

# Cloud Computing: Issues and Challenges

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.

摘要 - 许多人认为，云计算将重塑整个ICT行业的革命。在本文中，我们旨在确定云计算的挑战和问题。我们首先讨论两种相关的计算范例 - 面向服务的计算和网格计算，以及它们与云计算的关系。然后我们从云计算采用的角度确定了几个挑战。最后，我们将重点介绍值得进行大量进一步研究和开发的云互操作性问题。

Keywords: Cloud computing; Service-Oriented Computing; Distributed Computing; Web Services

关键词：云计算;面向服务的计算;分布式计算;Web服务

## 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 coexistent 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.

云计算最近成为分布式计算社区的热门话题。许多人认为，云计算将重塑IT行业的革命。那么，什么是云计算？它与面向服务的计算和网格计算有何不同？云提供商和消费者的一般挑战和问题是什么？在回答这些问题时，我们的目标是定义关键的研究问题，并阐明未来云计算的研究挑战和方向。为此，我们采用从外到内的方法来组织本文。我们首先研究了许多具有几个关键特性的云应用程序。然后，我们讨论云计算和面向服务的计算（SOC）之间的关系以及云和网格计算（即高性能计算）之间的关系。我们比较这三种计算范式，并提请注意它们如何以共存的方式相互受益。接下来，我们将讨论云计算的服务模型和部署模型。我们详细阐述了云的服务模型和部署模型，从而讨论了多个与数据相关的问题和挑战，如多租户，安全性等。最后，我们讨论互操作性和标准化问题。

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

## A.定义

什么是云计算？尽管学术界和工业界都提出了许多正式定义，但美国国家标准与技术研究院（国家标准与技术研究院）[1]提供的定义似乎包括云计算社区广泛使用的关键共同要素：

*云计算是一种模型，用于实现对可配置计算资源（例如，网络，服务器，存储，应用程序和服务）的共享池的方便的按需网络访问，这些资源可以通过最少的管理工作或服务提供商交互进行快速配置和发布 [1]。*

该定义包括云架构，安全性和部署策略。特别是，云计算的五个基本要素被清楚地表达出来：

**按需自助服务：**在特定时间段具有即时需求的消费者可以通过自动（即方便，自助服务）方式利用计算资源（例如CPU时间，网络存储，软件使用等），而无需诉诸与这些资源的提供者的人际互动。

广泛的网络访问：这些计算资源通过网络（例如，因特网）传送，并由位于消费者站点的异构平台（例如移动电话，膝上型电脑和PDA）的各种客户端应用程序使用。

资源池：云服务提供商的计算资源被“汇集”在一起，以便使用多租户或虚拟化模型为多个消费者服务，“根据消费者需求动态分配和重新分配不同的物理和虚拟资源”[1]。建立这种基于池的计算范式的动机在于两个重要因素：规模经济和专业化。基于池的模型的结果是物理计算资源对于消费者变得“不可见”，消费者通常不具有对这些资源（例如，数据库，CPU等）的位置，形成和原始性的控制或知识。例如，消费者无法分辨他们的数据将存储在云中的哪个位置。

快速弹性：对于消费者而言，计算资源变得直接而非持久：没有前期承诺和合同，因为他们可以随时使用它们进行扩展，并在缩小规模后释放它们。此外，资源供应对他们来说似乎是无限的，消费可以迅速上升，以便随时满足高峰需求。

测量服务。虽然计算资源由多个消费者汇集和共享（即多租户），但是云基础设施能够使用适当的机制通过其计量能力来测量每个个体消费者对这些资源的使用。

## **B. Service Model**

In addition to these five essential characteristics, the cloud community has extensively used the following three service models to categorize 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 upfront 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.

