Cloud Computing- SPI Framework, Deployment Models, Challenges
Nikita Goel¹, Toshi Sharma²
nikitagoel09@gmail.com, toshisharma03@gmail.com

Abstract— The recent emergence of cloud computing has drastically altered everyone’s perception of infrastructure architectures, software delivery and development models. This paper discusses the concept of “cloud” computing with some of the available definitions, SPI framework, and some of the challenges the technology is facing. This paper also includes a brief description of the comparison of Cloud Computing and Grid Computing.

Keywords— Cloud Computing, Cloud Definition, SPI Framework, Virtualization, Deployment Models

I. INTRODUCTION

Cloud Computing is a “buzz word” encompassing a wide variety of aspects. Clouds are a large pool of easily useable and accessible virtualized resources (hardware, development platforms and/or services). Most of us are already using cloud computing in our daily lives for personal use, and now enterprises are rapidly moving key applications to the clouds for agility (speed of implementation and speed of deployment), improved customer experience, scalability, and cost control. The cloud is where you go to use technology when you need it, for as long as you need it. The pool of resources is typically exploited by a pay-per-use model. You do not install anything on your desktop, and you do not pay for the technology when you are not using it.

The Cloud concept is based on some of the existing technologies such as virtualization [13], utility computing and distributed computing. Virtualization is the first step towards building a cloud. Cloud Computing is having secure access to all your applications and data from any network devices.

As cloud computing is achieving increased popularity, concerns are being voiced about the security issues introduced through the adoption of this new model. The effectiveness and efficiency of traditional protection mechanisms are being reconsidered, as the characteristics of this innovative deployment model, differ widely from them of traditional architectures.

This paper comprises of grid and cloud comparison, SPI framework, deployment models, concepts, benefits and challenges of cloud computing.

II. CLOUDS AND GRIDS COMPARISON

A source of confusion around the Cloud concept is its relation with Grid Computing [21, 22]. The distinctions are not clear may be because Clouds and grids share similar visions: reduce computing costs and increase flexibility and reliability by using third-party operated hardware.

A. Grid Computing

In the mid 1990s, the term “Grid Computing” was derived from the electrical power grid to emphasize its characteristics like pervasiveness, simplicity and reliability (Foster and Kesselman 1999). The development of Grid Computing and its standards was mainly driven by scientific Grid Communities. The major community is the Open Grid Forum (OGF, http://www.ogf.org/). In 2002, Ian Foster [3] proposed a definition of the Grid as “a system that coordinates resources which are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service”. For Foster (2002), Grid Computing is characterized by

- decentralized resource control, i.e. the Grid resources are locally dispersed and span multiple administrative domains;
- standardization, i.e. the Grid middleware is based upon open and common protocols and interfaces;
- non-trivial qualities of service, e. g. regarding latency, throughput, and reliability;
- Diversity and Dynamism.
- Distributed job management and scheduling.

B. Cloud Computing

Cloud is basically a collection of resources connected over an internet (WAN). It is made up of data centres. For Cloud Computing, there is no established definition yet, which describes its impact on the technology and business landscape. There are many definitions of Cloud Computing [1], but they all seem to focus on just some aspects of the technology. Cloud computing is based on five attributes: multitenancy (shared resources), massive scalability, elasticity, pay as you go, and self-provisioning of resources.
Cloud computing is based on a business model in which resources are shared (i.e., multiple users use the same resource) at the network level, host level, and application level.

Massive scalability Cloud computing provides the ability to scale to thousands of systems, as well as the ability to massively scale bandwidth and storage space.

Elasticity Users can rapidly increase and decrease their computing resources as needed, as well as release resources for other uses when they are no longer required.

Pay as you go Users pay for only the resources they actually use and for only the time they require them.

Self-provisioning of resources Users self-provision resources, such as additional systems (processing capability, software, storage) and network resources.

Buyya et al. [5] have defined it as follows:

“Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers.”

The National Institute of Standards and Technology (NIST) [24,6] characterizes cloud computing as

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

For Boss et al. (2007, p. 4) [19], “a Cloud is a pool of virtualized computer resources”. They consider Clouds to complement Grid environments by supporting the management of Grid resources. In particular, according to Boss et al. (2007), Clouds allow

- the dynamic scale-in and scale-out of applications by the provisioning and de-provisioning of resources, e.g. by means of virtualization;
- the monitoring of resource utilization to support dynamic load-balancing and reallocations of applications and resources.

