1

Cloud Computing: An Overview

Abhishek Kalapatapu and Mahasweta Sarkar
San Diego State University, San Diego, California

CONTENTS

1.1 Introduction .......................................................... 3
1.2 Cloud Computing: Past, Present, and Future ................ 4
1.3 Cloud Computing Methodologies ............................... 7
1.4 The Cloud Architecture and Cloud Deployment Techniques ... 8
1.5 Cloud Services .......................................................... 13
1.6 Cloud Applications ..................................................... 16
1.7 Issues with Cloud Computing ....................................... 17
1.8 Cloud Computing and Grid Computing: A Comparative Study 19
1.9 Conclusion .............................................................. 25

1.1 Introduction

Cloud computing, with the revolutionary promise of turning computing into a 5th utility, after water, electricity, gas, and telephony, has the potential to transform the face of Information Technology (IT), especially the aspects of service-rendition and service management. Though there are myriad ways of defining the phenomenon of Cloud Computing, we put forth the one coined by NIST (National Institute of Standards and Technology). According to them, Cloud Computing is defined as “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” [84]. Loosely speaking, Cloud computing represents a new way to deploy computing technology to give users the ability to access, work on, share, and store information using the Internet [762]. The cloud itself is a network of data centers, each composed of many thousands of computers working together that can perform the functions of software on a personal or business computer by providing users access to powerful applications, platforms, and services delivered over the Internet. It is in essence a set of network enabled
Cloud Computing: Methodology, System, and Applications

services that is capable of providing scalable, customized and inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way by a wide range of geographically dispersed users. The Cloud also assures application based Quality-of-Service (QoS) guarantees to its users. Thus, Cloud Computing provides the users with large pools of resources in a transparent way along with a mechanism for managing the resources so that a user can access it ubiquitously and without incurring unnecessary performance overhead. The ideal way to describe Cloud Computing then would be to term it as “Everything as a Service” abbreviated as XaaS [100]. Below, we sum up the key features of Cloud Computing:

- Agility – helps in rapid and inexpensive re-provisioning of resources.
- Location Independence – resources can be accessed from anywhere and everywhere.
- Multi-Tenancy – resources are shared amongst a large pool of users.
- Reliability – dependable accessibility of resources and computation.
- Scalability – dynamic provisioning of data helps in avoiding various bottleneck scenarios.
- Maintenance – users (companies/organizations) have less work in terms of resource upgrades and management, which in the new paradigm will be handled by service providers of Cloud Computing.

However, Cloud Computing doesn’t imply that it consists of only one cloud. The term “Cloud” symbolizes the Internet, which in itself is a network of networks. Also, not all forms of remote computing are Cloud Computing. On the contrary, Cloud Computing is nothing but services offered by providers who might have their own systems in place. [18]

1.2 Cloud Computing: Past, Present, and Future

A hundred years ago, companies stopped generating their own power with steam engines and dynamos and plugged into the newly built electric grid. The cheap power pumped out by such electric utilities did not just change how businesses operated — it set off a chain reaction of economic and social transformations that brought the modern world into existence. Today, a similar revolution is under way. Hooked up to the Internet’s global computing grid, massive information-processing plants have begun pumping data and software code into our homes and businesses. This time, it’s computing (instead of electricity) that’s turning into a utility. Nicholas Carr in his book The Big
"Switch: Rewiring the world from Edison to Google" [164] has finely portrayed the transition of Cloud from its past till its present form. Cloud Computing has integrated various positive aspects of different computing paradigms, resulting in a hybrid model that has evolved gradually over the years beginning in 1960 when John McCarthy rightfully stated that “computation may someday be organized as a public utility” [608].

“The Cloud” is based on the idea of generating computing facility on demand. Just the way one turns on a faucet to get water or plug into an electric socket on the wall to get electricity, similarly Cloud Computing intends to create a paradigm where most of the features and functions of stand-alone computers today can be streamed for a user over the Internet [764]. Further probing into the philosophy of cloud computing will reveal that the concept dates back to the era of “Mainframes,” where resources (like memory, computational capabilities) of centralized powerful computers owned by large organizations were used/shared by several users over a small geographical local area. Today Cloud Computing boasts of an architecture where the powerful computers are replaced by supercomputers and perhaps even a network of supercomputers and the users are dispersed over vast geographic areas, accessing these computing resources via the Internet (network of networks). In the past, issues like dearth of bandwidth, perception, lack of control, trust and feasibility proved to be major road blocks in realizing the Cloud concept of service rendition. Today most of these challenges have been overcome, or countermeasures are in place to resolve the challenges. Faster bandwidth, virtualization and more particular skills around cloud type technologies help in realizing the Cloud Computing paradigm.

The concept of Cloud Computing — as nascent as it might appear — itself, has undergone significant evolution. The first generation of Cloud Computing which evolved along with the “Internet Era” was mainly intended for “e-business services.” The current generation of Cloud services has progressed several steps to now include “IT as a Service” which can be thought of as “consumerized internet services.” Using standardized, highly virtualized infrastructure and applications, IT can drive higher degrees of automation and consolidation, thus reducing the cost of maintaining existing solutions and delivering new ones. In addition, externally supplied infrastructure, software, and platform services are delivering capacity augmentation and a means of using operating expense funding instead of a heavy duty capital. [388]

FIGURE 1.1
History of Computing.
Today, “the network is the (very big, very powerful) computer.” The capabilities of the Cloud as a centralized resource can match to industrial scales. This implies that processing power involving thousands of machines embedded in a network has surpassed even the capabilities of the very high-performance supercomputers. By making this technology available through the network on an on-demand and as-needed basis, the Cloud holds the promise of giving individuals, businesses, organizations and governments around the world access to extraordinary computing power from any location and any device. It is a fact that data and information is growing at a break-neck pace. HP predicted that within three more years (by 2014) there will be more information produced and consumed than in the history of mankind. Thus, the next version of Cloud will enable access to information through services that are set in the context of the consumer experience. This is significantly different — it means that data will be separated from the applications — a paradigm where processes can be broken into smaller pieces and automated through a collection of services, woven together with access to massive amounts of data. It will eliminate the need for large scale, complex applications that are built around monolithic processes. Changes can be accomplished by refactoring service models, and integration achieved by subscribing to new data feeds. This will create new connections, new capabilities, and new innovations surpassing those that exist today. [388]. Thus it is envisioned that by 2020 most people will access

