

Chapter 2

Cloud Computing

Cloud computing technology represents a new paradigm for the provisioning of computing resources. This paradigm shifts the location of resources to the network to reduce the costs associated with the management of hardware and software resources. It represents the long-held dream of envisioning computing as a utility [68] where the economy of scale principles help to effectively drive down the cost of computing resources. Cloud computing simplifies the time-consuming processes of hardware provisioning, hardware purchasing and software deployment. Therefore, it promises a number of advantages for the deployment of data-intensive applications, such as elasticity of resources, pay-per-use cost model, low time to market, and the perception of unlimited resources and infinite scalability. Hence, it becomes possible, at least theoretically, to achieve unlimited throughput by continuously adding computing resources if the workload increases.

To take advantage of cloud-hosted data storage systems, it is important to well understand the different aspects of the cloud computing technology. This chapter provides an overview of cloud computing technology from the perspectives of key definitions (Sect. 2.1), related technologies (Sect. 2.2), service models (Sect. 2.3) and deployment models (Sect. 2.4), followed by Sect. 2.5 which analyzes state-of-the-art of current public cloud computing platforms, with focus on their provisioning capabilities. Section 2.6 summarizes the business benefits for building software applications using cloud computing technologies.

2.1 Definitions

Cloud computing is an emerging trend that leads to the next step of computing evolution, building on decades of research in virtualization, autonomic computing, grid computing, and utility computing, as well as more recent technologies in networking, web, and software services [227]. Although cloud computing is widely accepted nowadays, the definition of cloud computing has been arguable, due to the diversity of technologies composing the overall view of cloud computing.

From the research perspective, many researchers have proposed their definitions of cloud computing by extending the scope of their own research domains. From the view of service-oriented architecture, Dubrovnik [227] implied cloud computing as *“a service-oriented architecture, reduced information technology overhead for the end-user, greater flexibility, reduced total cost of ownership, on-demand services, and many other things”*. Buyya et al. [91] derived the definition from clusters and grids, acclaiming for the importance of service-level agreements (SLAs) between the service provider and customers, describing that cloud computing is *“a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on SLAs”*. Armbrust et al. [68] from Berkeley highlighted three aspects of cloud computing including illusion of infinite computing resources available on demand, no up-front commitment, and pay-per-use utility model, arguing that cloud computing *“consists of the service applications delivered over the Internet along with the data center hardware and systems software that provide those services”*. Moreover, from the industry perspective, more definitions and excerpts by industry experts can be categorized from the perspectives of scalability, elasticity, business models, and others [225].

It is hard to reach a singular agreement upon the definition of cloud computing, because of not only a fair amount of skepticism and confusion caused by various technologies, but also the prevalence of marketing hype. For that reason, National Institute of Standards and Technology has been working on proposing a guideline of cloud computing. The definition of cloud computing in the guideline has received fairly wide acceptance. It is described as [181]:

“a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”

According to this definition, cloud computing has the following essential characteristics:

1. *On-demand self-service*. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
2. *Broad network access*. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
3. *Resource pooling*. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, virtual networks and virtual machines.

- 4. *Rapid elasticity*. Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- 5. *Measured Service*. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

2.2 Related Technologies for Cloud Computing

Cloud computing has evolved out of decades of research in different related technologies from which it has inherited some features and functionalities such as virtualized environments, autonomic computing, grid computing, and utility computing. Figure 2.1 illustrates the evolution towards cloud computing in hosting software applications [214]. In fact, cloud computing is often compared to the following technologies, each of which shares certain aspects with cloud computing. Table 2.1 provides a summary of the feature differences between those technologies

