Cloud Computing: Issues and Challenges

Tharam Dillon
Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology
Perth, Australia
tharam.dillon@curtin.edu.au

Chen Wu and Elizabeth Chang
Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology
Perth, Australia
{chen.wu, change}@curtin.edu.au

***Abstract*— Many believe that Cloud will reshape the entire ICT industry as a revolution. In this paper, we aim to pinpoint the challenges and issues of Cloud computing. We first discuss two related computing paradigms - Service-Oriented Computing and Grid computing, and their relationships with Cloud computing We then identify several challenges from the Cloud computing adoption perspective. Last, we will highlight the Cloud interoperability issue that deserves substantial further research and development.**

Keywords: Cloud computing; Servcice-Oriented Computing; Distributed Comptuing; Web Services

I. INTRODUCTION

Cloud computing has recently emerged as a buzz word in the distributed computing community. Many believe that Cloud is going to reshape the IT industry as a revolution. So, what is Cloud Computing? How is it different from service-oriented computing and Grid computing? What are those general challenges and issues for both cloud providers and consumers?

In answering these questions, we aim to define key research issues and articulate future research challenges and directions for cloud computing. To do this, we take an outside-in approach to organize this paper. We first examine a number of cloud applications that exhibit several key characteristics. We then discuss the relationship between Cloud computing and Service-Oriented Computing (SOC) and the relationship between Cloud and Grid computing (i.e. High-Performance Computing). We compare these three computing paradigms and draw attention to how they will benefit each other in a co-existent manner. Next, we discuss service models and deployment models of cloud computing. We elaborate service model and deployment model of a cloud, which leads to the discussion of several data-related issues and challenges such as multi-tenancy, security, and so forth. Finally, we discuss interoperability and standardization issues.

II. CLOUD: OVERVIEW

A. Definition

What is Cloud Computing? Although many formal definitions have been proposed in both academia and industry, the one provided by U.S. NIST (National Institute of Standards and Technology) [1] appears to include key common elements widely used in the Cloud Computing community:

Cloud computing is 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 [1].

This definition includes cloud architectures, security, and deployment strategies. In particular, five essential elements of cloud computing are clearly articulated:

On-demand self-service: A consumer with an instantaneous need at a particular timeslot can avail computing resources (such as CPU time, network storage, software use, and so forth) in an automatic (i.e. convenient, self-serve) fashion without resorting to human interactions with providers of these resources.

Broad network access: These computing resources are delivered over the network (e.g. Internet) and used by various client applications with heterogeneous platforms (such as mobile phones, laptops, and PDAs) situated at a consumer's site.

Resource pooling. A cloud service provider's computing resources are 'pooled' together in an effort to serve multiple consumers using either the *multi-tenancy* or the *virtualization* model, "with different physical and virtual resources dynamically assigned and reassigned according to consumer demand" [1]. The motivation for setting up such a pool-based computing paradigm lies in two important factors: *economies of scale* and *specialization*. The result of a pool-based model is that physical computing resources become 'invisible' to consumers, who in general do not have control or knowledge over the location, formation, and originalities of these resources (e.g. database, CPU, etc.) . For example, consumers are not able to tell where their data is going to be stored in the Cloud.

Rapid elasticity. For consumers, computing resources become immediate rather than persistent: there are no up-front commitment and contract as they can use them to scale up whenever they want, and release them once they finish to scale down. Moreover, resources provisioning appears to be infinite to them, the consumption can rapidly rise in order to meet peak requirement at any time.

Measured Service. Although computing resources are pooled and shared by multiple consumers (i.e. multi-tenancy), the cloud infrastructure is able to use appropriate mechanisms

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.187

to measure the usage of these resources for each individual consumer through its metering capabilities.

B. Servcice Model

In addition to these five essential characteristics, the cloud community has extensively used the following three service models to categories the cloud services:

Software as a Service (SaaS). Cloud consumers release their applications on a hosting environment, which can be accessed through networks from various clients (e.g. web browser, PDA, etc.) by application users. Cloud consumers do not have control over the Cloud infrastructure that often employs a multi-tenancy system architecture, namely, different cloud consumers' applications are organized 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. Examples of SaaS include SalesForce.com, Google Mail, Google Docs, and so forth.

