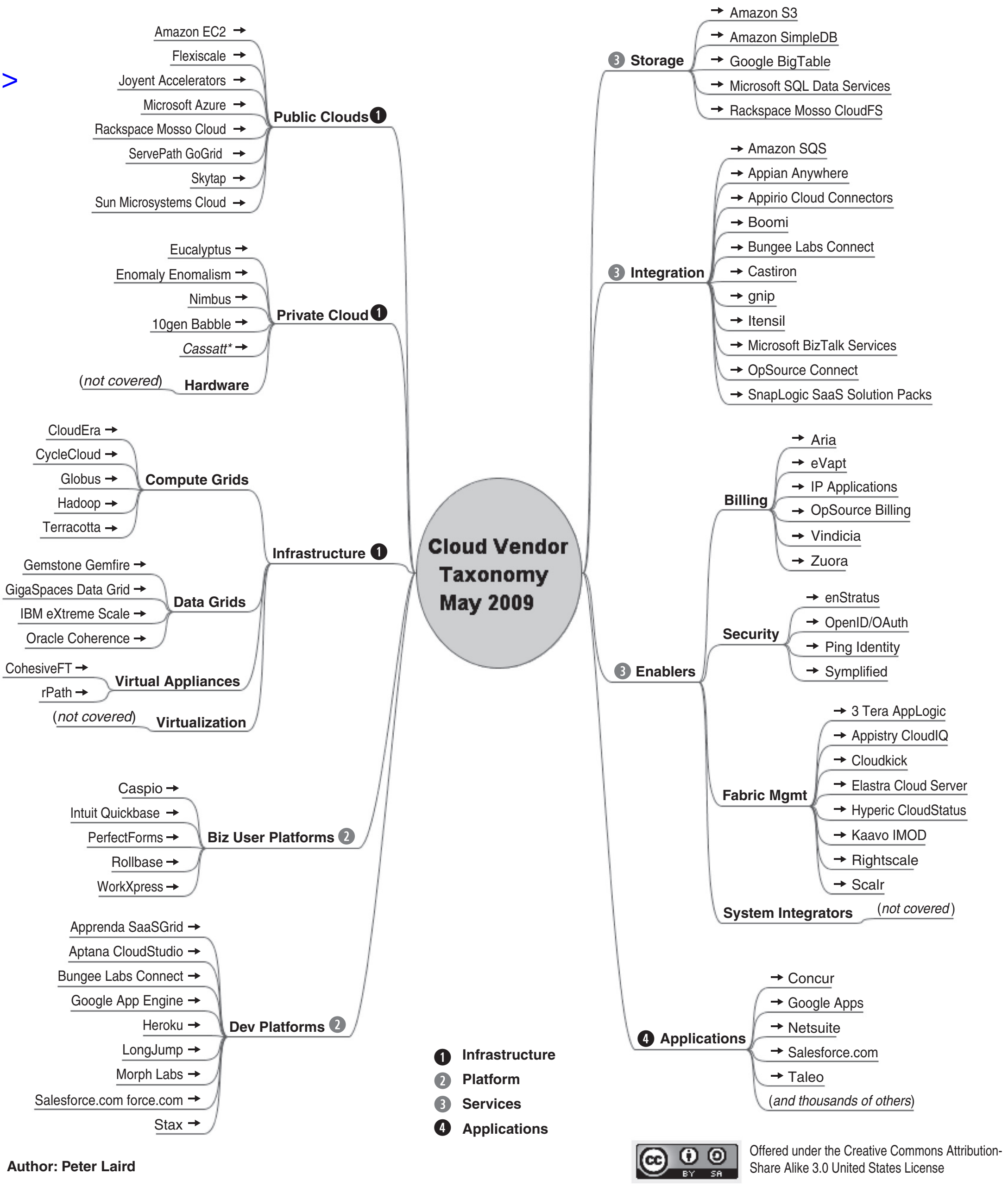


Figure 1.6 Cloud technologies are evolving as various vendors attempt to provide services populating the cloud ecosystem. These services run the gamut from the hardware systems used to build cloud infrastructure to integration services and cloud-based applications. Source: Peter Laird, http://peterlaird.blogspot.com.