![FIGURE 1.2
Clouds Past, Present, and Future.](image)

software applications online and share and access information through the use of remote server networks, rather than depending primarily on tools and information housed on their individual, personal computers. It is predicted
that cloud computing will become more dominant than the desktop in the next decade [84]. In other words, most users will perform the majority of their computing and communicating activities through connections to servers that will be operated and owned by service-providing organizations. Most technologists and industrialists believe that cloud computing will continue to expand and come to dominate information transactions because it offers many advantages, allowing users to have easy, instant, and individualized access to tools and information they need wherever they are, locatable from any networked device. To validate this claim, the PEW INTERNET & AMERICAN LIFE PROJECT carried out a survey with a highly diverse population set. 71% of the survey takers believed that by 2020, most people won’t do their work with software running on a general-purpose PC. Instead, they will work in Internet-based applications such as Google Docs, and in applications run from smartphones [630]. However, quality of service guarantees, interoperability between existing working platforms and security concerns are some of the issues that still continue to plague the growing popularity of Cloud computing.

1.3 Cloud Computing Methodologies

Cloud Computing is based on two main techniques — (i) Service Oriented Architecture and (ii) Virtualization.

(i) Service Oriented Architecture (SOA): Since the paradigm of Cloud computing perceives all tasks accomplished as a “Service” rendered to users, it is said to follow the Service Oriented Architecture. This architecture comprises a flexible set of design principles used during the phases of system development and integration. The deployment of a SOA-based architecture will provide a loosely-integrated suite of services that can be used within multiple business domains. The enabling technologies in SOA allow services to be discovered, composed, and executed. For instance, when an end-user wishes to accomplish a certain task, a service can be employed to discover the required resources for the task. This will be followed by a composition service which will plan the road-map to provide the desired functionality and quality of service to the end-user. [579, 761]

(ii) Virtualization: The concept of virtualization is to relieve the user from the burden of resource purchases and installations. The Cloud brings the resources to the users. Virtualization may refer to Hardware (execution of software in an environment separated from the underlying hardware resources), Memory (giving an application program the impression that it has contiguous working memory, isolating it from the underlying physical memory implementation), Storage (the process of completely abstracting logical storage from physical storage), Software (hosting of multiple virtualized environments within a sin-
gle Operating System (OS) instance), *Data* (the presentation of data as an abstract layer, independent of underlying database systems, structures and storage) and *Network* (creation of a virtualized network addressing space within or across network subnets) [486]. Virtualization has become an indispensable ingredient for almost every Cloud; the most obvious reasons being the ease of abstraction and encapsulation. Amongst the other important reasons for which the Clouds tend to adopt virtualization are:

(i) Server and application consolidation – as multiple applications can be run on the same server resources can be utilized more efficiently.

(ii) Configurability – as the resource requirements for various applications could differ significantly, (some require large storage, some require higher computation capability) virtualization is the only solution for customized configuration and aggregation of resources which are not achievable at the hardware level.

(iii) Increased application availability – virtualization allows quick recovery from unplanned outages as virtual environments can be backed up and migrated with no interruption in services.

(iv) Improved responsiveness – resource provisioning, monitoring and maintenance can be automated, and common resources can be cached and reused. [784]

In addition, these benefits of virtualization tend to facilitate the Cloud to meet stringent SLA (Service Level Agreement) requirements in a business setting which otherwise cannot be easily achieved in a cost-effective manner. Without virtualization, systems have to be over provisioned to handle peak load and hence waste valuable resources during idle periods.

### 1.4 The Cloud Architecture and Cloud Deployment Techniques

Geared with the knowledge of SOA and Virtualization, we now take a look at the overall Cloud architecture. From the end user’s perspective, Figure 4.1 depicts a basic Cloud Computing architecture involving multiple components. Cloud architecture closely resembles the UNIX philosophy of involving multiple components which work together over universal interfaces [762]. Recall that the Cloud computing paradigm represents a Service oriented mechanism of managing and dispatching resources. Before we delve into studying the actual architecture of Cloud computing it will be beneficial to examine the possible characteristics that will be required to realize such a system. It is common knowledge that the architectural requirements of the Cloud will vary depending on the application for which the Cloud is being used. For instance, social networking applications like Facebook and Orkut will have a very differ-
ent set of requirements, constraints and deliverables from the architecture in comparison to, say, a remote patient health monitoring application. However, some common architectural characteristics can still be identified. For instance, (i) the system should be scalable with the ability to include thousands to perhaps tens of thousands of members. (ii) It should be able to interoperate between various service requirements and effectively share resources amongst its users. (iii) The system should be easy to maintain and upgrade, maintaining user transparency during these processes. (iv) As outlined earlier, managing resources like servers and storage devices virtually, thereby creating a virtual organization, is absolutely crucial.

To mitigate the problem of designing customized Cloud architecture for each and every application and also to streamline the architecture design process of the Cloud, scientists resorted to the age old concept of a generalized “Layered approach” [163]. As with the classical 7-layer OSI model of data networks [759], the layered model in Cloud computing serves the same general purpose. Depending on the service requirement of an application, these layers are shuffled to create a customized architecture. The layered architecture adheres to the principle of Service Oriented Architecture (SOA) which forms the core of the Cloud computing paradigm. The components of a basic layered architecture are shown in the Figure 1.4, below [630], namely the Client, its required Services, the Applications that the Client runs, the Platform on which these applications run, the Storage requirement and finally the Infrastructure required to support the Client’s computing needs.