Platform as a Service (PaaS). PaaS is a development platform supporting the full "Software Lifecycle" which allows cloud consumers to develop cloud services and applications (e.g. SaaS) directly on the PaaS cloud. Hence the difference between SaaS and PaaS is that SaaS only hosts completed cloud applications whereas PaaS offers a development platform that hosts both completed and in-progress cloud applications. This requires PaaS, in addition to supporting application hosting environment, to possess development infrastructure including programming environment, tools, configuration management, and so forth. An example of PaaS is Google AppEngine.

Infrastructure as a Service (IaaS). Cloud consumers directly use IT infrastructures (processing, storage, networks, and other fundamental computing resources) provided in the IaaS cloud. Virtualization is extensively used in IaaS cloud in order to integrate/decompose physical resources in an ad-hoc manner to meet growing or shrinking resource demand from cloud consumers. The basic strategy of virtualization is to set up independent virtual machines (VM) that are isolated from both the underlying hardware and other VMs. Notice that this strategy is different from the multi-tenancy model, which aims to transform the application software architecture so that multiple instances (from multiple cloud consumers) can run on a single application (i.e. the same logic machine). An example of IaaS is Amazon's EC2.

Data storage as a Service (DaaS). The delivery of virtualized storage on demand becomes a separate Cloud service - data storage service. Notice that DaaS could be seen as a special type IaaS. The motivation is that on-premise enterprise database systems are often tied in a prohibitive up-front cost in dedicated server, software license, post-delivery services, and in-house IT maintenance. DaaS allows consumers to pay for what they are actually using rather than the site license for the entire database. In addition to traditional storage interfaces such as RDBMS and file systems, some DaaS offerings provide table-style abstractions that are designed to scale out to store and retrieve a huge amount of data within a very compressed timeframe, often too large, too expensive or too slow for most commercial RDBMS to cope with. Examples of this kind of DaaS include Amazon S3, Google BigTable, and Apache HBase, etc.

C. Deployment Model

More recently, four cloud deployment models have been defined in the Cloud community:

Private cloud. The cloud infrastructure is operated solely within a single organization, and managed by the organization or a third party regardless whether it is located premise or off premise. The motivation to setup a private cloud within an organization has several aspects. First, to maximize and optimize the utilization of existing in-house resources. Second, security concerns including data privacy and trust also make Private Cloud an option for many firms. Third, data transfer cost [2] from local IT infrastructure to a Public Cloud is still rather considerable. Fourth, organizations always require full control over mission-critical activities that reside behind their firewalls. Last, academics often build private cloud for research and teaching purposes.

Community cloud. Several organizations jointly construct and share the same cloud infrastructure as well as policies, requirements, values, and concerns. The cloud community forms into a degree of economic scalability and democratic equilibrium. The cloud infrastructure could be hosted by a third-party vendor or within one of the organizations in the community.

Public cloud. This is the dominant form of current Cloud computing deployment model. The public cloud is used by the general public cloud consumers and the cloud service provider has the full ownership of the public cloud with its own policy, value, and profit, costing, and charging model. Many popular cloud services are public clouds including Amazon EC2, S3, Google AppEngine, and Force.com.

Hybrid cloud. The cloud infrastructure is a combination of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds). Organizations use the hybrid cloud model in order to optimize their resources to increase their core competencies by margining out peripheral business functions onto the cloud while controlling core activities on-premise through private cloud. Hybrid cloud has raised the issues of standardization and cloud interoperability that will be discussed in later sections.

Interestingly, Amazon Web Services (AWS) has recently rolled out a new type of deployment model - *Virtual Private Cloud (VPC)*, a secure and seamless bridge between an organization's existing IT infrastructure and the Amazon public cloud. This is positioned as a mixture between Private Cloud and Public Cloud. It is Public because it still uses computing resources pooled by Amazon for the general public. However, it is virtually private for two reasons. Firstly, the connection between IT legacy and the cloud is secured through a virtual private network, thereby having the security advantage of Private Cloud. In fact, all corporate security policies still apply to resources on the cloud even though it is on the Public Cloud.