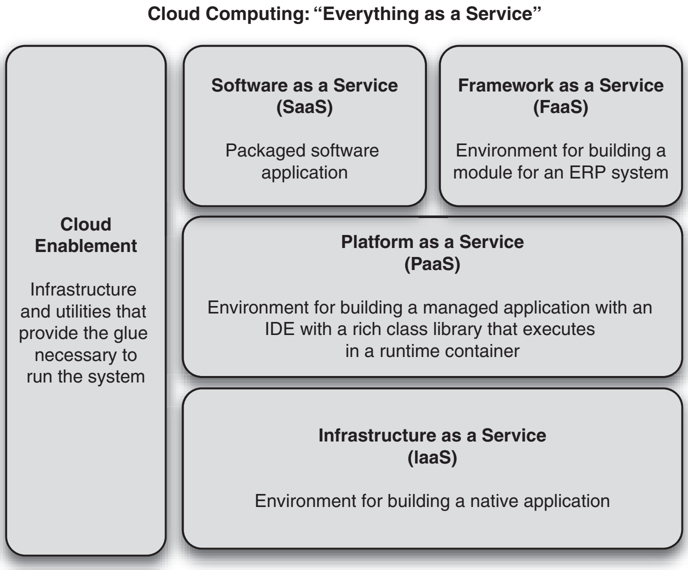

Cloud Computing: "Everything as a Service"
Cloud Enablement
Infrastructure and utilities that provide the glue necessary to run the system
Software as a Service (SaaS)
Packaged software application
Framework as a Service (FaaS)
Environment for building a module for an ERP system
Platform as a Service (PaaS)
Environment for building a managed application with an IDE with a rich class library that executes in a runtime container
Infrastructure as a Service (IaaS)
Environment for building a native application

Figure 1.7 In the X-as-a-Service taxonomy, cloud services are classified by the level of prepackaging offered to the consumer of the specific service. An IaaS provides computing capabilities in the rawest form and hence offers the greatest flexibility. At the highest layers, there is less flexibility but also less complexity to be managed.

What does XaaS mean generically? It means on demand, requiring little or no capital expenditure. It means consumable remotely and across any mode of access over the internet, and in a metered billing model. Let's now go through the boxes representing the different classes of clouds in figure 1.7. First up is IaaS.

1.4.1 Infrastructure as a Service (IaaS)

The lowest level of XaaS is known as IaaS, or sometimes as Hardware as a Service (HaaS). A good example of IaaS is the Amazon Elastic Compute Cloud (EC2).

A user of IaaS is operating at the lowest level of granularity available and with the least amount of prepackaged functionality. An IaaS provider supplies virtual machine images of different operating system flavors. These images can be tailored by the developer to run any custom or packaged application. These applications can run natively on the chosen OS and can be saved for a particular purpose. The user can bring online and use instances of these virtual machine images when needed. Use of these images is typically metered and charged in hour-long increments.

Storage and bandwidth are also consumable commodities in an IaaS environment, with storage typically charged per gigabyte per month and bandwidth charged for transit into and out of the system.

IaaS provides great flexibility and control over the cloud resources being consumed, but typically more work is required of the developer to operate effectively in the environment. In chapter 2, we'll delve into IaaS and see how it works in greater detail.

1.4.2 Platform as a Service (PaaS)

PaaS's fundamental billing quantities are somewhat similar to those of IaaS: consumption of CPU, bandwidth, and storage operates under similar models. Examples of PaaS include Google AppEngine and Microsoft Azure. The main difference is that PaaS requires less interaction with the bare metal of the system. You don't need to directly interact with or administer the virtual OSs. Instead, you can let the platform abstract away that interaction and concentrate specifically on writing the application. This simplification generally comes at the cost of less flexibility and the requirement to code in the specific languages supported by the particular PaaS provider.

1.4.3 Software as a Service (SaaS) and Framework as a Service (FaaS)

SaaS, as described earlier in the chapter, refers to services and applications that are available on an on-demand basis. Salesforce.com is an example. FaaS is an environment adjunct to a SaaS offering and allows developers to extend the prebuilt functionality of the SaaS applications. Force.com is an example of a FaaS that extends the Salesforce.com SaaS offering.