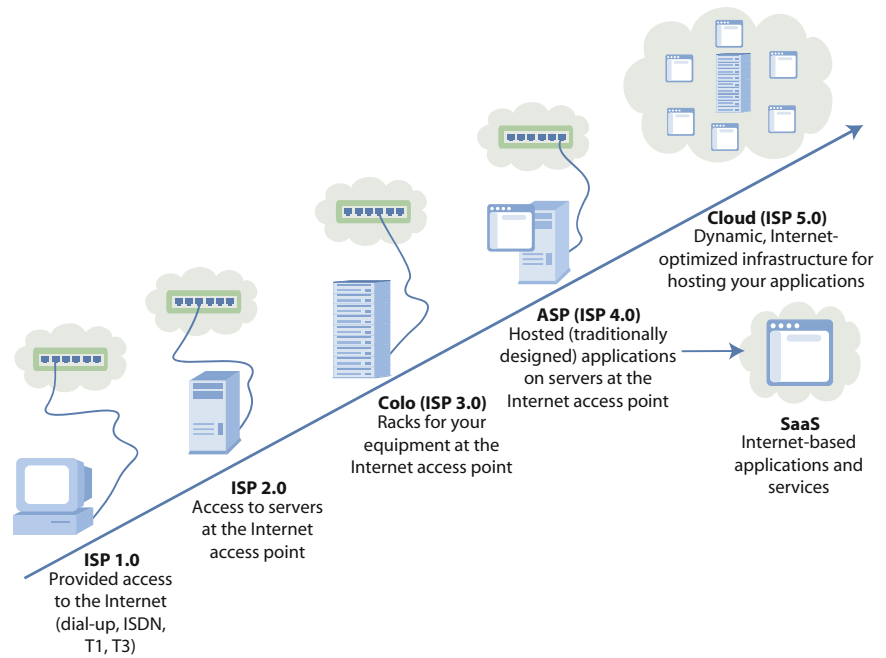


Fig. 2.1 The evolution towards cloud computing in hosting software applications

Table 2.1 Feature similarities and differences between related technologies and cloud computing

Technologies	Differences	Similarities
Virtualization	Cloud computing is not only about virtualizing resources, but also about intelligently allocating resources for managing competing resource demands of the customers.	Both isolate and abstract the low-level resources for high-level applications.
Autonomic computing	The objective of cloud computing is focused on lowering the resource cost rather than to reduce system complexity as it is in autonomic computing.	Both interconnect and integrate distributed computing systems.
Grid computing	Cloud computing however also leverages virtualization to achieve on-demand resource sharing and dynamic resource provisioning.	Both employ distributed resources to achieve application-level objectives.
Utility computing	Cloud computing is a realization of utility computing.	Both offer better economic benefits.

and cloud computing in short, while details of related technologies are discussed as following [239]:

Virtualization

Virtualization is a technology that isolates and abstracts the low-level resources and provides virtualized resources for high-level applications. In the context of hardware virtualization, the details of physical hardware can be abstracted away with support of hypervisors, such as Linux Kernel-based Virtual Machine [33] and Xen [48]. A virtualized server managed by the hypervisor is commonly called a virtual machine. In general, several virtual machines can be abstracted from a single physical machine. With clusters of physical machines, hypervisors are capable of abstracting and pooling resources, as well as dynamically assigning or reassigning resources to virtual machines on-demand. Therefore, virtualization forms the foundation of cloud computing. Since a virtual machine is isolated from both the underlying hardware and other virtual machines. Providers can customize the platform to suit the needs of the customers by either exposing applications running within virtual machines as services, or providing direct access to virtual machines thereby allowing customers to build services with their own applications. Moreover, cloud computing is not only about virtualizing resources, but also about intelligent allocation of resources for managing competing resource demands of the customers. Figure 2.2 illustrates a sample exploitation of virtualization technology in the cloud computing environments [214].

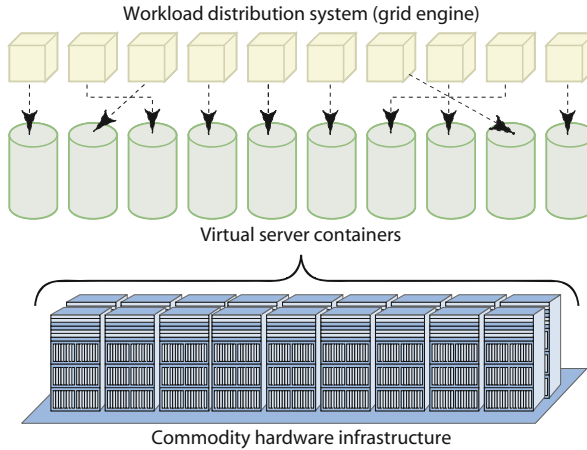


Fig. 2.2 Exploitation of virtualization technology in the architecture of cloud computing

Autonomic computing aims at building computing systems capable of self-management, which means being able to operate under defined general policies and rules without human intervention. The goal of autonomic computing is to overcome the rapidly growing complexity of computer system management, while being able to keep increasing interconnectivity and integration unabated [161]. Although cloud computing exhibits certain similarities to automatic computing the way that it interconnects and integrates distributed data centers across continents, its objective somehow is to lower the resource cost rather than to reduce system complexity.