Identify applicable sponsor/s here. *(sponsors)*

Second, AWS will dedicate a set of 'isolated' resources to the VPC. However, this does not mean users have to pay these isolated resources up-front. Users still enjoy "pay-per-use" on these isolated resources. VPC represents a perfect balance between control (Private Cloud) and flexibility (Public Cloud).

Notice that the service model is orthogonal to the deployment model. For example, an SaaS could provisioned on a Public cloud or Private cloud.

III. CLOUD, SOC, AND GRID

In this section, we identify relationships between Cloud Computing, Service-Oriented Computing (SOC) and Grid Computing.

A. Cloud and Service-Oriented Computing

The encapsulation, componentization, decentralization, and integration capability provided by SOC are substantial: they provide both architectural principles and software specifications to connect computers and devices using standardized protocols across the Internet [3]. In fact, the notion of Cloud is more or less based on the evolving development on SOC, in particular the SaaS service model.

Advances in SOC can benefit Cloud Computing in several ways.

Service Description for Cloud Services. Web Services Description Language (WSDL) and the REST protocol are two widely used interface languages to describe Web services. They have been utilized to describe Cloud API specification.

Service Discovery for Cloud Services. Various service discovery models can be leveraged for cloud resource discovery, selection and service-level agreement verification.

Service Composition for Cloud Service. Since Web services are born to compose business applications, a great deal of research in this area can be leveraged for cloud services integration, collaboration, composition.

Service Management for Cloud Service. Research and practices in SOA governance and services management can be adapted and reused in the cloud infrastructure management.

On the other hand, we need to consider what is missing in SOC, especially from the perspective of small and medium enterprises? SOC represents a high level of abstraction from the integration and business process perspective. However, SOC does not provide a practical computational models for running services. For example, how to run my services with minimum cost? How to scale in/out my applications built on top of Service-Oriented Architecture. These computational details have to be dealt with in a project-specific and ad-hoc manner, which burdens the workload for SOC developers and IT department in SMEs. In addition, how to include services at different levels into a coherent organizational entity is an open question in SOC. For example, how to maximize the utilization of my IT services in order to support my business services? Therefore, we believe Cloud computing can benefit Service-Oriented Computing research in several important ways.

Cloud for Web Service Development. Cloud can host service-oriented development under the PaaS service deploy model. SOC development often requires distributed computing resources that are difficult to obtain for SMEs. For example, Google's AppEngine (the platform, its SDK, and client IDE) provides a full-fledged development platform in which developers can develop and deploy Java Web services to build their applications. In addition, the Yahoo Pipe platform illustrates the potential that Cloud can serve as the design-time and run-time for service Mashups and Composition.

Cloud for Web Service Testing. Web services developers could tap into infinite computing resources in a Public Cloud to simulate real-world automated machine requests and network flows as a means of load testing and stress testing for services. The ability and cost to simulate network traffic for WS testing has been an inhibitor to overall Web reliability. The low cost and accessibility of the Cloud’s extremely large computing resources provides the ability to replicate real world usage of these systems by geographically distributed users, executing wide varieties of user scenarios, at scales previously unattainable in traditional testing environments.

Cloud for Web Service Deployment. Using IaaS, Web services deployment can be streamlined. For example, under the Amazon EC2 setting, service deployers can use the Amazon Machine Image (AMI) to distribute their offerings. When requests are present, a service deployment image will be loaded into a specified virtual machine to serve the client requests. Stateful information produced during service interactions can be also kept persistent onto the AMI when Web services are resumed from the suspended mode (e.g. from a long-last transaction).

Cloud for Service Process Enactment. The integration and composition of services become frequent problems and their solutions can be packaged as services deployed in the cloud environment. Therefore, a prevailing approach is to exploit the power of crowd (service users) to allow the re-use of solutions that are ready-to-use with minor configuration and composition patterns using various algorithms (e.g. Case-Based Reasoning).

The integration question of Cloud and SOA/SOC is an interesting one. Are they at the same technical/business level? Do they aim to achieve the same goal? Can they be employed at the same time? If so, how? These are research challenges that can be addressed with the further development and adoption of cloud computing.