We devote a few sentences on each component below.
Clous Computing: Methodology, System, and Applications

FIGURE 1.4
Layered Architecture for a Customized Cloud Service [760].

Clients: the Clients of a Cloud comprise computer hardware and/or computer software that relies on the computational capability of the Cloud for application or service delivery. Examples include computers, mobile devices, operating systems, and browsers.

Services: this refers to the different service models made available by the Cloud like SaaS (Software-as-a-Service), IaaS (Infrastructure-as-a-Service) and PaaS (Platform-as-a-Service). This layer acts as a middleman between the user and the vast amount of resources accessible to the user. A resource includes “products, services and solutions that are delivered and consumed in real time over the Internet” [630]. Examples include location services and Search Engines among others.

Application: the Cloud enables resource management and user activity tracking from central locations rather than at each customer’s site, enabling customers to access applications remotely via the Internet. Cloud application services deliver software as a service over the Internet, eliminating the need to install and run the application on the customer’s own computer, thereby simplifying (almost eliminating) maintenance and support at the customer’s end. Examples include Web Application and Peer-to-Peer computing.

Platform: it facilitates deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers. This layer delivers a computing platform and/or solution stack as a service, often consuming Cloud infrastructure and sustaining Cloud applications. Examples include Web Application Frameworks like Ruby on Rails and Web Hosting.

Storage: the storage layer consists of computer hardware and/or computer software products that are specifically designed for the storage of Cloud services. Computer hardware comprises huge data centers that are used for resource sharing. Examples include Amazon SimpleDB, and Nirvanix SDN.

© 2012 by Taylor & Francis Group, LLC
Cloud Computing: An Overview

Cloud Computing: An Overview

(Storage Delivery Network).

**Infrastructure:** this layer delivers computer infrastructure, typically a platform virtualization environment as a service. It includes management of virtual resources too. Rather than purchasing servers, software, data center space or network equipment, clients instead buy those resources as a fully outsourced service. Examples include Network Attached Storage and Database services.

The main advantage of such a layered architecture is the ease with which they can be modified to suit a particular service. The way in which these components interact leads to various architectural styles. There are two basic architectural styles on which most of the services are based. They are:

- **Outside-In:** This architectural style is inherently a top-down design emphasizing the functionality of the components. Implementing this style leads to a better architectural layering with various functionalities. It infuses more feasibility enabling better integration and interoperability of components.

- **Inside-Out:** This architectural style, on the other hand, is inherently a bottom-up design which takes an infrastructural point of view of the components. This style is more application oriented than service oriented.

It is to be noted that incorporating new functionalities in a pre-existing architectural scheme is done in an incremental fashion. The ease of transforming an existing architecture into another, depends on the complexity of the architecture, the functionalities of the components and their integration. The vast landscape of the services and their growing complexity has lead to implementation of innovative architectural styles and several hybrid architectures. [221, 631]

**Cloud Deployment Techniques**

Cloud deployment is the manner in which a Cloud is designed to provide a particular service. Obviously these deployment methods will vary according to the way in which a Cloud provides service to the users. Thus, their deployment techniques are user specific [494]. For instance, a deployment technique might depend on the level of security commissioned for a particular user. Figure 1.5 depicts the various Cloud deployment techniques which predominantly comprise (i) the Public deployment, (ii) the Private deployment or (iii) the Hybrid deployment. We discuss each of these deployment strategies briefly below.

(i) **Public Cloud:** it is the traditional mainstream Cloud deployment technique whereby resources are dynamically provisioned by third party providers who share them with the users and bill the users on a fine grained utility computing basis. It offers easy resource management, scalability and flexibility
with an economical pay-as-you-go model which is extremely viable especially for small businesses. On the negative side, the user lacks visibility and control over the computing infrastructure. Since computing infrastructures are shared between various organizations, these Clouds face various security and compliance issues. Amazon’s Web Services and Google’s AppEngine are few examples of Public Clouds, also known as external Clouds. [630]

(ii) **Private Cloud**: in this Cloud deployment technique, the computing infrastructure is solely dedicated to a particular organization or business. These Clouds are more secure because they belong exclusively to a particular organization [494]. These Clouds are more expensive because one needs in-house expertise for their maintenance. Private Clouds are further classified based on their location as:

(a) **On-Premise Clouds** – these refer to Clouds that are for a particular organization hosted by the organization itself. Examples of such Clouds would include Clouds related to military services which have a considerable amount of confidential data.

(b) **Externally hosted Clouds** – these refer to Clouds that are also dedicated for a particular organization but are hosted by a third party specializing in Cloud infrastructure. These are cheaper than On-premise Clouds. Examples of such Clouds would be small businesses using services from VMware, Amazon etc. Such Clouds are also known as Internal Clouds. [631]

(iii) **Hybrid Cloud**: this deployment technique integrates the positive attributes of both the Public Cloud and Private Cloud paradigm. For instance, in a Hybrid Cloud deployment, critical services with stringent security requirements may be hosted on Private Clouds while less critical services can be
hosted on the Public Clouds. The criticality, flexibility and scalability requirement of a service governs its classification into either the Public or Private Cloud domain. Each Cloud in the Hybrid domain retains its unique entity. However, they function synchronously to gracefully accommodate any sudden rise in computing requirements. Hybrid Cloud deployment is definitely the current trend amongst the major leading Cloud providers currently [630].

(iv) **Community Cloud:** this deployment technique is similar to Public Clouds with the only difference being the distribution of the sharing rights on the computing resources. In a Community Cloud, the computing resources are shared amongst organizations of the same community. So this Cloud covers a particular group of organizations, which have the same functionalities. For example, all Government organizations within the state of California may share the computing infrastructure on the Cloud to manage data related to citizens residing in California [608].

