

Cloud Computing: Problems and Difficulties

Dr. Ranjan Kumar Mondal

Assistant Professor, Department of CSE, Swami
Vivekananda University, Kolkata, WB

ranjankm@svu.ac.in

Abstract: Many people think that the cloud will revolutionize the ICT sector. Our goal in this research is to identify the problems and obstacles associated with cloud computing. Following a discussion of two related computing paradigms – Service-Oriented Computing and Grid Computing – and how they relate to cloud computing, we list several obstacles to cloud computing adoption. Finally, we will draw attention to the Cloud interoperability problem, which warrants significant further study and development.

Keywords: distributed computing, cloud computing, web services, and service-oriented computing.

I. Introduction

Recently, cloud computing has become a popular term in the field of distributed computing. Many think the cloud will revolutionize and change the IT sector drastically. Describe cloud computing, please. What distinguishes it from Grid computing and service-oriented computing? What are the common problems and difficulties that affect both cloud service providers and users?

By providing answers to these questions, we want to clarify important research topics and outline potential paths and obstacles for cloud computing research in the future. We structure this article using an outside-in method to achieve this. Firstly, we look at many cloud apps that share numerous essential features. Next, we go over how cloud computing, Service-Oriented Computing (SOC), and Grid computing (also known as High-Performance Computing) are related. We provide a comparative analysis of these three computing

paradigms, emphasizing their mutual benefits. Next, we will go into cloud computing deployment and service concepts. We detail a cloud's deployment and service models, which opens the door to discussing several data-related problems and difficulties, including multi-tenancy, security, etc. Lastly, we talk about standards and interoperability problems.

II.CLOUD: SUMMARY

A. Definition

What is the meaning of cloud computing? Even though several formal definitions have been put out in academia and business, the U.S. The following critical common aspects seem to be included by NIST (National Institute of Standards and Technology) [1 and are extensively utilized in the Cloud Computing community:

The cloud computing model allows a shared pool of reconfigurable computing resources (such as networks, servers, storage, apps, and services) to be quickly provisioned and released with little management work or service provider interaction [1].

Security, deployment methodologies, and cloud architectures are all included in this description. Specifically, five fundamental components of cloud computing are explained in detail:

On-demand self-service: Without requiring human interaction with resource providers, a customer with an immediate need at a specific time can automatically (i.e., conveniently, self-serve) obtain computing resources (such as CPU time, network storage, software use, and so forth).

Wide-ranging network access: These computing resources are available to different client applications running on heterogeneous platforms (such as PDAs, laptops, and mobile phones) at the customer's location via a network (like the Internet).

pooling of resources. Using either the multi-tenancy or virtualization paradigm, a cloud service provider's computer resources are "pooled" together in an attempt to serve

numerous customers, “with different physical and virtual resources dynamically assigned and reassigned according to consumer demand” [1]. The establishment of a pool-based computing paradigm is driven by two significant factors: specialization and economies of scale. A pool-based paradigm makes actual computing resources (such as databases and CPUs) “invisible” to users, who often have no control or understanding over these resources’ location, composition, or unique features. Customers, for instance, cannot choose where their data will be kept in the cloud.

quick elasticity. Customers no longer have to worry about upfront contracts or commitments since they can utilize computer resources to scale up and down as needed, and they may release them once they’re done. This makes computing resources instantaneous rather than permanent. Furthermore, they believe resource provisioning is limitless, as consumption may spike to satisfy peak demand at any moment.

Service Measurement. Even though numerous users share and pool computer resources (also known as multi-tenancy), the cloud architecture may make use of the necessary technologies

To use its metering features to determine how these resources are used for every customer.

B. Service Model

The cloud community has widely used the following three service models, in addition to these five fundamental qualities, to categorize cloud services:

Cloud-based software (SaaS). Application users may access cloud-based applications over networks using a variety of clients, such as web browsers and PDAs, when cloud consumers release their apps in a hosting environment. The applications of various cloud consumers are arranged in a single logical environment on the SaaS cloud to achieve economies of scale and optimization in terms of speed, security, availability, disaster recovery, and maintenance. However, cloud consumers have no control over the cloud infrastructure, which frequently uses a multi-tenancy system

architecture. SaaS products include things like Google Docs, Salesforce.com, and Mail.

