Chapter 7

Cloud Architecture and Datacenter Design

Summary: This chapter covers the design principles and enabling technologies for cloud platform architectural design. We start with datacenter design and management. Then we present the design choices of cloud platforms. The topics covered include layered platform design, virtualization support, resource provisioning, and infrastructure management. Chapter 8 will cover cloud computing platforms built by Google, Amazon, IBM, Microsoft, and Salesforce.com. Case studies of some current and future clouds will be given in Chapter 9.

7.1 Cloud Computing and Service Models

Over the past two decades, the world economy is rapidly moving from manufacturing to services. In 2010, 80% of the US economy is driven by service industry, leaving only 15% by manufacturing and 5% from the agriculture. Cloud computing benefits primarily the service industry and advance the business computing to a new paradigm. It has been forecasted that global revenue in cloud computing may reach $ 150 billion by 2013 from the $ 59 billion reported in 2009. We have introduced the basic concept of cloud computing in Chapter 1. In this and next 2 chapters, we will study cloud computing from all angles. T

In this chapter, we study cloud architecture and infrastructure design. The next chapter focuses on real cloud platforms built in recent years, their service offerings, programming and application development. Virtualized cloud platforms are often built on top of datacenters, we will study the design and roles of datacenters first in support of the cloud development. In this sense, clouds aim to power the next generation datacenters by architecting them as a network of virtual computing services including hardware, database, user-interface, application logic, etc.

The users are able to access and deploy applications from anywhere in the world on demand at competitive costs depending on users QoS (*Quality of Service*) requirements. Developers with innovative ideas for new Internet services no longer require large capital outlays in hardware to deploy their service or human expense to operate it. The cloud offers significant benefit to IT companies by freeing them from the low level task of setting up hardware (servers) and software infrastructures. This will free up users to focus on innovation and creating business value for the computing services they need.

7.1.1 Public, Private, and Hybrid Clouds

Cloud computing applies a virtual platform with elastic resources putting together by on-demand provisioning of hardware, software, and datasets, dynamically. The idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at datacenters. Cloud computing leverages its low cost and simplicity to both providers and users. Cloud computing intends to leverage multitasking by serving many heterogeneous applications simultaneously. The computations (programs) are sent to where the data is located, rather than copying the data to millions of desktops. Cloud computing avoids large data movement resulting in better network bandwidth utilization. Furthermore, machine virtualization has enabled the cost-effectiveness in using the cloud platforms.

The concept of cloud computing has evolved from the concepts of cluster, grid, and utility computing and providing software as a service. Cluster and grid computing leverage the use of many computers in parallel to solve a few large problems. Utility and SaaS provides the computing resources as a service with a notion of pay per use. Cloud computing leverage multiple resources to deliver a service to the end user. It is a HTC paradigm where the infrastructure provide the services through a large datacenter or server farms. Cloud computing model enables the users to share access of resources from anywhere at any time through their connected devices.

Some people argued that cloud computing is centralized computing at datacenters. We argue that cloud computing is indeed practicing distributed parallel computing over datacenter resources. All computations associated with a single cloud application are still distributed to many servers in multiple datacenters. These centers may have to communicate with each other around the globe. In this sense, cloud platforms are indeed distributed systems. Figure 7.1 shows three cloud classes: *private, public* and *hybrid clouds* and their analogy with offering various types of training services. They are deployed in the Intranets and over the open Internet as illustrated in Figure 7.2. Note that these clous are created over all Internet domains By no means, they are centralized in one place, just like many branch bank offices scattered around in a large banking systemAs clouds evolve, they will be interconnected to support the delivery of application services in a scalable and efficient manner to consumers around the world.

Public Clouds: A *public cloud* is built over the Internet, which can be accessed by any user who has paid for the service. Public clouds are owned by service providers. They are accessed by subscription. Many companies have built public clouds, namely Google App Engine, Amazon AWS, Microsoft Azure, IBM Blue Cloud, and Salesforce Force.com. These are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure. A public cloud delivers selected set of business processes. The application and infrastructure services are offered qith quite flexible price per use basis.

Private Clouds: The *private cloud* is built within the domain of an intranet owned by a single organization. Therefore, they are client owned and managed. Their access is limited to the owning clients and their partners. Their deployment was not meant to sell capacity over the Internet through publicly accessible interfaces. Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains. A private cloud is supposed to deliver more efficient and convenient cloud services. They may impact the cloud standardization, while retai8ning greater customization and organizational control.