1.5 **Cloud Services**

The Cloud can provide us with a myriad service models and services. They include SaaS (Software as a Service), PaaS (Platform as a Service), HaaS (Hardware as a Service), DaaS ([Development, Database, Desktop] as a Service), IaaS (Infrastructure as a Service), BaaS (Business as a Service), FaaS (Framework as a Service), OaaS (Organization as a Service) amongst oth-
Cloud Computing: Methodology, System, and Applications

However, Cloud Computing products can be broadly classified into three main Services (SaaS, PaaS and IaaS) which are showcased in Figure 1.6 along with their relationship to a user (Enterprise). The following section is an attempt to familiarize the reader with the several different Cloud Services that are currently rendered:

**Infrastructure-as-a-Service (IaaS):** This service provisions for hardware related services like storage, and virtual servers on a pay-as-you-go basis. The main advantage of IaaS is the usage of latest technology at all times with regard to computer infrastructure which allows users to achieve faster service. Organizations can use IaaS to quickly build new versions of applications or environments without incurring unnecessary purchase and configuration delay. On-demand scaling via resource virtualization and use-based billing makes IaaS competent enough for any kind of businesses. The major companies already providing IaaS are Amazon [57, 58], Rackspace, GoGrid, AT&T and IBM. [367]

**Platform-as-a-Service (PaaS):** PaaS offerings may include facilities for application design, application development, testing, deployment and hosting as well as application services such as team collaboration, web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation and developer community facilitation. These services may be provisioned as an integrated solution over the web, providing an existent managed higher-level software infrastructure for building particular classes of applications and services. The platform includes the use of underlying computing resources, typically billed similar to IaaS products, although the infrastructure is abstracted away below the platform. Major companies providing PaaS are Google’s AppEngine [92], Microsoft Azure, and Force.com etc. [608, 630, 631]

**Software-as-a-Service (SaaS):** Provides specific already created applications as fully or partially remote services. Sometimes it is in the form of web-based applications and other times it consists of standard non-remote applications with Internet-based storage or other network interactions. It allows a user to use the provider’s application using a thin client interface. Users can access a software application hosted by the Cloud vendor on pay-per-use basis [527]. It is a multi-tenant platform. The pioneer in this field has been Salesforce.com offering online Customer Relationship Management (CRM) space. Other examples are online email providers like Google’s Gmail and Microsoft’s hotmail, Google docs and Microsoft’s online version of office called BPOS (Business Productivity Online Standard Suite). [16, 608, 630, 631]

Other than the above services David Linthicum has described a more granular classification of services which includes:

**Storage-as-a-Service (SaaS):** Storage as a Service is a business model that helps a smaller company or individual in renting storage spaces from a large company. Storage as a Service is generally seen as a good alternative for a
small or mid-sized business that lacks the capital budget and/or technical personnel to implement and maintain their own storage infrastructure. SaaS is also being promoted as a way for all businesses to mitigate risks in disaster recovery, provide long-term retention for records and enhance both business continuity and availability. Examples include Nirvanix, Cleversafe’s dsNET etc.

**Database-as-a-Service (DbaaS):** It constitutes delivery of database software and related physical database storage as a service. A managed service, offered on a pay-per-usage basis that provides on-demand access to a database for the storage of application data is what constitutes DbaaS. Examples include Amazon, Force.com etc.

**Information-as-a-Service (IfaaS):** Information as a service accepts the idea that data resides within many systems and repositories. Its main function is to standardize the access of data by applying a standard set of transformations to the various sources of data thus enabling service requestors to access the data regardless of vendor or system. Examples include IBM, Microsoft etc.

**Process-as-a-Service (PraaS):** Refers to a remote resource that’s able to bind many resources together, either hosted within the same Cloud computing resource or remote, to create business processes. These processes are typically easier to change than applications, and thus provide agility to those who leverage these process engines that are delivered on-demand. Process-as-a-service providers include Appian Anywhere, Akemma, and Intensil.

**Integration-as-a-Service (InaaS):** Integration-as-a-Service includes most of the features and functions found within traditional Enterprise Application Integration technology, but delivered as a service. Integration-as-a-Service takes the functionality of system integration and puts it into the Cloud, providing for data transport between the enterprise and SaaS applications or third parties. Examples include Amazon SQS, OpSource Connect, Boomi, and Mule OnDemand.

**Security-as-a-Service (SeaaS):** Delivers core security services remotely over the Internet like anti-virus, log management etc. While typically the security services provided are rudimentary, more sophisticated services are becoming available such as identity management. Security-as-a-Service providers include Cisco, McAfee, Panda Software, Symantec, Trend Micro and VeriSign.

**Management/Governance-as-a-Service (MaaS):** Provides the ability to manage one or more Cloud services, typically simple things such as topology, resource utilization, virtualization, and uptime management. Governance systems are becoming available as well, such the ability to enforce defined policies on data and services. Management/governance-as-a-service providers include RightScale, rPath, Xen, and Elastra.

**Testing-as-a-Service (TaaS):** These systems have the ability to test other Cloud applications, Web sites, and internal enterprise systems, and do not require a hardware or software footprint within the enterprise. They also have the ability to test services that are remotely hosted. SOASTA is one of the
many Testing-as-a-Service providers [608, 631].

1.6 Cloud Applications

Cloud Applications are not only developed with a business perspective but also take into account activities oriented towards socializing and sharing information. This information may be as basic as checking news headlines or more sensitive in nature, such as searches for health or medical information. Thus Cloud Computing is often a better alternative than local servers handling such applications.

Virtualization is the main basis of Cloud Computing, thus it is fully enabled with virtual appliances. A virtual appliance is an application which has all its components bundled and streamlined with the operating system [762]. The main advantage of a Cloud Computing application is that the provider can run various instances of an application with minimum labor and expense. A Cloud service provider needs to anticipate a few issues before launching its application in the Cloud computing environment. Keeping in mind the issues an application must be designed to scale easily, tolerate failures and include management tools [388]. We discuss these issues in the following section.