Software as a Platform (SaaS). Developers may create cloud services and apps (like SaaS) directly on the PaaS cloud thanks to PaaS, a platform for cloud development that supports the whole “Software Lifecycle.” Thus, the distinction between PaaS and SaaS is that the former provides a development platform that houses finished and ongoing cloud applications. At the same time, the latter exclusively hosts finished cloud applications. This means that PaaS has to provide development infrastructure, such as a programming environment, tools, configuration management, and other things, in addition to offering an application hosting environment. Google AppEngine is a PaaS example.

As a Service Infrastructure (IaaS). IT infrastructures (processing, storage, networks, and other essential computer resources) offered by the IaaS cloud are directly used by cloud users. To integrate and disassemble physical resources on demand to accommodate changing resource requirements from cloud users, virtualization is widely used in infrastructure as a service (IaaS) clouds. The fundamental virtualization method creates separate virtual machines (VMs) insulated from one another and the underlying hardware. This approach differs from the multi-tenancy paradigm, which seeks to change the architecture of the application software to enable the operation of numerous instances (from various cloud customers) on a single application (i.e., the same logic machine). EC2 from Amazon is an example of IaaS.

Data-as-a-service (DaaS) data storage. Data storage services are now distinct cloud services, providing virtualized storage on demand. Observe that DaaS may be thought of as a particular kind of IaaS. The driving force is the exorbitant upfront costs associated with dedicated servers, software licenses, post-delivery services, and internal IT maintenance for on-premise business database systems. Instead of paying for a site license covering the whole database, DaaS enables users to pay for their services. Some DaaS offerings offer table-style abstractions that can scale out to store and retrieve massive amounts of data quickly, in addition to traditional

storage interfaces like file systems and RDBMS. These abstractions are frequently too large, costly, or slow for most commercial RDBMS to handle. As an example

Here, indicate which sponsor(s) applies. (Partners)

Examples of this kind of DaaS include Apache HBase, Google BigTable, and Amazon S3.

C.Model of Deployment

The cloud community has recently established our cloud deployment methods:

Private cloud. Whether housed on-site or off-site, the organization or a third party manages and administers the cloud infrastructure. It is only used by one organization. There are several reasons why a company would want to build up a private cloud. Initially, the goal is to use the current internal resources as best as possible. Second, many businesses choose the private cloud due to security concerns about data protection and trust. Third, a significant expense is still associated with data transfer [2] from local IT infrastructure to a public cloud. Fourth, mission-critical operations that take place behind firewalls must always be entirely under the authority of organizations. Finally, academics often create private clouds for use in teaching and research.

Cloud community. The cloud infrastructure is cooperatively built by several enterprises, sharing the same rules, needs, values, and concerns. A certain level of democratic balance and economic scalability is formed within the cloud community. One of the community groups or a third-party provider might host the cloud infrastructure.

Public cloud. This is the deployment model currently in use for cloud computing. Cloud service providers possess all rights to the public cloud, which is utilized by the general public and has its policies, value, profit, pricing, and billing structure. Public clouds include Google AppEngine, Amazon EC2, S3, and Force.com, among many other well-known cloud services.

cloud hybrid. A collection of two or more clouds—private, communal, or public—that are each separate entities but are connected by proprietary or standardized technology that allows data and application mobility (e.g., cloud bursting for load-balancing across clouds) makes up the cloud infrastructure. via the margining out of peripheral business tasks into the cloud and managing critical activities on-premise via the private cloud; organizations use the hybrid cloud model to maximize their resources and improve their core capabilities. A hybrid cloud has brought up the topics of standardization and cloud interoperability, which will be covered in the following parts.

Amazon Web Services (AWS) has introduced an attractive new deployment option: the Virtual Private Cloud (VPC). It is a safe and smooth link between an organization's on-premises IT infrastructure and the Amazon public cloud. This is presented as a hybrid of the public and private clouds. It continues to utilize computational resources that Amazon has pooled for public use, which makes it public. It is, nonetheless, essentially private for two reasons. First, a virtual private network provides security for the link between IT heritage and the cloud, offering the benefit of private cloud security. Even if the cloud resources are on the Public Cloud, all company security standards apply to them.

Secondly, the VPC will get “isolated” resources from AWS. This does not, however, imply that consumers must pay for these exclusive resources in advance. “Pay-per-use” is still an option for users on these remote sites. VPC is the ideal combination of freedom (Public Cloud) and control (Private Cloud).