## B. 服务模式

除了这五个基本特征外，云社区还广泛使用以下三种服务模型对云服务进行分类：

**软件即服务（SaaS）。**云消费者在托管环境中发布他们的应用程序，托管环境可以由应用程序用户通过来自各种客户端（例如网络浏览器，PDA等）的网络访问。云消费者无法控制通常采用多租户系统架构的云基础架构，即不同的云消费者应用程序组织在SaaS云上的单一逻辑环境中，以实现规模经济和速度优化，安全性，可用性，灾难恢复和维护。SaaS的示例包括SalesForce.com，Google Mail，Google Docs等。

**平台即服务（PaaS）。**PaaS是一个支持完整“软件生命周期”的开发平台，允许云消费者直接在PaaS云上开发云服务和应用程序（例如SaaS）。因此，SaaS和PaaS之间的区别在于SaaS仅托管已完成的云应用程序，而PaaS提供了一个托管已完成和正在进行的云应用程序的开发平台。这要求PaaS除支持应用程序托管环境外，还拥有包括编程环境，工具，配置管理等在内的开发基础架构。PaaS的一个例子是Google AppEngine。

**基础架构即服务（IaaS）。**云消费者直接使用IaaS云中提供的IT基础架构（处理，存储，网络和其他基础计算资源）。虚拟化广泛用于IaaS云，以便以临时方式集成/分解物理资源，以满足云消费者不断增长或缩减的资源需求。虚拟化的基本策略是建立与底层硬件和其他VM隔离的独立虚拟机（VM）。请注意，此策略与多租户模型不同，后者旨在转换应用程序软件体系结构，以便多个实例（来自多个云消费者）可以在单个应用程序（即同一逻辑机器）上运行。IaaS的一个例子是亚马逊的EC2。

**数据存储即服务（DaaS）。**按需提供虚拟化存储成为单独的云服务 - 数据存储服务。请注意，DaaS可以被视为特殊类型的IaaS。其动机是，内部部署的企业数据库系统通常在专用服务器，软件许可证，交付后服务和内部IT维护中处于禁止的前期成本中。DaaS允许消费者为实际使用的内容而不是整个数据库的站点许可付费。除了传统的存储接口（如RDBMS和文件系统）之外，一些DaaS产品还提供表格格式抽象，旨在扩展以在非常压缩的时间范围内存储和检索大量数据，通常过大，过于昂贵或过于庞大。对于大多数商业RDBMS来说，应对的速度很慢。此类DaaS的示例包括Amazon S3，Google BigTable和Apache HBase等。

## 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. 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 be provisioned on a Public cloud or Private cloud.

最近，Cloud社区中定义了四种云部署模型：

私有云。云基础架构仅在单个组织内运行，由组织或第三方管理，无论其位于局部还是外部。在组织内设置私有云的动机有几个方面。首先，最大化和优化现有内部资源的利用。其次，包括数据隐私和信任在内的安全问题也使私有云成为许多公司的选择。第三，从本地IT基础设施到公共云的数据传输成本[2]仍然相当可观。第四，组织总是要求完全控制驻留在防火墙后面的关键任务活动。最后，学者们经常为研究和教学目的构建私有云。

社区云。一些组织共同构建和共享相同的云基础架构以及策略，要求，价值和关注点。云社区形成一定程度的经济可扩展性和民主均衡。云基础架构可以由第三方供应商托管，也可以在社区中的一个组织内托管。

公共云。这是当前云计算部署模型的主要形式。公共云由普通公共云消费者使用，云服务提供商拥有公共云的完全所有权，具有自己的策略，价值和利润，成本计算和计费模型。许多流行的云服务都是公共云，包括Amazon EC2, S3, Google AppEngine和Force.com。

混合云。云基础架构是两个或多个云（私有云，社区云或公共云）的组合，这些云仍然是独特的实体，但通过标准化或专有技术绑定在一起，从而实现数据 and 应用程序的可移植性（例如，用于云之间负载均衡的云爆发）。组织使用混合云模型来优化其资源，通过将外围业务功能保留到云上，同时通过私有云控制内部部署的核心活动来增加其核心竞争力。混合云提出了标准化和云互操作性的问题，将在后面的章节中讨论。