Most importantly, Boss et al. (2007, p. 4) stress that Clouds are not limited to Grid environments, but also support “interactive, user-facing applications” such as Web applications and three-tier architectures.

For Weiss (2007) [23], Cloud Computing is not a fundamentally new paradigm. It draws on existing technologies and approaches, such as Utility Computing, Software-as-a-Service, distributed computing, and centralized data centers. What is new is that Cloud Computing combines and integrates these approaches. Especially the combination with Utility Computing and data centers seems to differentiate Cloud Computing from Grid Computing.

III. SPI FRAMEWORK FOR CLOUD COMPUTING

A commonly agreed upon framework for describing cloud computing services goes by the acronym “SPL.” This acronym stands for the three major services provided through the cloud: software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service (IaaS) [17].

A. Infrastructure as a Service - The IaaS model provides the required infrastructure to run the applications. A cloud infrastructure enables on-demand provisioning of servers running several types of operating systems and a customized software stack. The provider is in complete control of the infrastructure. Infrastructure services are considered to be the bottom layer of cloud computing systems. Example Amazon Web Services [7,8] mainly offers IaaS, which in the case of its EC2 service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized.

Benefits

- Tremendous control to use whatever content makes sense.
- Flexibility to secure data to whatever degree necessary
- Physical Independence from infrastructure (you don’t have to ensure that proper cooling is there etc.)

Issues

- Integration of all aspects of application (database, application program, plugin etc.)
- Responsible for all configuration implemented on the server (and in application)
- Responsible for keeping software up to date.
- Multi-tenancy at hypervisor level.

B. Platform as a Service - In a platform-as-a-service (PaaS) model, the service provider offers a development environment to application developers, who develop applications and offer those services through the provider’s platform.
A cloud platform offers an environment on which developers create and deploy applications and do not necessarily need to know how many processors or how much memory that applications will be using. Example Google App Engine [9], an example of Platform as a Service, offers a scalable environment for developing and hosting Web applications, which should be written in specific programming languages such as Python or Java, and use the services’ own proprietary structured object data store.

**Benefits**
- Deploy consumer created applications using programming language and tools supported by the cloud service provider.
- Reduce complexity because CSP is maintaining the environment.
- The cloud service provider often uses it’s API (a benefit to the developer)

**Issues**
- Still responsible to keep software updated.
- Locked into provider API.
- Multi-tenancy at platform layer.

C. **Software as a Service** - In a SaaS model, the customer does not purchase software, but rather rents it for use on a subscription or pay-per-use model. Services provided by this layer can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionality. This model removes the burden of software maintenance for customers and simplifies development and testing for providers. Example Salesforce.com [10], which relies on the SaaS model, offers business productivity applications (CRM) that reside completely on their servers, allowing customers to customize and access applications on demand.

**Benefits**
- Scaling the environment is not the customer problem.
- Updates/configuration/security are all managed by the CSP.

**Issues**
- Very little application customization.
- No control of components.
- No control over security.
- Multi-tenancy issue at the application layer

---

**IV. Deployment Models**

There are various deployment models, with variations in physical location and distribution, which have been adopted by the cloud computing. A cloud can be classified as public, private, community, or hybrid based on model of deployment [16]. The majority of cloud computing infrastructure consists of reliable services delivered through data centers and built on servers with different levels of virtualization technologies. The services are accessible anywhere, the requirement is that access to networking infrastructure is available. All consumer computing needs are served at a single point known as cloud. The offered services should meet the quality of service requirements of customers and typically offer service-level agreements (SLAs) [25,26].

- Armbrust et al. propose definitions for **public cloud** as a “cloud made available in a pay-as-you-go manner to the general public” and **private cloud** as “internal data centre of a business or other organization, not made available to the general public.”
- A **community cloud** is “shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations)”.
- A **hybrid cloud** is composed of two or more clouds. The approach of temporarily renting capacity to handle spikes in load is known as “cloud-bursting”.