- **Scale:** Application in a Cloud environment needs to have maximum scalability and to ensure this, one should start building the application in the simplest manners avoiding complex design patterns and enhancements. The next step would be to split the functions of an application and integrate them loosely. The most important step in ensuring on demand scalability of an application is sharding, which can be described as splitting up the system into many smaller clusters instead of scaling the single system up so as to serve all users.

- **Failures:** Any application due to one or the other reason is bound to face failure at some point of time. To tolerate failures, an application must operate in an asynchronous fashion and one should spread the load across multiple clusters so that the impact of failure gets distributed. The best way to tolerate failures would be testing the application for all kinds of failure scenarios, and also users should be aware of the real cost incurred if an application faces any kind of failure.

- **Management Tools:** Having a proper management tool helps in automating the application configuration and updates, thereby reducing management overhead. The management system helps in not just minimizing economic expenditures but also leading to optimized usage of resources. The most
difficult and expensive problem that can be handled with a proper management tool is variability \([795]\). This helps in delivering an application that can boast of consistent performance.

Applications in Cloud Computing have been designed keeping in mind the various types of services offered by it. Cloud Computing has impacted people’s perception of applications over the Internet. The applications of Cloud computing are streamlined with each and every field of sciences. In Figure 1.7 we have highlighted a few areas in which Cloud Computing has shown great potential for developing various applications which have fostered the growth of businesses and enterprises.

These applications can be categorized into broad areas like datacenters and storage, security, military applications, the health industry, platforms and software usages, applications with high performance computing resources, and last but not the least, the growing need for virtual environments. \([221, 494]\)

1.7 Issues with Cloud Computing

So far we have focused on the promises that the Cloud holds for users and applications alike. However, there are issues that need to be resolved before this technology can be exploited to its maximum potential \([238]\). In the section below we enumerate and discuss the burning issues which might deter the phenomenal growth of Cloud Computing technology.

**(i)** Security Issues: Security is as much of a concern in Cloud computing as it would be in any other computing paradigms. Cloud Computing can be vaguely defined as outsourcing of services, which in turn causes users to lose significant control over their data. There is always a risk of seizure associated with the public Clouds. For instance, an organization sharing data in an environment
where other organizations are doing the same is always under the threat of compromising the security and privacy of its data if any other organization in the shared scheme happens to violate the security protocols [390]. Moreover, in a virtualized environment one needs to consider the security of not just the physical host but also the virtual machine. This is because if the security of a physical host is compromised, then automatically all virtual machines face security threat and vice versa. Since the majority of services in Cloud computing are provided using web browsers, there are many security issues related with it as well [392]. Flooding is also a major issue where an attacker sends huge amounts of illegitimate service requests which cause the system to run slow thereby hampering the performance of the overall system. Cloud networks stand the potential threat of both Indirect Denial of Service attacks and Distributed Denial of Service attacks.

(ii) Legal and Compliance Issues: Clouds are sometimes bounded by geographical boundaries. Provision of various services is not location dependent but because of this flexibility Clouds face Legal & Compliance issues. These issues are related mainly to the vendors though they still affect the end users. These issues are broadly classified as functional (services in the Clouds that have legal implications for both service providers and end users), jurisdictional (where governments administer laws to follow) and contractual (terms and conditions). Issues include (a) Physical Location of the data referring to where the data is physically located and if a dispute occurs, which jurisdiction will help in resolving it (b) Responsibilities of the data where if a vendor is hit by a disaster will the businesses using its services be covered under insurance (c) Intellectual Property Rights which deals with the way trade secrets are maintained [732].

(iii) Performance and QoS Related Issues: For any computing paradigm performance is of utmost importance. Quality of Service (QoS) varies as the user requirements vary. One of the critical QoS related issues is the optimized way in which commercial success can be achieved using Cloud computing. If a provider is not able to deliver the promised QoS it may tarnish its reputation [238]. Since Software-as-a-Service (SanS) deals with provision of softwares on virtualized resources, one faces the issue of Memory and Licensing constraints which directly hamper the performance of a system.

(iv) Data Management Issues: The main purpose of Cloud Computing is to put the entire data on the Cloud with minimum infrastructure requirements for the end users. The main issues related to data management are scalability of data, storage of data, data migration from one Cloud to another and also different architectures for resource access [127]. Since data in Cloud computing even includes high confidential information it is of utmost importance to manage these data effectively. There has been an instance where an online storage service called The Linkup got shut down after losing access to as much as 45% of its customers. While transferring data, i.e., data migration, in a Cloud has to be done very carefully as it could lead to bottlenecks at each
and every layer of the network model, as huge chunks of data are associated with the Cloud [762].

(v) Interoperability Issues: The Cloud computing interoperability idea was conceived by Reuven Cohen. Reuven Cohen is founder and Chief Technologist for Toronto based Enomaly Inc. Vint Cerf, who is a co-designer of the Internet’s TCP/IP standards and widely considered a father of the Internet, spoke about the need for data portability standards for Cloud computing. Companies such as Microsoft, Amazon, IBM, and Google all own their independent Clouds but they lack interoperability amongst them. Each service provider has its own architecture, which caters to a specific application. To make such uniquely distinct Clouds interoperate is a non-trivial problem. The lack of standardized protocols in the domain of Cloud computing further makes interoperability a challenge. The key issues hampering implementation of interoperable Clouds are the large scale access and computational capabilities of the Clouds, resource contention and the dynamic nature of the Cloud. However, interoperability amongst the various Clouds would only add to the value of this technology, making it more widely accessible, fault tolerant, and thereby robust.