Observe that the deployment and service models are perpendicular. SaaS, for instance, may be provided on a private or public cloud.

III.GRID, SOC, AND CLOUD

This section delineates the connections between Cloud Computing, Service-Oriented Computing (SOC), and Grid Computing.

A. Service-oriented computing and the cloud

SOC offers significant benefits in encapsulation, componentization, decentralization, and integration: they provide software specifications and architectural guidelines for connecting computers and devices via standardized protocols over the Internet [3]. The idea of the cloud is mainly derived from how SOC—specifically, the SaaS service model—is developing.

SOC developments may help cloud computing in several ways.

This is an explanation of the cloud services' services. Two popular interface languages for describing Web services are Web Services Description Language (WSDL) and the REST protocol. They have been applied to the description of the Cloud API standard.

Cloud Service Service Discovery. Service discovery models may be used for cloud resource selection, discovery, and service-level agreement verification.

Cloud Service Composition Service. Web services are designed to create business applications; therefore, much of this study may be used to integrate, compose, and cooperate with cloud services.

Cloud service management. Research and methods from SOA governance and services administration can be Adapted and reused for cloud infrastructure management.

However, we also need to consider what SOC lacks, particularly for small and medium-sized businesses. From the standpoint of business process and integration, SOC denotes high abstraction. SOC does not, however, provide any applicable computational models for operating services. How can I manage my services for the least amount of money? How to scale up/down my service-oriented architecture-based apps. The strain for SOC developers and the IT department in SMEs is increased by the need to handle these computational aspects on an as-needed basis and in a project-specific way. Furthermore, there is still debate in SOC on integrating services at various levels into a cohesive organizational

unit. For instance, how can I assist my business services by maximizing my IT services' utilization? Thus, we think that there are several significant ways in which cloud computing might advance research in service-oriented computing.

Web service development using the cloud. Under the PaaS service deploy paradigm, cloud computing may support service-oriented development. Distributed computing resources are often needed for SOC development, but they might be difficult for SMEs. For example, Google's AppEngine platform, SDK, and client IDE provide developers with a complete development environment to create and implement Java Web services to develop apps. Furthermore, the Yahoo Pipe platform demonstrates how the cloud can be used as both design and run-time for service composition and mashups.

Cloud for Web Service Testing. To load test and stress test services, web services developers might use the limitless processing power of a public cloud to mimic automated machine requests, and network flows seen in the real world. The difficulty and expense of simulating network traffic for WS testing has hampered the general dependability of the Web. It is now possible to replicate real-world usage of these systems by geographically dispersed users, executing a wide variety of user scenarios at scales previously unachievable in traditional testing environments thanks to the low cost and accessibility of the Cloud's vast computing resources.

They are using the cloud to deploy web services. Web services implementation may be expedited using IaaS. For instance, service providers may distribute their products using the Amazon Machine Image (AMI) in the Amazon EC2 configuration. A service deployment image will be loaded into a designated virtual machine in response to queries to fulfill client demands. When Web services are restarted from the suspended mode, stateful information generated during service interactions may also be persistently maintained on the AMI (e.g., from a long-last transaction).

Cloud for Enacting Service Processes. Service composition and integration are growing issues; cloud environments may bundle solutions for these issues as services. To enable the

reuse of ready-to-use solutions with minimal configuration and composition patterns, a popular strategy is to use the power of the crowd (service users) utilizing a variety of algorithms (e.g., Case-Based Reasoning).

An intriguing subject is how to integrate cloud computing with SOA/SOC. Are they on the same business/technical level? Do they have the same objective in mind? Are they able to work two jobs at once? If so, how? These research issues may be resolved when cloud computing becomes more widely used and developed.

B.Grid and Cloud Computing

A hardware and software framework known as “grid computing” [4] was inspired by actual issues that arose in cutting-edge scientific research. As far as we can tell, the Grid is “middleware” for distributed computing that gives sophisticated computational applications like research and engineering “coordinated cross-organizational resource sharing.” There are many parallels between grid computing and cloud computing. To start with, their shared goal is resource virtualization. They do, nonetheless, vary greatly:

Grid stresses “resource sharing” to establish a virtual organization. Except for community clouds, which are owned by the community, cloud computing is typically owned by a single physical entity that distributes resources across many currently used instances.