有趣的是，亚马逊网络服务（AWS）最近推出了一种新型部署模式 - 虚拟私有云（VPC），它是组织现有IT基础架构与亚马逊公共云之间的安全无缝桥梁。这是私有云和公共云之间的混合体。它是公开的，因为它仍然使用亚马逊汇集的计算资源给公众。然而，由于两个原因，它几乎是私人的。首先，IT遗留和云之间的连接通过虚拟专用网络得到保护，从而具有私有云的安全优势。事实上，即使是在公共云上，所有企业安全策略仍然适用于云上的资源。其次，AWS将为VPC提供一组“隔离”资源。但是，这并不意味着用户必须预先支付这些孤立的资源。用户仍然喜欢这些孤立资源的“按使用付费”。VPC代表了控制（私有云）和灵活性（公共云）之间的完美平衡。

请注意，服务模型与部署模型正交。例如，可以在公共云或私有云上配置SaaS。

### 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.

在本节中，我们将确定云计算，面向服务的计算（SOC）和网格计算之间的关系。

#### A. 云和面向服务的计算

SOC提供的封装，组件化，分散和集成功能非常重要：它们提供了体系结构原则和软件规范，以便通过Internet使用标准化协议连接计算机和设备[3]。实际上，云的概念或多或少基于SOC的不断发展，特别是SaaS服务模型。

SOC的进步可以通过多种方式使云计算受益。

云服务的描述。Web服务描述语言（WSDL）和REST协议是两种广泛使用的接口语言来描述Web服务。它们已被用于描述Cloud API规范。

云服务的发现。可以利用各种服务发现模型进行云资源发现，选择和服务级别协议验证。

Cloud Service的服务组合。由于Web服务是由组成业务应用程序而诞生的，因此可以利用该领域的大量研究来实现云服务集成，协作和组合。

云服务的管理。SOA治理和服务管理的研究和实践可以在云基础架构管理中进行调整 and 重用。

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.

另一方面，我们需要考虑SOC中缺少什么，特别是从中小企业的角度来看？SOC代表了集成和业务流程视角的高级抽象。但是，SOC没有提供运行服务的实用计算模型。例如，如何以最低成本运行我的服务？如何扩展/扩展基于面向服务的体系结构构建的应用程序。这些计算细节必须以项目特定和临时方式处理，这会增加SOC开发人员和中小企业IT部门的工作量。此外，如何将不同层次的服务纳入一个连贯的组织实体是SOC中的一个悬而未决的问题。例如，如何最大限度地利用我的IT服务以支持我的业务服务？因此，我们认为云计算可以通过几种重要方式使面向服务的计算研究受益。

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.

用于Web服务开发的云。云可以在PaaS服务部署模型下托管面向服务的开发。SOC开发通常需要为中小企业难以获得的分布式计算资源。例如，Google的AppEngine（平台，其SDK和客户端IDE）提供了一个成熟的开发平台，开发人员可以在其中开发和部署Java Web服务来构建他们的应用程序。此外，Yahoo Pipe平台还说明了Cloud可以作为服务Mashups和Composition的设计时和运行时的潜力。

用于Web服务测试的云。Web服务开发人员可以利用公共云中的无限计算资源来模拟真实的自动化机器请求和网络流，作为负载测试和服务压力测试的一种手段。模拟WS测试的网络流量的能力和成本一直是整体Web可靠性的抑制因素。云的极大计算资源的低成本和可访问性提供了复制地理分布式用户对这些系统的实际使用的能力，执行各种各样的用户场景，这在以前在传统测试环境中无法实现的规模。

用于Web服务部署的云。使用IaaS，可以简化Web服务部署。例如，在Amazon EC2设置下，服务部署者可以使用Amazon Machine Image（AMI）来分发他们的产品。当存在请求时，服务部署映像将加载到指定的虚拟机中以提供客户端请求。当从暂停模式（例如，从最后一次交易）恢复Web服务时，在服务交互期间产生的有状态信息也可以保持在AMI上。

服务流程制定的云。服务的集成和组合成为常见问题，其解决方案可以打包为部署在云环境中的服务。因此，流行的方法是利用人群（服务用户）的力量来允许使用各种算法（例如，基于案例的推理）重复使用具有较小配置和组成模式的即用型解决方案。