1.8 Cloud Computing and Grid Computing: A Comparative Study

Cloud computing should not be confused with Grid Computing, Utility Computing and Autonomic Computing. Grid computing consists of clusters of loosely coupled, networked computers acting in concert to perform very large tasks. Utility computing packages computing resources as a metered service. Autonomic computing stresses self management of resources. Grids require many computers, typically in the thousands, and commonly use servers, desktops, and laptops. Clouds also support non-grid environments [284]. The differences between Utility computing and Cloud computing are crucial. Utility computing relates to the business model in which application, infrastructure, resources, hardware and/or software are delivered. In contrast, Cloud computing relates to the way we design, build, deploy and run applications that operate in a virtualized, shared-resource environment accompanied by the coveted ability to dynamically grow, shrink and self-heal. Thus the major aspects which separate Cloud Computing from other computing paradigms are user centric interfaces, on-demand services, QoS guarantees, autonomous system organization, scalability and flexibility of services [369].

In the mid 1990s, the term “Grid” was coined to describe technologies that would allow consumers to obtain computing power on demand. Ian Foster and others posited that by standardizing the protocols used to request com-
Cloud Computing: Methodology, System, and Applications

FIGURE 1.8
Cloud and Grid Computing [283].

Web 2.0 covers almost the whole spectrum of service-oriented applications, whereas Cloud Computing lies mostly on the large-scale computing domain. Supercomputing and Cluster computing have been more focused on traditional non-service applications. Grid Computing overlaps with all these fields and is generally considered to be catering to smaller scaled computing requirements than the Supercomputers and the Clouds [284].

Half a decade ago, Ian Foster gave a three point checklist to help define what is, and what is not a Grid. We present his checklist below:
1.) Coordinates resources that are not subject to centralized control,
2.) Uses standard, open, general-purpose protocols and interfaces, and
3.) Delivers non-trivial qualities of service.

Although the third point holds true for Cloud Computing, neither point one nor two is applicable for today’s Clouds. The vision for Clouds and Grids are similar but the implementation details and technologies used differ considerably. In the following section, we discuss the differences in these two technologies based on their architecture, business model, and resource management techniques.

Architecture
Grids provide protocols and services at five different layers as identified in the Grid protocol architecture shown in Figure 1.9.
At the fabric layer, Grids provide access to different resource types such as computation, storage and network resource, code repository, etc. The connectivity layer defines core communication and authentication protocols for easy and secure network transactions. The resource layer defines protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources. The collective layer captures interactions across collections of resources and directory services. The application layer comprises whatever user applications are built on top of the above protocols and APIs environments. [284, 370]

In contrast, the Cloud architecture presents a four-layer architecture composed of (i) Fabric (ii) Unified resource (iii) Platform and (iv) Application layers. (It is to be noted that there are other paradigms of cloud architecture as well. We however, choose to focus on the 4-layer architecture for our discussion).

The Fabric layer contains the raw hardware level resources, such as computational resources, storage resources, and network resources. The Unified Resource Layer contains resources that have been abstracted/encapsulated (usually by virtualization) so that they can be exposed to upper layer and end users as integrated resources, for instance, a virtual computer/cluster, a logical file system, a database system, etc. The Platform layer adds a collection of specialized tools, middleware and services on top of the unified resources to provide a development and/or deployment platform. Finally, the Application layer contains the applications that run in the Clouds. Clouds in general provide services at three different levels, namely IaaS, PaaS, and SaaS, although some providers can choose to expose services at more than one level. [284]

Resource Management
With regard to resource management we compare Clouds and Grids using the following models:

(i) **Computational Model:** Most Grids use a batch-scheduled computational model, in which a local resource manager (LRM), such as PBS, Condor and SGE manages the computational resources for a Grid site. Users submit batch jobs via GRAM that occupies a certain amount of resources for a stipulated period of time \[787\]. In contrast to the Grid’s paradigm of allocating dedicated resources to specific “jobs” for a stipulated amount of time with a scheduling algorithm governing the process, the Cloud computing model takes a resource sharing approach by all users at all times \[371\]. The goal here (in Cloud computing) is to achieve low queuing latency for resource access. Moreover, this design should allow latency sensitive applications to operate natively on Clouds, although ensuring the required level of QoS is likely to be one of the major challenges for Cloud Computing, especially as the Cloud scales to incorporate more users. \[284\]

(ii) **Data model:** The importance of data has caught the attention of the Grid community for the past decade; Data Grids have been specifically designed to tackle data intensive applications in Grid environments, with the concept of virtual data playing a crucial role. Virtual data captures the relationship between data, programs, and computations and prescribes various abstractions that a data grid can provide. For instance, (a) location transparency where data can be requested without regard to data location, a distributed metadata catalog is engaged to keep track of the locations of each piece of data (along with its replicas) across grid sites, and privacy and access control are enforced; (b) materialization transparency where data can be either recomputed on the fly or transferred upon request, depending on the availability of the data and the cost to re-compute \[378\]. There is also representation transparency where data can be consumed and produced no matter what their actual physical formats and storages are, data are mapped into some abstract structural representation and manipulated in that way. In contrast, the Cloud computing paradigm is centralized around Data. Cloud Computing and Client Computing will coexist and evolve hand in hand, while data management (mapping, partitioning, querying, movement, caching, replication, etc.) will become more and more important for both Cloud Computing and Client Computing with the increase of data-intensive applications. \[163\]

(iii) **Virtualization:** Grids do not rely on virtualization as much as Clouds do, but that might be more due to policy issues and having each individual organization maintain full control of their resources (i.e., not by virtualization). However, there are efforts in Grids to use virtualization as well, such as Nimbus, which provides the same abstraction and dynamic deployment capabilities. A virtual workspace is an execution environment that can be deployed dynamically and securely in the Grid. In addition, Nimbus can also provision
Cloud Computing: An Overview

23

a virtual cluster for Grid applications (e.g., a batch scheduler, or a workflow system), which is also dynamically configurable — a growing trend in Grid Computing. [284]

(iv) Monitoring: Another challenge that virtualization brings to the Clouds is the potential difficulty in retaining fine-grained control over the monitoring of resources. Although many Grids (such as TeraGrid) also enforce restrictions on what kind of sensors or long-running services a user can launch, Cloud monitoring is not as straightforward as in Grids, because Grids in general have a different trust model in which users via their identity delegation can access and browse resources at different Grid sites, and Grid resources are not highly abstracted and virtualized as in Clouds. Monitoring can be argued to be less important in Clouds, as users interact with a more abstract layer that is potentially more sophisticated. This abstract layer could respond to failures and quality of service (QoS) requirements automatically in a general-purpose way irrespective of application logic. In the near future, user-end monitoring might be a significant challenge for Clouds, but it will become less important as Clouds become more sophisticated and more self-maintained and self-healing.