---

**Fig. 1 Service Layers of Cloud Computing**

**Cloud Clients**
Web browser, mobile app, thin client, terminal emulator, ...

**SaaS**
CRM, Email, virtual desktop, communication, games, ...

**PaaS**
Execution runtime, database, web server, development tools, ...

**IaaS**
Virtual machines, servers, storage, load balancers, network, ...
V. CONCEPTS

A powerful underlying and enabling concept is computing through service-oriented architectures (SOA) – delivery of an integrated and orchestrated suite of functions to an end-user through composition of both loosely and tightly coupled functions, or services – often network-based.

A. Service-oriented Architecture

The service-oriented architecture is a design model in which there is no restriction on who the customer will be. SOA is a framework but not a language.

It is an IT architecture for request-reply applications.

The service-oriented architecture approach is based on creating stand-alone, task-specific reusable software components that function and are made available as services [2].

“A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.”

Characteristics of SOA

- Services have platform independent, self-describing interfaces (XML).
- Messages are formally defined.
- Services can be discovered.
- Services have quality of service characteristics defined in policies.
- SOA is independent of programming language.
- Services are re usable.
- Loosely coupled.

B. Virtualization

Virtualization is another very useful concept in cloud computing. It allows abstraction and isolation of lower level functionalities and underlying hardware [11]. Virtualization refers to various techniques, methods or approaches of creating a virtual version of something such as a virtual hardware, platform, operating system, storage device, or network resources. It is the key technology that enables cloud computing. It enables aggregation of the physical resources. Virtualization is about separating the logical view of server assets from the physical resources upon which they run. By severing the tentacles that crept in between traditional operating systems and hardware, it enables virtual machines with hardware independence and mobility (among many other capabilities). Virtualization is the simulation of the software and/or hardware upon which other software runs. This simulated environment is called a virtual machine (VM).

Virtualization is a technique, method or approach of creating a virtual(not actual) version of something such as a virtual hardware platform, operating system (OS), storage device, or network resources.

VI. BENEFITS AND CHALLENGES OF CLOUD COMPUTING

A. Benefits

For IT buyers, investors and developers, the advantages of cloud platforms are tremendous. Creating a cloud application from the ground up is a complex process, involving not just ordinary coding, but also adding a layer of abstraction, and incorporating a far-flung communications layer as well as security protocols. If every SaaS provider had to create each of these things from scratch, then cloud-based application development would be hindered, and limited to only the larger software companies. Cloud platforms address this problem by allowing developers to build cloud applications on top of an existing architecture that includes core functionality. In essence, developers can use platforms to get their software to “80-yard line” without programming, and avoid reinventing the wheel.

The benefits are many:

- Lower costs – In some cases, a cloud platform can reduce costs by 80% or more, because non-core code is already engineered;


- **Lower risks** – Likewise, a cloud platform can reduce risks by as much, because common functions are already tested, sometimes over a period of years;
- **Faster time-to-market** – Cloud platforms dramatically reduce time-to-market, because they serve as a launch pad for software engineering efforts;
- **Higher profit margins** – Software developers and system integration firms can deliver more for substantially less, thus higher margins on fixed price contracts;
- **Rapid prototyping** – Create and deploy concept applications without writing code;
- **Higher security and interoperability** – NIST indicates that the cloud suffers from major security issues, largely because vendors are implementing disparate and unproven security models. Cloud platforms provide a common, proven security model. If cloud software uses the platform, then it is inherently secure.

**B. Challenges**

The following are some of the notable challenges associated with cloud computing, and although some of these may cause a slowdown when delivering more services in the cloud, most also can provide opportunities, if resolved with due care and attention in the planning stages. Most enterprise IT organizations have either implemented or are studying cloud projects. The two most commonly expressed fears [14] are:

- How do we keep our data safe?
- How do we prevent being locked in to a single vendor?
- How do we move legacy applications to the cloud?

**VI.B.1 Security Privacy and Trust** - Current cloud offerings are essentially public making the system vulnerable to attacks. Because of this vulnerability there are challenges to make cloud computing environments as secure as in-house IT systems. Well understood technologies can be leveraged, such as data encryption, VLANs, and firewalls in order to provide security. **Security and privacy** affect the entire cloud computing environment. Privacy implies security but not vice-versa. There is a massive use of third-party services and infrastructures so the third party provider will have control over large part of data, network, system, applications etc. In this scenario, there should be transparency between cloud service provider and consumer and the tools and technology that are being adopted should be transparent to the user.