云和SOA / SOC的集成问题是一个有趣的问题。他们是否处于相同的技术/业务水平？他们的目标是实现同一目标吗？他们可以同时受雇吗？如果是这样，怎么样？这些是可以通过进一步开发和采用云计算来解决的研究挑战。

## 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.

## B. 云和网格计算

网格计算[4]是由先进科学研究中出现的实际问题所驱动的硬件和软件基础设施。据我们所知，Grid是分布式计算“中间件”，为科学和工程等高端计算应用提供“协调的跨组织资源共享”。云计算与网格计算之间存在明显的相似之处。首先，它们都旨在实现资源虚拟化。但是，它们确实存在显著差异：

Grid强调“资源共享”以形成虚拟组织。云通常由单个物理组织拥有（社区云除外，在这种情况下，它由社区拥有），他们将资源分配给不同的运行实例。

Grid旨在通过资源共享为巨大的任务提供最大的计算能力。云旨在根据用户的实时要求尽可能多地完成中小型任务。因此，多租户是云计算的一个非常重要的概念。



网格交换（科学）高性能计算的可重用性。云计算直接受到各种业务需求驱动的即时用户需求的影响。

Grid致力于实现最大化计算。云是按需计算 - 在扩展和上下扩展，同时优化整体计算能力。

### 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 "selfservice" nature of cloud computing, in which we often do not know a-prior how many physical processors do we need or have we used.

### C.云和高性能计算

高性能计算（HPC）旨在利用超级计算机和计算机集群来解决高级（科学）计算问题。云计算和HPC的最初意图可能明显不同，从而产生不同的计算范例和应用程序。虽然HPC已被广泛用于科学任务，但云计算已经开始用于服务业务应用程序。尽管并行化已经在HPC中得到充分利用，但许多业务应用程序之间的高度复杂的状态和数据依赖性使得在云计算中利用并行化计算方法对于业务应用程序变得更加困难。[5]中的作者指出，目前的云不适用于HPC，原因如下：首先，它在HPC上还不够成熟。其次，与集群计算不同，云基础架构侧重于整体上提升整体系统性能。第三，HPC旨在使用跨多个组织的资源来增强特定科学应用程序的性能。但关键的区别在于弹性：对于集群计算，容量通常是固定的，因此运行HPC应用程序通常需要相当多的人工交互（例如，基于具有固定数量的同质计算节点的特定集群进行调整）。这与云计算的“自助服务”性质形成鲜明对比，在云计算中，我们通常不知道我们需要或使用过多少个物理处理器。

## 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

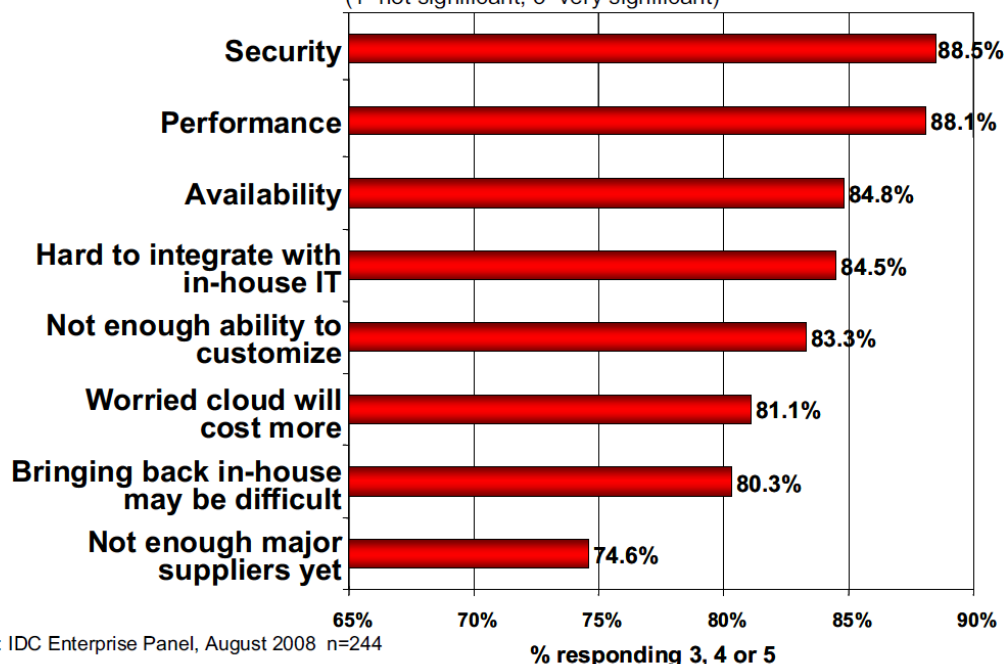
planning to use Cloud to organize botnet as Cloud often provides more reliable infrastructure services at a relatively cheaper price for them to start an attack [6].