Grid uses resource sharing to deliver the most processing power possible for a large operation. Cloud computing seeks to meet as many small-to-medium-sized job needs as feasible in response to consumers’ immediate needs. As a result, the idea of multi-tenancy is crucial to cloud computing.

Grid gives up reusability in favor of high-performance (scientific) computation. Users’ current demands drive cloud computing, driven by different corporate objectives.

A.Safety

The biggest obstacle to cloud computing has been the security problem. Undoubtedly, many people find storing their data

and executing their program on someone else's hard drive and CPU intimidating. Well-known security problems, including data loss, phishing, and botnets—remotely operating networks of machines—pose significant risks to the software and data of a business. Furthermore, new security concerns have been presented by the multi-tenancy model and the pooled computing resources in cloud computing [6], which call for the development of innovative solutions. For instance, since the cloud often offers more dependable infrastructure services at a comparatively lower cost for them to launch an attack, hackers intend to utilize it to create botnets [6].

Q: How would you rank the problems with the “cloud” and on-demand models?

One denotes insignificance, and five have great significance.

Grid wants to do as much computation as possible. On-demand computing is what the cloud is all about: it allows you to scale up and down, in and out, while maximizing your computer's total capacity.

It was designed To provide business application Services. While parallelization has been used in high-performance computing (HPC), it is still challenging to use parallelization computing methodologies for business applications in Cloud computing due to the complex state and data dependencies across multiple business apps. The existing Cloud is not optimized for HPC for several reasons, according to authors in [5]. The first is that it is not yet developed sufficiently for HPC. Second, cloud architecture prioritizes improving overall system performance, unlike cluster computing. Third, HPC uses resources from several organizations to improve a particular scientific application's performance. The main distinction is elasticity, however, since capacity in cluster computing is often set, necessitating significant human intervention to execute HPC applications (e.g., tuning based on a specific cluster with a defined number of homogeneous compute nodes). This is in sharp contrast to cloud computing's “self-service” model, where we often don't know in advance how many physical processors we'll need or have already utilized.

IV.ADOPTION OF CLOUD CHALLENGES

Since cloud computing is still in its early stages of adoption, many obstacles remain.



Figure 1

Illustrates the main barriers enterprises face in adopting Cloud Computing, as identified by an IDC report 2008.

Figure 1. Adoption Difficulties (Source: August 2008 IDC Survey)

The multi-tenancy concept has brought forth at least two additional security risks. First, shared resources on the same physical computer (hard drive, data, virtual machine) attract unanticipated side channels between malicious and normal resources. Furthermore, the problem of “reputation fate-sharing” will seriously harm the standing of several law-abiding Cloud “citizens” who regrettably share computing resources with a fellow tenant who is an infamous user with a criminal past. Any inappropriate behavior will be ascribed to all users because they can have the same network address, making it impossible to distinguish between actual subversives and regular users.

B.Pricing Scheme

Cloud computing users must weigh the integration, connectivity, and processing trade-offs. Although moving to the cloud can significantly reduce infrastructure costs, there

are costs associated with data communication or moving an organization's data to and from public and community clouds [7]. Additionally, the cost of computing resources used per unit, such as a virtual machine (VM), will likely increase. This issue is most noticeable if the customer deploys a hybrid cloud, which divides the company's data between many public and private (in-house IT infrastructure)

cloud communities. Gray's [8] maxim "Put the computation near the data" is still relevant in cloud computing. On-demand computing seems to make sense only for tasks requiring much CPU power. Put differently, transactional apps like ERP/CRM may not be appropriate for cloud computing from a

A pure economic perspective is needed if the cost savings do not cover the additional data transport costs.

Additionally, data integration may be expensive since separate clouds often utilize proprietary protocols and interfaces. To transfer and integrate diverse resources and data assets to and from different clouds (even within a single business), the cloud consumer must create ad-hoc adaptors and use cloud provider-specific APIs to communicate with multiple clouds. For instance, to address the security issue, cloud users (like the Eli Lilly research lab [9]) might need to divide sensitive data (like the medication usage for each patient) into smaller portions and distribute them across multiple clouds to prevent a security compromise in one cloud from causing a catastrophe as a whole. However, dividing and combining data may hurt system performance (i.e., time cost) and add significant financial expense.