Grid Computing

Grid computing is a distributed computing paradigm that coordinates networked resources to achieve a common computational objective. The development of grid computing was originally driven by scientific applications which are usually computation-intensive, but applications requiring the transfer and manipulation of a massive quantity of data was also able to take advantage of the grids [142, 143, 171]. Cloud computing appears to be similar to grid computing in the way that it also employs distributed resources to achieve application-level objectives. However, cloud computing takes one step further by leveraging virtualization technologies to achieve on-demand resource sharing and dynamic resource provisioning.

Utility Computing

Utility computing represents the business model of packaging resources as a metered services similar to those provided by traditional public utility companies. In particular, it allows provisioning resources on demand and charging customers based on usage rather than a flat rate. The main benefit of utility computing is better economics. Cloud computing can be perceived as a realization of utility computing. With on-demand resource provisioning and utility-based pricing, customers are able to receive more resources to handle unanticipated peaks and only pay for resources they needed; meanwhile, service providers can maximize resource utilization and minimize their operating costs.

2.3 Cloud Service Models

The categorization of three cloud service models defined in the guideline are also widely accepted nowadays. The three service models are namely *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS), and *Software as a Service* (SaaS).

As shown in Fig. 2.3, the three service models form a stack structure of cloud computing, with Software as a Service on the top, Platform as a Service in the middle, and Infrastructure as a Service at the bottom, respectively. While the inverted triangle shows the possible proportion of providers of each model, it is worth mentioning that definitions of three service models from the guideline paid more attentions to the customers' view. In contrast, Vaquero et al. [225] defined

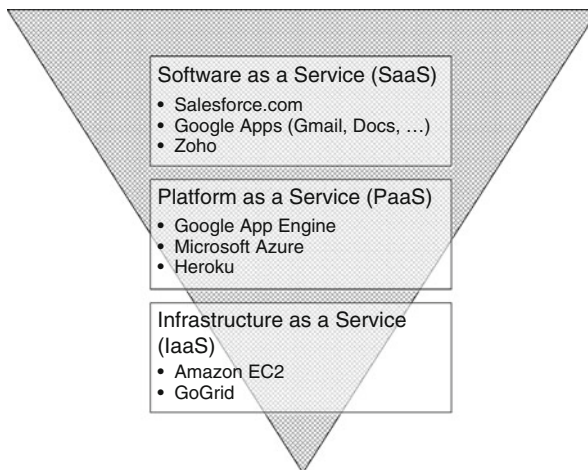


Fig. 2.3 The service models of cloud computing

the three service models from the perspective of the providers' view. The following definitions of the three models combines the two perspectives [181,225], in the hope of showing the whole picture.

1. *Infrastructure as a Service*: Through virtualization, the provider is capable of splitting, assigning, and dynamically resizing the cloud resources including processing, storage, networks, and other fundamental computing resources to build virtualized systems as requested by customers. Therefore, the customer is able to deploy and run arbitrary operating systems and applications. The customer does not need to deploy the underlying cloud infrastructure but has control over which operating systems, storage options, and deployed applications to deploy with possibly limited control of select networking components. The typical providers are Amazon Elastic Compute Cloud (EC2) [4] and GoGrid [17].
2. *Platform as a Service*: The provider offers an additional abstraction level, which is a software platform on which the system runs. The change of the cloud resources including network, servers, operating systems, or storage is made in a transparent manner. The customer does not need to deploy the cloud resources, but has control over the deployed applications and possibly application hosting environment configurations. Three platforms are well-known in this domain, namely Google App Engine [19], Microsoft Windows Azure Platform [37], and Heroku [28] which is a platform built on top of Amazon EC2. The first one offers Python, Java, and Go as programming platforms. The second one supports languages in .NET Framework, Java, PHP, Python, and Node.js. While the third one is compatible with Ruby, Node.js, Clojure, Java, Python, and Scala.
3. *Software as a Service*: The provider provides services of potential interest to a wide variety of customers hosted in its cloud infrastructure. The services are accessible from various client devices through a thin client interface such as a web browser. The customer does not need to manage the cloud resources or even individual application capabilities. The customer could, possibly, be granted limited user-specific application configuration settings. A variety of services, operating as Software as a Service, are available in the Internet, including Salesforce.com [43], Google Apps [21], and Zoho [55].