B. Cloud and Grid Computing

Grid computing [4] is a hardware and software infrastructure motivated by real problems appearing in advanced scientific research. To our understanding, the Grid is distributed computing ‘middleware’ that provides ‘coordinated cross-organizational resource sharing’ to high-end computational applications such as science and engineering. There exists evident similarities between Cloud Computing and Grid Computing. For one thing, they both aim to achieve *resource virtualization*. However, they do have significant differences:

- Grid emphasizes the “resource sharing” to form a virtual organization. Cloud is often owned by a single physical organization (except the community Cloud, in this case, it is owned by the community), who allocates resources to different running instances.
- Grid aims to provide the maximum computing capacity for a huge task through resource sharing. Cloud aims to suffice as many small-to-medium tasks as possible based on users’ real-time requirements. Therefore, multi-tenancy is a very important concept for Cloud computing.
- Grid trades re-usability for (scientific) high performance computing. Cloud computing is directly pulled by immediate user needs driven by various business requirements.
- Grid strives to achieve maximum computing. Cloud is after on-demand computing – Scale up and down, in and out at the same time optimizing the overall computing capacity.

C. *Cloud and High Performance Computing*

High-performance computing (HPC) aims to leverage supercomputers and computer clusters to solve advanced (scientific) computation problems. The original intent of Cloud computing and HPC can be evidently different, which yields different computing paradigms as well as applications. While HPC has been widely used for scientific tasks Cloud computing was set out for serving business applications. Whereas parallelization has been fully exploited in HPC, the highly complicated state and data dependencies amongst many business applications have made more difficult to leverage parallelization computing approaches for business applications in Cloud computing. Authors in [5] have pointed out that the current Cloud is not geared for HPC for several reasons: First, it is not yet matured enough for HPC. Second, unlike Cluster computing, Cloud infrastructure focuses on enhancing the overall system performance as a whole. Third, HPC aims to enhance the performance of a specific scientific application using resources across multiple organizations. But the key difference lies in elasticity: for cluster computing, the capacity is often fixed, therefore running an HPC application often require considerable human interaction (e.g. tuning based on a particular cluster with a fixed number of homogenous computing nodes). This is in stark contrast with the "self-service" nature of cloud computing, in which we often do not know a-prior how many physical processors do we need or have we used.

IV. Cloud Adoption Challenges

As Cloud Computing is still in its infancy, current adoption is associated with numerous challenges. Based on a survey conducted by IDC in 2008, the major challenges that prevent Cloud Computing from being adopted are recognized by organizations as shown in Figure 1.

A. *Security*

It is clear that the security issue has played the most important role in hindering Cloud computing. Without doubt, putting your data, running your software at someone else's hard disk using someone else's CPU appears daunting to many. Well-known security issues such as data loss, phishing, botnet (running remotely on a collection of machines) pose serious threats to organization's data and software. Moreover, the multi-tenancy model and the pooled computing resources in cloud computing has introduced new security challenges [6] that require novel techniques to tackle with. For example, hackers are planning to use Cloud to organize botnet as Cloud often provides more reliable infrastructure services at a relatively cheaper price for them to start an attack [6].

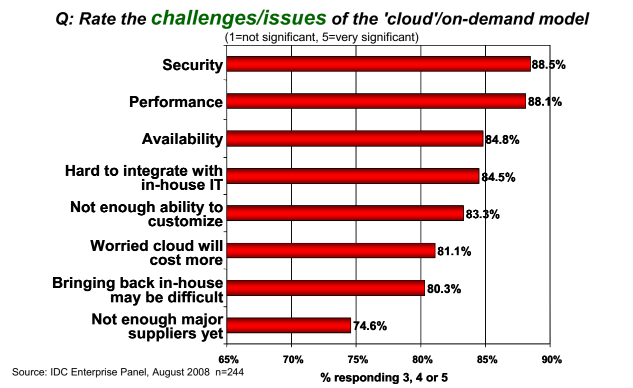

Figure 1. Adoption Challenges (Source: IDC Survey, Aug 2008)

The multi-tenancy model has at least created two new security issues. First, shared resources (hard disk, data, VM) on the same physical machine invites unexpected side channels between a malicious resource and a regular resource. Second, the issue of "reputation fate-sharing" will severely damage the reputation of many good Cloud "citizens" who happen to, unfortunately, share the computing resources with their fellow tenant - a notorious user with a criminal mind. Since they may share the same network address, any bad conduct will be attributed to all the users without differentiating real subverters from normal users.