(v) Programming Model: Grids primarily target large-scale scientific computations, so they must scale to leverage large number/amount of resources, and we would also naturally want to make programs run fast and efficiently in Grid environments. Programs must also run correctly, so reliability and fault tolerance must be considered. Clouds (such as Amazon Web Services, Microsoft’s Azure Services Platform) have generally adopted Web Services APIs where users access, configure and program Cloud services using pre-defined APIs exposed as Web services, and HTTP and SOAP are the common protocols chosen for such services. Although Clouds adopted some common communication protocols such as HTTP and SOAP, the integration and interoperability of all the services and applications remain the biggest challenge as users need to tap into a federation of Clouds instead of a single Cloud provider. [519]

(v) Application model: Grids generally support many different kinds of applications, ranging from high-performance computing (HPC) to high-throughput computing (HTC). HPC applications are efficient at executing tightly coupled parallel jobs within a particular machine with low-latency interconnects and are generally not executed across a wide area network Grid. On the other hand, Cloud computing could in principle cater to a similar set of applications. The one exception that will likely be hard to achieve in Cloud computing (but has had much success in Grids) are HPC applications that require fast and low latency network interconnects for efficient scaling to many processors. As Cloud computing is still in its infancy, the applications that will run on Clouds are not well defined, but we can certainly characterize them to be loosely coupled, transaction oriented (small tasks in the order of millisec-
onds to seconds), and likely to be interactive (as opposed to batch scheduled as they are currently in Grids).

**(vi) Security Model:** Clouds mostly comprise dedicated data centers belonging to the same organization, and within each data center, hardware and software configurations and supporting platforms are in general more homogeneous as compared with those in the Grid environments. Interoperability can become a serious issue for cross-data center and cross-administration domain interactions. Imagine running your accounting service in Amazon EC2 while your other business operations are on Google infrastructure. Grids, however, are built on the assumption that resources are heterogeneous and dynamic, and each Grid site may have its own administration domain and operation autonomy. Thus security has been engineered in the fundamental Grid infrastructure. The key issues considered are: single sign-on, so that users can log on only once and have access to multiple Grid sites; this also facilitates accounting and auditing, delegation (so that a program can be authorized to access resources on a user’s behalf and it can further delegate to other programs), privacy, integrity and segregation. Moreover, resources belonging to one user cannot be accessed by unauthorized users and cannot be tampered with during transfer. In contrast, currently, the security model for Clouds seems to be relatively simpler and less secure than the security model adopted by the Grids. Cloud infrastructure typically relies on Web forms (over SSL) to create and manage account information for end-users, and allows users to reset their passwords and receive new passwords via emails in an unsafe and unencrypted communication. Note that new users could use Clouds relatively easily and almost instantly, with a credit card and/or email address. To contrast this, Grids are stricter about security.

**Business models:**
In a Cloud-based business model, a consumer pays the provider on the basis of resource consumption, akin to the utility companies charging for basic utilities such as electricity, gas, and water. The model relies on economics of scale in order to drive prices down for users and profits up for providers. Today, Amazon essentially provides a centralized Cloud consisting of Compute Cloud EC2 and Data Cloud S3. The former is charged based on per instance-hour consumed for each instance type and the latter is charged by per GB-Month of storage used. In addition, data transfer is charged by TB/month data transfer, depending on the source and target of such transfer. The business model for Grids (at least that found in academia or government labs) is project-oriented, in which the users or community represented have a certain number of service units (i.e., CPU hours) they can spend.

Thus Clouds and Grids share a lot of commonality in their vision, architecture and technology, but they also differ in various aspects such as security, programming model, business model, computational model, data model, ap-
Applications, and abstractions. We believe a close comparison such as this can help the two communities understand, share, and evolve infrastructure and technology within and across, and accelerate Cloud computing to leap from early prototypes to production systems. [360]

1.9 Conclusion

Cloud Computing plays a significant role in varied areas like e-business, search engines, data mining, virtual machines, batch oriented scientific computing, online TV amongst many others. Cloud computing has the potential to become an integral part of our lives. Examples include, (i) the Cloud operating system which provides users with all the basic features of an operating system like data storage and applications; (ii) mapping services which help users in finding routes to various places; (iii) Telemedicine applications for collecting data of a patient and calling emergency services in dire need.

As an increasing number of businesses move toward Cloud based services, issues like interoperability, security, portability, migration and standardized protocols are proving to be critical concerns. For instance, the need for higher transparency in scheduling tasks with guaranteed QoS is proving to be a challenging issue in the area of data management over the Clouds. In addition, the service driven model of Cloud Computing often leads to concerns and queries regarding the Service Level Agreement (SLA) of enterprises.

The research and business community alike are coming up with new and innovative solutions to tackle the numerous issues that are making its way as Cloud Computing is shaping itself as the Future of Internet. The potentials are endless and the sky is the limit as we watch “a computing environment to elastically provide virtualized resources as a service over the Internet in a pay-as-you-go manner” [631] bloom and blossom to its maximum potential.