2.4 Cloud Deployment Models

The guideline also defines four types of cloud deployment models [181], which are described as follows:

1. *Private cloud*. A cloud that is used exclusively by one organization. It may be managed by the organization or a third party and may exist on premise or off premise. A private cloud offers the highest degree of control over performance, reliability and security. However, they are often criticized for being similar to traditional proprietary server farms and do not provide benefits such as no up-front capital costs.

2. *Community cloud*. The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
3. *Public cloud*. The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services (e.g. Amazon, Google, Microsoft). Since customer requirements of cloud services are varying, service providers have to ensure that they can be flexible in their service delivery. Therefore, the quality of the provided services is specified using Service Level Agreement (SLA) which represents a contract between a provider and a consumer that specifies consumer requirements and the provider's commitment to them. Typically an SLA includes items such as uptime, privacy, security and backup procedures. In practice, Public clouds offer several key benefits to service consumers such as: including no initial capital investment on infrastructure and shifting of risks to infrastructure providers. However, public clouds lack fine-grained control over data, network and security settings, which may hamper their effectiveness in many business scenarios.
4. *Hybrid cloud*. The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds). In particular, cloud bursting is a technique used by hybrid clouds to provide additional resources to private clouds on an as-needed basis. If the private cloud has the processing power to handle its workloads, the hybrid cloud is not used. When workloads exceed the private cloud's capacity, the hybrid cloud automatically allocates additional resources to the private cloud. Therefore, Hybrid clouds offer more flexibility than both public and private clouds. Specifically, they provide tighter control and security over application data compared to public clouds, while still facilitating on-demand service expansion and contraction. On the down side, designing a hybrid cloud requires carefully determining the best split between public and private cloud components.

Table 2.2 summarizes the four cloud deployment models in terms of ownership, customership, location, and security.

2.5 Public Cloud Platforms: State-of-the-Art

Key players in public cloud computing domain including Amazon Web Services, Microsoft Windows Azure, Google App Engine, Eucalyptus [16], and GoGrid offer a variety of prepackaged services for monitoring, managing, and provisioning resources. However, the techniques implemented in each of these clouds do vary.

For Amazon EC2, the three Amazon services, namely Amazon Elastic Load Balancer [5], Amazon Auto Scaling [2], and Amazon CloudWatch [3], together expose functionalities which are required for undertaking provisioning of application

Table 2.2 Summary of cloud deployment models

Deployment model	Ownership	Customership	Infrastructure location to customers	Security	Examples
Public cloud	Organization(s)	General public customers	Off-premises	No fine-grained control	Amazon Web Services
Private cloud	An organization/ A third party	Customers within an organization	On/Off-premises	Highest degree of control	Internal cloud platform to support business units in a large organization
Community cloud	Organization(s) in a community/ A third party	Customers from organizations that have shared concerns	On/Off-premises	Shared control among organizations in a community	Healthcare cloud for exchanging health information among organizations
Hybrid cloud	Composition of two or more from above	Composition of two or more from above	On/Off-premises	Tighter control, but require careful split between distinct models	Cloud bursting for load balancing between cloud platforms

services on EC2. The Elastic Load Balancer service automatically provisions incoming application workload across available EC2 instances while the Auto Scaling service can be used to dynamically scale-in or scale-out the number of EC2 instances for handling changes in service demand patterns. Finally the CloudWatch service can be integrated with the above services for strategic decision making based on collected real-time information.

Eucalyptus is an open source cloud computing platform. It is composed of three controllers. Among the controllers, the *cluster controller* is a key component that supports application service provisioning and load balancing. Each cluster controller is hosted on the *head node* of a cluster to interconnect the outer public networks and inner private networks together. By monitoring the state information of instances in the pool of server controllers, the cluster controller can select any available service/server for provisioning incoming requests. However, as compared to Amazon services, Eucalyptus still lacks some of the critical functionalities, such as auto scaling for its built-in provisioner.

Fundamentally, Microsoft Windows Azure *fabric* has a weave-like structure, which is composed of node including servers and load balancers, and edges including power and Ethernet. The *fabric controller* manages a *service node* through a built-in service, named Azure Fabric Controller Agent, running in the background, tracking the state of the server, and reporting these metrics to the controller. If a fault state is reported, the controller can manage a reboot of the server or a migration of services from the current server to other healthy servers. Moreover, the controller also supports service provisioning by matching the VMs that meet required demands.