B. *Costing Model*

Cloud consumers must consider the tradeoffs amongst computation, communication, and integration. While migrating to the Cloud can significantly reduce the infrastructure cost, it does raise the cost of data communication, i.e. the cost of transferring an organization's data to and from the public and community Cloud [7] and the cost per unit (e.g. a VM) of computing resource used is likely to be higher. This problem is particularly prominent if the consumer uses the hybrid cloud deployment model where the organization's data is distributed amongst a number of public/private (in-house IT infrastructure) /community clouds. The argument made by Gray [8] that "Put the computation near the data" still applies in cloud computing. Intuitively, on-demand computing makes sense only for CPU-intensive jobs. In other words, transactional applications such as ERP/CRM may not be suitable for cloud computing from a pure economic view if the cost-saving do not offset the extra data transfer cost.

In addition, the cost of data integration can be substantial as different clouds often use proprietary protocols and interfaces. This requires the cloud consumer to interact with various clouds using cloud provider-specific APIs and to develop ad-hoc adaptors in order to distribute and integrate heterogeneous resources and data assets to and from different clouds (even within a single organization). For example, to tackle the security issue, cloud consumers (e.g. the Eli Lilly research lab [9]) may have to split confidential data (e.g. the drug usage for each patient) into pieces and distribute them onto different clouds so that security compromise in one cloud will not lead to disaster as a whole. However, splitting and mixing data not only adds substantial extra financial cost, but can also severely affect the system performance (i.e. the time cost).

C. Charging Model

From a cloud provider's perspective, the elastic resource pool (through either virtualization or multi-tenancy) has made the cost analysis a lot more complicated than regular data centers, which often calculates their cost based on consumptions of static computing. Moreover, an instantiated virtual machine has become the unit of cost analysis rather than the underlying physical server. A sound charging model needs to incorporate all the above as well as VM associated items such as software licenses, virtual network usage, node and hypervisor management overhead, and so on.

For SaaS cloud providers, the cost of developing multi-tenancy within their offering can be very substantial. These include: re-design and re-development of the software that was originally used for single-tenancy, cost of providing new features that allow for intensive customization, performance and security enhancement for concurrent user access, and dealing with complexities induced by the above changes. Consequently, SaaS providers need to weigh up the trade-off between the provision of multi-tenancy and the cost-savings yielded by multi-tenancy such as reduced overhead through amortization, reduced number of on-site software licenses, etc. Therefore, a strategic and viable charging model for SaaS provider is crucial for the profitability and sustainability of SaaS cloud providers.

D. Service Level Agreement

Although cloud consumers do not have control over the underlying computing resources, they do need to ensure the quality, availability, reliability, and performance of these resources when consumers have migrated their core business functions onto their entrusted cloud. In other words, it is vital for consumers to obtain guarantees from providers on service delivery. Typically, these are provided through Service Level Agreements (SLAs) negotiated between the providers and consumers. The very first issue is the definition of SLA specifications in such a way that has an appropriate level of granularity, namely the tradeoffs between expressiveness and complicatedness, so that they can cover most of the consumer expectations and is relatively simple to be weighted, verified, evaluated, and enforced by the resource allocation mechanism on the cloud. In addition, different cloud offerings (IaaS, PaaS, SaaS, and DaaS) will need to define different SLA meta-specifications.

This also raises a number of implementation problems for the cloud providers. For example, resource managers need to possess precise and updated information on the resource usage at any particular time within the cloud. By updated information, we mean any changes in the cloud environment would fire an event subscribed to by the resource manager in order to make real-time evaluation and adjustment for SLA fulfillment. The resource managers need to employ fast and effective decision models and optimization algorithms to do this. It may need to reject certain resource requests when SLAs cannot be met. All these need to be carried out in a nearly automatic fashion due to the promise of "self-service" in the cloud computing. Furthermore, advanced SLA mechanisms need to constantly incorporate user feedback and customization features into the SLA evaluation framework.