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.

由于云计算仍处于起步阶段，目前的采用与众多挑战相关。根据IDC在2008年进行的一项调查，阻止云计算被采用的主要挑战得到了组织的认可，如图1所示。

### Q: Rate the *challenges/issues* of the 'cloud'/on-demand model

(1=not significant, 5=very significant)



#### A. 安全

很明显，安全问题在阻碍云计算方面发挥了最重要的作用。毫无疑问，使用别人的CPU将数据运行到其他人的硬盘上对许多人来说是令人生畏的。众所周知的安全问题，如数据丢失，网络钓鱼，僵尸网络（在一组计算机上远程运行）对组织的数据和软件构成严重威胁。此外，多租户模型和云计算中的池化计算资源引入了新的安全挑战[6]，需要新技术来解决。例如，黑客计划使用云来组织僵尸网络，因为云通常以相对便宜的价格提供更可靠的基础设施服务，以便他们开始攻击[6]。

多租户模型至少创建了两个新的安全问题。首先，同一物理机器上的共享资源（硬盘，数据，VM）会在恶意资源和常规资源之间发出意外的侧通道。其次，“声誉分享”的问题将严重损害许多优秀云“公民”的声誉，不幸的是，他们碰巧与他们的同伴——一个有犯罪心理的臭名昭着的用户共享计算资源。由于它们可能共享相同的网络地址，因此任何不良行为都将归因于所有用户，而无需区分真正的颠覆者与普通用户。

#### 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).

## B. 成本计算模型

云消费者必须考虑计算，通信和集成之间的权衡。迁移到云可以显著降低基础架构成本，但它确实会增加数据通信的成本，即将组织的数据传输到公共云和社区云[7]以及每单位成本（例如VM）的成本使用的计算资源可能更高。如果消费者使用混合云部署模型，其中组织的数据分布在许多公共/私人（内部IT基础设施）/社区云中，则此问题尤为突出。Gray [8]提出的“将计算放在数据附近”的论点仍然适用于云计算。直观地说，按需计算仅对CPU密集型作业有意义。换句话说，如果节省成本不能抵消额外的数据传输成本，那么从纯粹的经济观点来看，诸如ERP / CRM之类的交易应用可能不适合云计算。

此外，由于不同的云通常使用专有协议和接口，因此数据集成的成本可能很高。这要求云消费者使用特定于云提供商的API与各种云进行交互，并开发ad-hoc（点对点）适配器，以便将异构资源 and 数据资产分布和集成到不同的云中（甚至在单个组织内）。例如，为了解决安全问题，云消费者（例如Eli Lilly研究实验室[9]）可能不得不将机密数据（例如每个患者的药物使用情况）分成几部分并将其分发到不同的云上，以便在一个云不会导致整体灾难。然而，分割和混合数据不仅增加了大量额外的财务成本，而且还会严重影响系统性能（即时间成本）。

## 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 multitenancy 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.