C. Model of Charging

In contrast to traditional data centers, which often base their cost calculations on static computer consumptions, cloud providers find that their cost analysis is much more complex due to the elastic resource pool (achieved via virtualization or multi-tenancy). Furthermore, the underlying physical server is no longer the unit of cost analysis—instead, it is an instantiated virtual machine. All of the elements above, as well as VM-related costs like software licensing, virtual network utilization, node and hypervisor administrative overhead,

and so forth, must be included in a solid billing model.

Creating multitenancy inside a SaaS cloud solution may be pretty expensive for providers. These include developing new features that enable extensive customization, improving speed and security for concurrent user access, redesigning and redeveloping the software initially used for single-tenancy, and managing the complications brought on by the modifications above. As a result, SaaS providers must balance the trade-off between offering multitenancy and the cost reductions that come with it, including fewer on-site software licensing, lower overhead from amortization, etc. Thus, SaaS cloud providers' profitability and long-term viability depend on a well-thought-out, workable pricing strategy.

D. Service Level Contract

Once customers have moved their essential business processes onto their trusted cloud, they still need to guarantee these resources' quality, availability, dependability, and performance, even if they have no control over the underlying computer resources. As stated differently, customers must get assurances of the provision of services from suppliers. These are usually supplied via Service Level Agreements (SLAs) arranged between the suppliers and the customers. The first problem is defining SLA specifications with a suitable level of detail, i.e., balancing expressiveness and complexity to meet most customer expectations, and it is reasonably easy for the cloud resource allocation mechanism to weigh, verify, evaluate, and enforce. Furthermore, several cloud services (IaaS, PaaS,

Different SLA meta-specifications will need to be defined for SaaS and DaaS.

For the cloud providers, this also presents several implementation-related issues. Resource managers, for instance, need accurate and up-to-date data on resource use in the cloud at any given moment. When we refer to updated information, we mean that any modifications made to the cloud environment would cause the resource management to subscribe to an event, allowing for real-time assessment and adjustment to fulfill SLAs. To achieve this, resource managers

must use optimization algorithms and quick and accurate decision models. If SLAs are unsatisfied, they could have to deny specific resource requests. Given the promise of “self-service” in cloud computing, all of these must be completed almost entirely automatically. Additionally, user input and customization features must be continuously included in the SLA assessment framework using sophisticated SLA processes.

E. Things to move

The seven IT systems/applications being moved to the cloud are: IT Management Applications (26.2%), Collaborative Applications (25.4%), Personal Applications (25%), Business Applications (23.4%), Applications Development and Deployment (16.8%), Server Capacity (15.6%), and Storage Capacity (15.5%), according to a survey (Sample size = 244) carried out by IDC in 2008. This finding indicates that enterprises still hesitate to move their data to the cloud due to security and privacy concerns. The most accessible IT systems to migrate are peripheral tasks like personal apps and IT management. Compared to SaaS, organizations use IaaS more cautiously. This is partly because core operations are maintained in-house while peripheral duties are often contracted to the cloud. According to the poll, 31.5% of the company plans to shift its storage capacity to the cloud in the next three years. Nonetheless, this figure remains relatively low when contrasted with Collaborative Applications (46.3%) over the same period.

V. INTEROPERABILITY CLOUD ISSUE

The “Hazy Cloud” phenomenon results from the fact that every cloud provider now provides a unique method for cloud clients, apps, and users to interact with the cloud [10]. Vendor lock-in, which prevents customers from selecting from several suppliers or offers simultaneously to maximize resources at various organizational levels, seriously impedes the growth of cloud ecosystems. Integrating cloud services with an organization’s legacy systems—such as an on-premise data center for highly interactive modeling applications in a pharmaceutical company—becomes very

challenging when private cloud APIs are used. Here, the term “interoperability” refers to the connections between several clouds and the internal systems of an organization. Realizing smooth-flowing data between clouds and between cloud and local applications is the main objective of interoperability.