E. What to migrate

Based on a survey (Sample size = 244) conducted by IDC in 2008, the seven IT systems/applications being migrated 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%). This result reveals that organizations still have security/privacy concerns in moving their data on to the Cloud. Currently, peripheral functions such as IT management and personal applications are the most easy IT systems to move. Organizations are conservative in employing IaaS compared to SaaS. This is partly because marginal functions are often outsourced to the Cloud, and core activities are kept in-house. The survey also shows that in three years time, 31.5% of the organization will move their Storage Capacity to the cloud. However this number is still relatively low compared to Collaborative Applications (46.3%) at that time.

V. CLOUD INTEROPERABIOLITY ISSUE

Currently, each cloud offering has its own way on how cloud clients/applications/users interact with the cloud, leading to the "Hazy Cloud" phenomenon [10]. This severely hinders the development of cloud ecosystems by forcing vendor lock-in, which prohibits the ability of users to choose from alternative vendors/offering simultaneously in order to optimize resources at different levels within an organization. More importantly, proprietary cloud APIs make it very difficult to integrate cloud services with an organization's own existing legacy systems (e.g. an on-premise data centre for highly interactive modeling applications in a pharmaceutical company). The scope of interoperability here refers both to the links amongst different clouds and the connection between a cloud and an organization's local systems. The primary goal of interoperability is to realize the seamless fluid data across clouds and between cloud and local applications.

There are a number of levels that interoperability is essential for cloud computing. First, to optimize the IT asset and computing resources, an organization often needs to keep in-house IT assets and capabilities associated with their core competencies while outsourcing marginal functions and activities (e.g. the human resource system) on to the cloud. In this case, frequent communications between cloud services (the HR system) and on-premise systems (e.g. an ERP system) becomes crucial and indispensable to run a business. Poor interoperability such as proprietary APIs and overly complex or ambiguous data structures used by a HR cloud SaaS will dramatically increase the integration difficulties, putting the IT department into a difficult situation. Second, more often than not, for the purpose of optimization, an organization may need to outsource a number of marginal functions to cloud services offered by different vendors. For example, it is highly likely that an SME may use Gmail for the email services and SalesForce.com for the HR service. This means that the many features (e.g. address book, calendar, appointment booking, etc.) in the email system must connect to the HR employee directory residing in the HR system.

A. Intermediary Layer

A number of recent works address the interoperability issue by providing an intermediary layer between the cloud consumers and the cloud-specific resources (e.g. VM). For example, Sotomayor et al. [11] proposed the notion of Virtual Infrastructure (OpenNebula) Management to replace native VM API interactions in order to accommodate multiple clouds - private or hybrid for an organization. OpenNebula works at the virtualization level, thus providing cloud consumers with a unified view and operation interfaces towards the underlying virtualization implementations of various types. Different from OpenNebula, Harmer et al. [12] developed an abstraction layer at a higher level. This provides a single resource usage model, user authentication model and API to shield cloud providers' heterogeneity that hinders the development of cloud-provider independent applications.

B. Standard

Standardization appears to be a good solution to address the interoperability issue. However, as cloud computing just starts to take off, the interoperability problem has not appeared on the pressing agenda of major industry cloud vendors. For example, neither Microsoft nor Amazon supports the *Unified Cloud Interface* (UCI) Project proposed by the Cloud Computing Interoperability Forum (CCIF) [13]. The standardization process will be very difficult to progress when these big players do not come forward to reach consensus. A widely used cloud API within the academia is the Eucalyptus project [14], which mirrors the well-known proprietary Amazon EC2 API for cloud operation. Although an Eucalyptus IaaS cloud consumer can easily connect to the EC2 cloud without substantial re-development, it cannot solve the general interoperability issue that requires an open API complied with by different types of Cloud providers.