### C. 收费模式

从云提供商的角度来看，弹性资源池（通过虚拟化或多租户）使得成本分析比常规数据中心复杂得多，常规数据中心通常根据静态计算的消耗来计算成本。此外，实例化的虚拟机已成为成本分析的单位，而不是底层的物理服务器。声音收费模型需要包含以上所有以及VM相关项目，例如软件许可证，虚拟网络使用，节点和管理程序管理开销等。

对于SaaS云提供商而言，在其产品中开发多租户的成本可能非常高。其中包括：重新设计和重新开发最初用于单租户的软件，提供允许密集定制的新功能的成本，用于并发用户访问的性能和安全性增强，以及处理由上述引起的复杂性变化。因此，SaaS提供商需要权衡多租户提供与多租户产生的成本节约之间的权衡，例如通过摊销减少管理费用，减少现场软件许可证数量等。因此，SaaS提供商的战略和可行的收费模式对于SaaS云提供商的盈利能力和可持续性至关重要。

### 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.

### D. 服务水平协议

虽然云消费者无法控制底层计算资源，但当消费者将其核心业务功能迁移到委托云时，他们确实需要确保这些资源的质量，可用性，可靠性和性能。换句话说，消费者在服务提供方面获得保证是至关重要的。通常，这些是通过提供商和消费者之间协商的服务水平协议（SLA）提供的。第一个问题是SLA规范的定义，其具有适当的粒度级别，即表达性和复杂性之间的权衡，以便它们可以覆盖大多数消费者期望，并且相对简单地加权，验证，由云上的资源分配机制评估和实施。此外，不同的云产品（IaaS，PaaS，SaaS和DaaS）需要定义不同的SLA元规范。

这也为云提供商带来了许多实施问题。例如，资源管理器需要拥有云中任何特定时间的资源使用情况的精确和更新信息。通过更新信息，我们的意思是云环境中的任何更改都会触发资源管理器订阅的事件，以便对SLA实现进行实时评估和调整。资源管理员需要采用快速有效的决策模型和优化算法来实现这一目标。当无法满足SLA时，可能需要拒绝某些资源请求。由于云计算中“自助服务”的承诺，所有这些都

需要以近乎自动的方式进行。此外，高级SLA机制需要不断将用户反馈和定制功能纳入SLA评估框架。

### 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.

### E. 迁移什么

根据IDC在2008年进行的一项调查（样本量= 244），迁移到云的七个IT系统/应用程序是：IT管理应用程序（26.2%），协作应用程序（25.4%），个人应用程序（25%），应用程序开发和部署（16.8%），服务器容量（15.6%）和存储容量（15.5%）。此结果表明，组织在将数据迁移到云时仍存在安全/隐私问题。目前，IT管理和个人应用程序等外围功能是最容易移动的IT系统。与SaaS相比，组织使用IaaS是保守的。这部分是因为边际功能通常外包给云，而核心活动则保留在内部。该调查还显示，在三年时间内，31.5%的组织将其存储容量迁移到云端。然而，与当时的协作应用程序（46.3%）相比，这个数字仍然相对较低。

## V. CLOUD INTEROPERABILITY 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 lockin, 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.

目前，每个云产品都有自己的方式了解云客户端/应用程序/用户如何与云交互，从而导致“朦胧云”现象[10]。这通过强制供应商锁定严重阻碍了云生态系统的发展，这阻碍了用户同时从备选供应商/产品中进行选择以优化组织内不同级别的资源的能力。更重要的是，专有云API使得将云服务与组织自己的现有遗留系统（例如，制药公司中的高度交互式建模应用程序的内部部署数据中心）集成起来非常困难。这里的互操作性范围既指不同云之间的链接，也指云与组织本地系统之间的连接。互操作性的主要目标是实现跨云以及云和本地应用程序之间的无缝流体数据。

互操作性有许多级别对于云计算至关重要。首先，为了优化IT资产和计算资源，组织通常需要保留与其核心能力相关的内部IT资产和能力，同时将边际功能和活动（例如人力资源系统）外包给云。在这种情况下，云服务（HR系统）和内部部署系统（例如ERP系统）之间的频繁通信变得至关重要，并且是运营企业必不可少的。诸如专有API之类的互操作性差以及HR云SaaS使用的过于复杂或模糊的数据结构将极大地增加集成难度，使IT部门陷入困境。其次，为了优化，组织可能需要将许多边际功能外包给不同供应商提供的云服务。例如，SME很可能使用Gmail作为电子邮件服务，使用SalesForce.com作为HR服务。这意味着电子邮件系统中的许多功能（例如地址簿，日历，预约预约等）必须连接到驻留在HR系统中的HR员工目录。