Interoperability is required for cloud computing on many levels. First, to maximize computer resources and IT assets, a business often has to maintain

Internal IT resources and competencies related to their core competencies while contracting out other tasks and services (like the HR system) to the cloud. In this scenario, regular communication between on-premise systems, such as an ERP system, and cloud services, like the HR system, becomes essential to operating a firm. Inadequate interoperability, such as proprietary APIs and excessively intricate or unclear data formats used by an HR cloud SaaS, can significantly worsen integration challenges, placing a challenging position for the IT department. Second, in most cases, a company must outsource several marginal operations to cloud services provided by various suppliers to optimize. For instance, an SME may utilize Salesforce.com for HR services and Gmail for email. This implies that the HR employee directory located in the HR system has to be connected to the various elements (such as the address book, calendar, appointment booking, etc.) in the email system.

A.Middle Layer

Some recent works provide an intermediate layer between cloud users and cloud-specific resources (such as virtual machines) to solve the interoperability problem. For instance, to support different clouds, Sotomayor et al. [11] suggested using Virtual Infrastructure (OpenNebula) Management to replace native VM API interactions.

- private or mixed for a company. Because Open Nebula operates at the virtualization layer, cloud users may access a uniform view and operating interfaces for different underlying virtualization implementations. In contrast to Open sNebula, Harmer et al. [12] created a higher-level abstraction layer. This shields the Heterogeneity of cloud providers and

impedes the creation of cloud-provider-independent apps by offering a standard resource consumption paradigm, user authentication methodology, and API.

B.Normal

Standardization seems to be a reasonable way to deal with the interoperability problem. However, as cloud computing is still starting, most industrial cloud suppliers have not listed the interoperability issue as a top priority. For instance, the Unified Cloud Interface (UCI) Project, which was put out by the Cloud Computing Interoperability Forum (CCIF), is not supported by either Microsoft or Amazon [13]. Moving the standardization process ahead will be tough if these significant actors do not compromise. The Eucalyptus project [14], which replicates the well-known proprietary Amazon EC2 API for cloud operation, is a popular cloud API in academics. While consumers of Eucalyptus IaaS may connect to the EC2 cloud with ease and without requiring significant redevelopment, this does not address the broader interoperability problem that necessitates an open API that is adhered to by many cloud providers.

C.Accessible API

Recently, SUN introduced the Sun Open Cloud Platform.

[15] with permission of the Creative Commons. This platform significantly contributes to its planned (still-in-progress) cloud API. It outlines several distinct and simple-to-understand

Cloud users may develop and manage cloud resources, such as computing, storage, and networking components, in an integrated manner using RESTful Web services interfaces. The open cloud API specifies major resource types: Cloud, Virtual Data Center, Cluster, Virtual Machine, Private Virtual Network, Public Address, Storage Volume, and Volume Snapshot. It uses HTTP as the application protocol and JSON as the resource representation language. There is a resemblance between these constructions and Eucalyptus's interior architecture [14]. The Eucalyptus team is prepared to put in the effort to guarantee that Eucalyptus Clouds and the Sun Cloud API are compatible [15]. This is consistent

with the ongoing research being conducted by DEBII to provide a lightweight PaaS open API that leverages RESTful Web services. It is worth noting that the idea of a Virtual Data Center, which serves as the fundamental unit for implementing the Sun Open Cloud, is similar to the Virtual Private Cloud concept recently established in Amazon EC2 (refer to Section II - C).

D. Interoperability between SaaS and PaaS

Although the methods discussed above primarily address IaaS interoperability issues, not much research has been done on other service deployment models. SaaS interoperability often encompasses many application areas, like CRM, ERP, etc. The data mining research community is one area in which our research group at DEBII is particularly interested. A group of data mining experts discussed creating a cloud-based data mining standard in the recent KDD09 panel discussion [16]. They specifically focused on “the practical use of statistical algorithms, reliable production deployment of models, and the integration of predictive analytics” among various SaaS clouds that use data mining. The Predictive Model Markup Language (PMML), a widely adopted standard that enables users to communicate predictive models between different software applications, is a promising step in this direction.

No significant attempts have been made to provide PaaS compatibility. It would be more challenging to achieve consistency in how users create and deploy cloud applications since PaaS encompasses the whole software development lifecycle on the cloud.

VI. Conclusion

This essay covered the problems and difficulties associated with cloud computing. We explained how cloud computing, grid computing, and service-oriented computing relate. We examined a few obstacles to the adoption of cloud computing. After highlighting the interoperability problem, we covered many solutions for various cloud service deployment methods.