C. Open API

SUN has recently launched the Sun Open Cloud Platform [15] under the Creative Commons license. A major contribution of this platform is the proposed (in-progress) the cloud API. It defines a set of clear and easy-to-understand RESTful Web services interfaces, through which cloud consumers are able to create and manage cloud resources, including compute, storage, and networking components in a unified way. Using the HTTP as the application protocol and JSON for resource representation, the open cloud API defines the following key resource types: Cloud, Virtual Data Center, Cluster, Virtual Machine, Private Virtual Network, Public Address, Storage Volume, and Volume Snapshot. These constructs share a certain degree of similarity with the internal architectural design of Eucalyptus [14]. In fact, the Eucalyptus project is willing to making efforts to ensure the compatibility between Eucalyptus clouds and the Sun cloud API [15]. This is aligned with DEBII's on-going research in providing a light-weight PaaS open API using RESTful Web services. Notice also that the notion of *Virtual Data Center*, which represents the core entity to instantiate the Sun Open Cloud, is equivalent to the concept of *Virtual Private Cloud* recently introduced in Amazon EC2 (See Section II - C).

D. SaaS and PaaS Interoperability

While the aforementioned solutions generally tackle with IaaS interoperability problems, few studies have focused on other service deployment models. SaaS interoperability often involves different application domains such as ERP, CRM, etc. A domain that is of particular interest to our research group at DEBII is the data mining research community. In the recent KDD09 panel discussion [16], a group of experts in the field of data mining raise the issue of establishing a data mining standard on the cloud, with a particular focus on "the practical use of statistical algorithms, reliable production deployment of models and the integration of predictive analytics" across different data mining-based SaaS clouds. Promising progress in this direction is the development of the Predictive Model Markup Language (PMML), a gradually accepted standard that allows users to exchange predictive models among various software tools.

To the best of our knowledge, we have not yet discovered considerable efforts made in providing PaaS interoperability. Since PaaS involves the entire software development lifecycle on the cloud, it would be more difficult to reach the uniformity with regards to the way consumers develop and deploy cloud applications.

VI. CONSLUSION

This paper discussed the challenges and issues of Cloud computing. We articulated the relationships amongst Cloud computing, Service-Oriented Computing, and Grid computing. We analyzed a few challenges on the way towards adopting Cloud computing. The interoperability issue was highlighted and a number of solutions are discussed thereafter for different cloud service deployment models.

REFERENCES

[1] P. Mell and T. Grance, "Draft nist working definition of cloud computing - v15," *21. Aug 2009,* 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A

Berkeley view of cloud computing," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28,* 2009.

[3] M. N. Huhns and M. P. Singh, "Service-Oriented Computing: Key Concepts and Principles," *IEEE Internet Computing,* vol. 09, pp. 75 - 81, 2005.

[4] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*: Morgan Kaufmann, 1998.

[5] J. Napper and P. Bientinesi, "Can cloud computing reach the top500?," in *Combined Workshops on UnConventional High Performance Computing Workshop plus Momory Access Workshop*, 2009, pp. 17-20.

[6] Y. Chen, V. Paxson, and R. Katz, "What's New About Cloud Computing Security?," 2010.

[7] A. Leinwand, "The Hidden Cost of the Cloud: Bandwidth Charges," *http://gigaom.com/2009/07/17/the-hidden-cost-of-the-cloud-bandwidth-charges/,* 2009.

[8] J. Gray, "Distributed computing economics," *ACM Queue,* vol. 6, pp. 63-68, 2008.

[9] M. May, "Forecast calls for clouds over biological computing," *Nature Medicine,* vol. 16, p. 6, 2010.

[10] M. Nelson, "Building an Open Cloud," *Science,* vol. 324, p. 1656, 2009.

[11] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing,* vol. 13, pp. 14-22, 2009.

[12] T. Harmer, P. Wright, C. Cunningham, and R. Perrott, "Provider-Independent Use of the Cloud," in *The 15th International European Conference on Parallel and Distributed Computing*, 2009, p. 465.

[13] "Unified Cloud Interface Project," *http://code.google.com/p/unifiedcloud/*.

[14] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," presented at the Proceedings of Cloud Computing and Its Applications, 2008.

[15] "Sun Microsystems Unveils Open Cloud Platform," *http://www.sun.com/aboutsun/pr/2009-03/sunflash.20090318.2.xml,* 2009.

[16] M. Zeller, R. Grossman, C. Lingenfelder, M. Berthold, E. Marcade, R. Pechter, M. Hoskins, and R. Holada, "Open standards and cloud computing: KDD-2009 panel report," in *KDD*, Paris, France, 2009, pp. 11-18.