GoGrid Cloud Hosting offers developers the F5 Load Balancer [18] for distributing application service traffic across servers, as long as IPs and specific ports of these servers are attached. The load balancer provides the round robin algorithm and least connect algorithm for routing application service requests. Additionally, the load balancer is able to detect the occurrence of a server crash, redirecting further requests to other available servers. But currently, GoGrid only gives developers a programmatic set of APIs to implement their custom auto-scaling service.

Unlike other cloud platforms, Google App Engine offers developers a scalable platform in which applications can run, rather than providing direct access to a customized virtual machine. Therefore, access to the underlying operating system is restricted in App Engine where load-balancing strategies, service provisioning, and auto scaling are all automatically managed by the system behind the scenes where the implementation is largely unknown. Chohan et al. [105] have presented initial efforts of building App Engine-like framework, *AppScale*, on top of Amazon EC2 and Eucalyptus. Their offering consists of multiple components that automate deployment, management, scaling, and fault tolerance of an App Engine application. In their design and implementation, a single *AppLoadBalancer* exists in AppScale for distributing initial requests of users to the *AppServers* of App Engine applications. The users initially contact AppLoaderBalancer to request a login to an App Engine application. The AppLoadBalancer then authenticates the login and redirects request to a randomly selected AppServer. Once the request is

redirected, the user can start contact the AppServer directly without going through the AppLoaderBalancer during the current session. The *AppController* sit inside the AppLoadBalancer is also in charge of monitoring the AppServers for growing and shrinking as the AppScale deployments happen over the time.

There is no single cloud infrastructure provider has their data centers at all possible locations throughout the world. As a result, all cloud application providers currently have difficulty in meeting SLA expectations for all their customers. Hence, it is logical that each would build bespoke SLA management tools to provide better support for their specific needs. This kind of requirements often arises in enterprises with global operations and applications such as Internet service, media hosting, and Web 2.0 applications. This necessitates building technologies and algorithms for seamless integration of cloud infrastructure service providers for provisioning of services across different cloud providers.

2.6 Business Benefits of Cloud Computing

With cloud computing, organizations can consume shared computing and storage resources rather than building, operating, and improving infrastructure on their own. The speed of change in markets creates significant pressure on the enterprise IT infrastructure to adapt and deliver. In principle, cloud computing enables organizations to obtain a flexible and cost-effective IT infrastructure in much the same way that national electric grids enable homes and organizations to plug into a centrally managed, efficient, and cost-effective energy source. When freed from creating their own electricity, organizations were able to focus on the core competencies of their business and the needs of their customers. In particular, cloud computing technologies have provided some clear business benefits for building software applications. Examples of these benefits are:

1. *No upfront infrastructure investment*: Building a large-scale system may cost a fortune to invest in real estate, hardware (racks, machines, routers, backup power supplies), hardware management (power management, cooling), and operations personnel. Because of the high upfront costs, it usually takes several rounds of management approvals before the project could even get started. With cloud computing, there is no fixed cost or startup cost to start your project.
2. *Just-in-time Infrastructure*: In the past, if your system got famous and your infrastructure could not scale well at the right time, your application may became a victim of its success. On the other hand, if you invested heavily and did not get famous, your application became a victim of your failure. By deploying applications in cloud environments, your application can smoothly scale as you grow.
3. *More efficient resource utilization*: System administrators usually worry about hardware procuring (when they run out of capacity) and better infrastructure utilization (when they have excess and idle capacity). With cloud technology,

they can manage resources more effectively and efficiently by having the applications request resources only what they need on-demand according to the *pay-as-you-go* philosophy.

4. *Potential for shrinking the processing time*: Parallelization is the one of the well-known techniques to speed up processing. For example, if you have a compute-intensive or data-intensive job that can be run in parallel takes 500 h to process on one machine. Using cloud technology, it would be possible to spawn and launch 500 instances and process the same job in 1 h. Having available an elastic infrastructure provides the application with the ability to exploit parallelization in a cost-effective manner reducing the total processing time.

Cloud Data Management

Zhao, L.; Sakr, S.; Liu, A.; Bouguettaya, A.

2014, XIX, 202 p. 86 illus., 50 illus. in color., Hardcover

ISBN: 978-3-319-04764-5