APPENDIX: Comparisons of different Cloud computing technologies

In this section, we present a comparative data-set that showcases the various solution and service strategies of some of the major Cloud service providers today, like Amazon Web Services, GoGrid, FlexiScale, Google, Nimbus and a few others. Detailed description of this data set is available at [608].
<table>
<thead>
<tr>
<th>Vendor</th>
<th>Service and outage</th>
<th>Outage Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>Malfunction in Windows Azure</td>
<td>22 hours</td>
</tr>
<tr>
<td>Google</td>
<td>Gmail and Google Apps</td>
<td>2.5 hours</td>
</tr>
<tr>
<td></td>
<td>Google search outage due to programming error</td>
<td>40 Mins</td>
</tr>
<tr>
<td></td>
<td>Gmail site unavailable due to outage in contacts system</td>
<td>1.5 hours</td>
</tr>
<tr>
<td></td>
<td>Google App engine partial outage</td>
<td>5 hours</td>
</tr>
<tr>
<td>Amazon</td>
<td>Authentication overload</td>
<td>2 hours</td>
</tr>
<tr>
<td></td>
<td>single bit error leading to protocol blowup</td>
<td>6–8 hours</td>
</tr>
<tr>
<td>Flexiscale</td>
<td>Core network failure</td>
<td>18 hours</td>
</tr>
</tbody>
</table>
### TABLE 1.2
Comparison of Different Cloud Computing Technologies and Solution Provider

<table>
<thead>
<tr>
<th>Feature</th>
<th>Amazon Web Services</th>
<th>GoGrid</th>
<th>Flexiscale</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Computing Architecture</strong></td>
<td>Provides IaaS. Gives client API’s to manage their Infrastructure</td>
<td>Provides IaaS. Designed to deliver a guaranteed QoS level and reconfigures itself depending on demand.</td>
<td>Provides IaaS. Functions similar to GoGrid’s architecture but allows multi-tier architectures.</td>
</tr>
<tr>
<td><strong>Fault Tolerance</strong></td>
<td>System alerts automatically, does failover and resyncs to last known state.</td>
<td>Instantly scalable and reliable file level backup service.</td>
<td>Full Self-Service.</td>
</tr>
<tr>
<td><strong>Interoperability</strong></td>
<td>Supports horizontal interoperability</td>
<td>Working towards interoperability.</td>
<td>Working towards interoperability.</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>S3 and SimpleDB</td>
<td>First connects each server to private network and then uses transfer protocols.</td>
<td>Persistent storage based on a fully virtualized high end SAN/NAS back end.</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Type-I, Firewall, SSL, Access Control list.</td>
<td>No guarantee of security.</td>
<td>Customers have their own VLAN.</td>
</tr>
</tbody>
</table>
### Table 1.3
Features of Different PaaS and SaaS Providers

<table>
<thead>
<tr>
<th>Feature</th>
<th>Google App Engine</th>
<th>Azure</th>
<th>Force.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing Architecture</td>
<td>Google's geo-distributed architecture</td>
<td>Platform is hosted on Microsoft datacenters. Provides an OS and a set of developer’s cloud services.</td>
<td>Facilitates multi-tenant architecture allowing a single application to serve multiple customers.</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>Automatic scaling and Load Balancing</td>
<td>Built in hardware Load Balancing. Containers are used as load balancer.</td>
<td>Load Balancing among tenants.</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td>Automatically pushed to a number of fault tolerant servers.</td>
<td>On Failure SQL data services will automatically begin using another replica of container.</td>
<td>Self-Management and Self-Learning.</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Supports interoperability between platforms of different vendors and programming languages.</td>
<td>Supports interoperability between platforms.</td>
<td>Application level integration between different clouds.</td>
</tr>
<tr>
<td>Storage</td>
<td>Proprietary database (Big Table distributed storage)</td>
<td>SQL Server Data Services (SSDS).</td>
<td>Database deals in terms of relationship fields.</td>
</tr>
<tr>
<td>Security</td>
<td>Google’s Secure Data Connector (SDC). SDC uses TLS based server authentication and uses RSA/128-bit or higher AES CBC/SHA.</td>
<td>Security Token Service (STS) creates a security assertion markup language token according to rule.</td>
<td>SysTest SAS, 70 Type II.</td>
</tr>
<tr>
<td>Programming Framework</td>
<td>MapReduce framework supporting Python and Java</td>
<td>Microsoft .NET.</td>
<td>Apex for database service and supports .NET, Apache Axis (Java, C++).</td>
</tr>
</tbody>
</table>
### TABLE 1.4
Comparisons of Different Open Source-Based Cloud Computing Services

<table>
<thead>
<tr>
<th>Feature</th>
<th>Eucalyptus</th>
<th>OpenNebula</th>
<th>Nimbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing Architecture</td>
<td>Can configure multiple clusters in a Private Cloud</td>
<td>It is based on Haizea scheduling. Focuses on efficient, dynamic and scalable management of VM’s within private clouds.</td>
<td>It has a client-side cloud computing interface to Globus enabled TeraPort cluster. Nimbus Context Broker combines several deployed VM’s into “turnkey” virtual cluster.</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>Simple Load Balancing cloud controller</td>
<td>Nginx server is used as Load Balancer with round-robin or weighted selection mechanism.</td>
<td>Triggers self configuring virtual clusters.</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td>Separate clusters within a cloud reduce the risk of correlated failure.</td>
<td>Daemon is restarted and a persistent database backend is used to store host and VM info.</td>
<td>Checks worker nodes periodically and performs recovery operation.</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Multiple private clouds use the same backend infrastructure.</td>
<td>Interoperable between intra cloud services.</td>
<td>Standards “rough consensus and working code.”</td>
</tr>
<tr>
<td>Storage</td>
<td>Walrus is used.</td>
<td>SQLite3 is used as a backend component. Persistent storage for ONE data structures.</td>
<td>GridFTP and SCP.</td>
</tr>
<tr>
<td>Security</td>
<td>WS-security for authentication, cloud controller generates the public/private key.</td>
<td>Firewall, virtual private network tunnel.</td>
<td>PKI credential required, works with GRID proxies VOMS, shibboleth custom PDP’s.</td>
</tr>
<tr>
<td>Programming Framework</td>
<td>Supports Hibernate, Axis2 and Java.</td>
<td>Supports Java and Ruby.</td>
<td>Supports Python and Java.</td>
</tr>
</tbody>
</table>