FaaS offerings are useful specifically for augmenting and enhancing the capabilities of the base SaaS system. You can use FaaS for creating either custom, specialized applications for a specific organization, or general-purpose applications that can be made available to any customer of the SaaS offering. Like a PaaS environment, a developer in a FaaS environment can only use the specific languages and APIs provided by the FaaS.

1.4.4 Private clouds as precursors of public clouds

--> In addition to the classifications we discussed earlier, we should introduce some important concepts relative to the different classifications of clouds. *Private clouds* are a variant of generic cloud computing where internal data-center resources of an enterprise or organization aren't made available to the general public—that is, these pooled computing resources are actually not available to *any* subscribing users but are instead controlled by an organization for the benefit of other members of that organization. The public clouds of providers such as Amazon and Google were originally used as private clouds by those companies for other lines of business (book retailing and internet search, respectively).

--> If an organization has sufficient users and enough overall capacity, a private cloud implementation can behave much like a public cloud, albeit on a reduced scale. There has been a tremendous amount of capital investment in data-center resources over the past decade, and one of the important movements is the reorienting of these assets toward cloud-usage models.

--> *Hybrid clouds* combine private and public clouds. You can use them in cases where the capacity of a private cloud is exhausted and excess capacity needs to be provisioned elsewhere.

1.5 Summary

The cloud offers the illusion of infinite resources, available on demand. You no longer need to play the guessing game of how many users need to be supported and how scalable the application is. The cloud takes care of the peaks and troughs of utilization times. In the world of the cloud, you pay for only the resources you use, when you use them. This is the revolutionary change: the ability to handle scale without paying a premium. In this realm of true utility computing, resource utilization mirrors the way we consume electricity or water.

In this chapter, we defined the cloud as computing services that are offered by a third party, are available for use when needed, and can be scaled dynamically in response to changing need. We then touched briefly on the evolution of computing and the developments that led to where we are today. Finally, we looked at a simple cloud classification that should help you understand the various flavors of cloud offerings that are available in the market today and should prevent you from making apples-and-oranges comparisons between incompatible classes of clouds.

As we delve deeper in the next chapter and look at how the cloud works, you'll gain a better understanding of these types of clouds and when it makes sense to use each kind.

ENTERPRISE DEVELOPMENT

THE Cloud AT Your Service

Jothy Rosenberg • Arthur Mateos

Practically unlimited storage, instant scalability, zero-down-time upgrades, low start-up costs, plus pay-only-for-what-you-use without sacrificing security or performance are all benefits of cloud computing. But how do you make it work in your enterprise? What should you move to the cloud? How? And when?

The Cloud at Your Service answers these questions and more. Written for IT pros at all levels, this book finds the sweet spot between rapidly changing details and hand-waving hype. It shows you practical ways to work with current services like Amazon's EC2 and S3. You'll also learn the pros and cons of private clouds, the truth about cloud data security, and how to use the cloud for high scale applications.

What's Inside

- How to build scalable and reliable applications
- The state of the art in technology, vendors, practices
- What to keep in-house and what to offload
- How to migrate existing IT to the cloud
- How to build secure applications and data centers

A PhD in computer science, **Jothy Rosenberg** is a former Duke professor, author of three previous books, and serial entrepreneur involved in the cloud movement from its infancy. A technology entrepreneur with a PhD in nuclear physics from MIT, **Arthur Mateos** has brought to market pioneering SaaS products built on the cloud.

For online access to the authors and a free ebook for owners of this book, go to manning.com/CloudatYourService

Free ebook
SEE INSERT

"Cuts through the complexity to just what's needed."

—From the Foreword by
Anne Thomas Manes

"A definitive source."

—Orhan Alkan
Sun Microsystems

"Approachable coverage of a key emerging technology."

—Chad Davis
Author of *Struts 2 in Action*

"Removes 'cloudiness' from the cloud."

—Shawn Henry
CloudSwitch, Inc.

"Refreshing... without fluff."

—Kunal Mittal
Sony Pictures Entertainment

ISBN 13: 978-1-935182-52-8
ISBN 10: 1-935182-52-8
52999
9 781935 182528

MANNING $29.99 / Can $34.99 [INCLUDING eBOOK]