The security mechanisms between organization and the cloud need to be robust and a Hybrid cloud could support such a deployment. Security management standards ITIL and ISO 27001/27002 are used to ensure that effective security measures are adopted.

**VI.B.2 Data Lock-In and Standardization** - The storage APIs for cloud computing are still essentially proprietary, or at least have not been the subject of active standardization. Thus, customers do not find it easy to extract their data and programs from one site to run on another. This concern of difficulty in extracting data from the cloud is preventing some organizations from adopting cloud computing. Customer lock-in makes the users vulnerable to price increases, to reliability problems, or even to providers going out of business however, it may be attractive to cloud computing providers. For example, “The Linkup” an online storage service shut down on Aug. 8, 2008 after losing access to about 45% of customer data. The Linkup’s 20,000 users were told the service was no longer available and were urged to try out another storage site. The **Cloud Computing Interoperability Forum (CCIF)** was formed by organizations such as Intel, Sun, and Cisco in order to “enable a global cloud computing ecosystem whereby organizations are able to seamlessly work together for the purposes of wider industry adoption of cloud computing technology.” The development of the **Unified Cloud Interface (UCI)** by CCIF aims at creating a standard programmatic point of access to an entire cloud infrastructure [18].

**VI.B.3 Bugs in Large-Scale Distributed Systems** - Removing errors in the very large-scale distributed systems is a difficult challenge. Debugging a large scale distributed system is extremely complex. These bugs cannot be reproduced in smaller configurations, so the debugging must occur at scale in the production data centers. Virtual machines can be used to handle the problem in cloud computing. Many traditional SaaS providers developed their infrastructure without using VMs, either because they preceded the recent popularity of VMs or because they felt they could not afford the performance hit of VMs.
Regardless of whether a cloud provider sells services at a low level of abstraction like EC2 or a higher level like AppEngine, we believe computing, storage, and networking must all focus on horizontal scalability of virtualized resources rather than on single node performance.

**REFERENCES**


[17] Cloud Computing – A Classification, Business Models, and Research Directions

[18] A View of Cloud Computing Clearing the clouds away from the true potential and obstacles posed by this computing capability. Communications of the ACM


VI.B.4 Business Continuity and Service Availability - Organizations worry about whether they will have adequate availability, and this makes the customers not adopt cloud computing. Existing SaaS products have set a high standard in service availability. Google Search has a reputation for being highly available, to the point that even a small disruption is picked up by major news sources.

Users expect similar availability from new services, which is difficult to do. Many services have suffered outage like Amazon Simple Storage Service (S3), AppEngine and Gmail in 2008. Despite the negative publicity due to these outages, few enterprise IT infrastructures are as good. Technical issues of availability aside, a cloud provider could suffer outages for non-technical reasons, including going out of business or being the target of regulatory action.

Cloud vendors could offer specialized hardware and software techniques in order to deliver higher reliability, may be at a high price. The high-availability computing community has long followed the mantra “no single point of failure,” yet the management of a computing service by a single company is in fact a single point of failure. Even if the company has multiple data centers in different geographic regions using different network providers, it may have common software infrastructure and accounting systems, or the company may even go out of business. Large customers will be reluctant to migrate to cloud computing without a business-continuity strategy for such situations. We believe the best chance for independent software stacks is for them to be provided by different companies, as it has been difficult for one company to justify creating and maintain two stacks in the name of software depend-ability. Just as large Internet service providers use multiple network providers so that failure by a single company will not take them off the air, we believe the only plausible solution to very high availability is multiple cloud computing providers.

VII. CONCLUSION

“Cloud Computing” sounds confusing. It sounds like a very fuzzy term [15] but have both advantages and disadvantages in migrating into cloud. There are two principle attributes of cloud computing: scalability and code reuse. We predict cloud computing will grow, so developers should take it into account.


[25] Best practices to develop SLAs for cloud computing IBM

[26] The Practical Guide for Service Level Agreements, published by the Cloud Standards Custom Council (CSCC), highlights the critical elements of a service level agreement for cloud computing and provides guidance on what to expect and what to be aware of when negotiating an SLA.