### 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.

### A.中间层

许多最近的工作通过在云消费者和云特定资源（例如VM）之间提供中间层来解决互操作性问题。例如，Sotomayor等。[11]提出了虚拟基础设施（OpenNebula）管理的概念，以取代本机VM API交互，以适应多个云 - 私有或混合组织。OpenNebula在虚拟化级别工作，从而为云用户提供统一的视图和操作界面，以实现各种类型的底层虚拟化实施。与OpenNebula不同，Harmer等人[12]在更高层次上开发了一个抽象层。这提供了单一资源使用模型，用户身份验证模型和API，以保护云提供商的异构性，从而阻碍云提供商独立应用程序的开发。

### 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 redevelopment, it cannot solve the general interoperability issue that requires an open API complied with by different types of Cloud providers.

### B.标准

标准化似乎是解决互操作性问题的良好解决方案。然而，随着云计算刚刚开始起飞，互操作性问题并未出现在主要行业云供应商的紧迫议程中。例如，Microsoft和亚马逊都不支持云计算互操作性论坛 (CCIF) [13]提出的统一云接口 (UCI) 项目。当这些大型企业未能达成共识时，标准化进程将很难取得进展。学术界广泛使用的云API是Eucalyptus项目[14]，它反映了众所周知的专有Amazon EC2 API，用于云操作。虽然Eucalyptus IaaS云消费者可以轻松连接到EC2云而无需大量重新开发，但它无法解决需要不同类型云提供商遵守的开放API的一般互操作性问题。

### 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 lightweight 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).

### C. 开放API

SUN最近根据知识共享许可推出了Sun Open Cloud Platform [15]。该平台的主要贡献是提议（正在进行）云API。它定义了一组清晰且易于理解的RESTful Web服务接口，通过这些接口，云消费者能够以统一的方式创建和管理云资源，包括计算，存储和网络组件。使用HTTP作为应用程序协议和JSON进行资源表示，开放云API定义了以下关键资源类型：云，虚拟数据中心，群集，虚拟机，专用虚拟网络，公共地址，存储卷和卷快照。这些结构与按树的内部建筑设计具有一定程度的相似性[14]。事实上，Eucalyptus项目愿意努力确保Eucalyptus云与Sun cloud API之间的兼容性[15]。这与DEBII正在进行的研究一致，即使用RESTful Web服务提供轻量级PaaS开放API。另请注意，虚拟数据中心（代表实例化Sun Open Cloud的核心实体）的概念等同于最近在Amazon EC2中引入的虚拟私有云的概念（参见第II-C节）。

### 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.

#### D. SaaS和PaaS互操作性

虽然上述解决方案通常解决IaaS互操作性问题，但很少有研究关注其他服务部署模型。SaaS互操作性通常涉及不同的应用程序域，如ERP，CRM等。我们DEBII研究小组特别感兴趣的领域是数据挖掘研究社区。在最近的KDD09小组讨论[16]中，数据挖掘领域的一组专家提出了在云上建立数据挖掘标准的问题，特别关注“统计算法的实际应用，可靠的生产部署模型和预测分析的集成”跨越不同的基于数据挖掘的SaaS云。在这方面取得的进展是预测模型标记语言（PMML）的发展，PMML是一种逐渐被接受的标准，允许用户在各种软件工具之间交换预测模型。

据我们所知，我们尚未发现在提供PaaS互操作性方面做出的巨大努力。由于PaaS涉及云上的整个软件开发生命周期，因此在消费者开发和部署云应用程序方面达成一致将更加困难。

## 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.

本文讨论了云计算的挑战和问题。我们阐述了云计算，面向服务的计算和网格计算之间的关系。我们分析了采用云计算的一些挑战。突出了互操作性问题，之后针对不同的云服务部署模型讨论了许多解决方案。