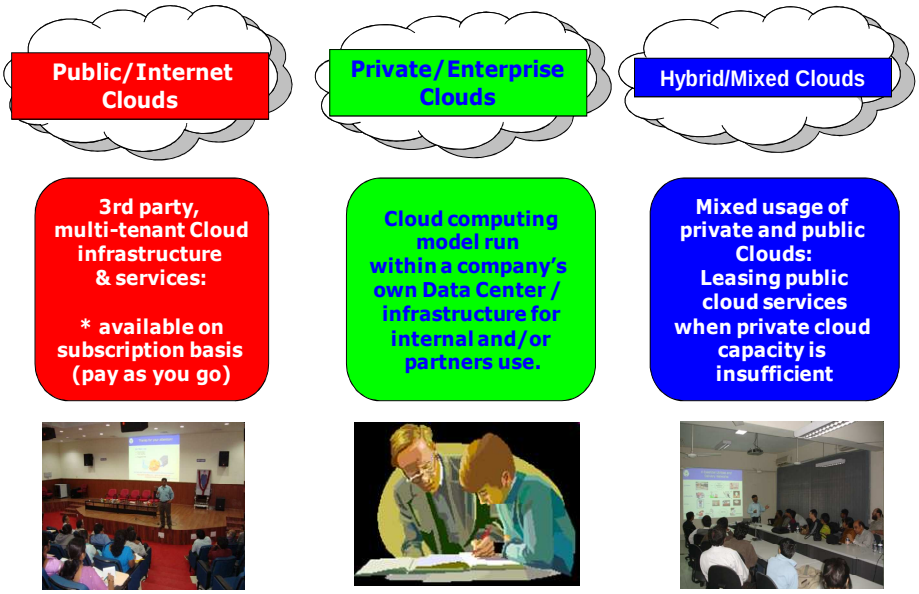


Figure 7.1: Classes of clouds and their analogy to training services

Hybrid Clouds: A *hybrid cloud* is built with both public and private clouds, as shown at the lower left corner of Fig.6.2. Private clouds can also support a *hybrid cloud* model by supplementing local infrastructure with computing capacity from an external public cloud. For example, the *research compute cloud* (RC2) is a private cloud built by IBM. The RC2 interconnects the computing and IT resources at 8 IBM Research Centers scattered in US, Europe, and Asia. A hybrid clouds provides access to client, partner network, and third party. In summary, public clouds promotes standardization, preserves capital investigation, offers application flexibility. The private clouds attempt to achieve customization and offer higher efficiency, resiliency, security, and privacy. The hybrid clouds operates in the middleway with compromises.

Cloud Core Structure: The core of a cloud is the server cluster (or VM cluster). Majority of the cluster nodes are used as compute nodes. A few control nodes are used to manage and monitor of the cloud activities. The scheduling of user jobs requires to assign the work to various virtual clusters. The gateway nodes provide the access points of the service from the outside world. These gateway nodes can be also used for security control of the entire cloud platform. In clusters and grids we have studied in chapter 3 and

4, we expect static demand of resources. Clouds are designed to face fluctuating workload and thus variable resource demand. It is anticipated that private clouds will satisfy this demand more efficiently.

For example, NASA (National Agency of Space and Aeronautics) of US government is building a private cloud to enable researchers to run climate models on remote systems provided by NASA. This can save the users from capital expenses in HPC at local sites. Furthermore, NASA can build the complex weather models around their datacenters, which is more cost effective. Another good example is CERN (Center of European Research on Nuclear) is developing a very big private cloud to distribute data, applications, and computing resources to thousands of scientists around the world. Therefore, most of the action is in private clouds today. Public clouds will be launched by HPC vendors in the years to come.

These cloud models demand different levels of performance, data protection, and security enforcement. Different *service level agreements* (SLAs) may be applied to satisfy the both providers and paid users. Cloud computing exploits many existing technologies. For example, grid computing is the backbone of cloud computing in that grid has the same goals of resource sharing with better utilization of research facilities. Grids were more focused to deliver storage and computing resources while cloud computing aims at the economy-of-scale with abstracted services and resources.

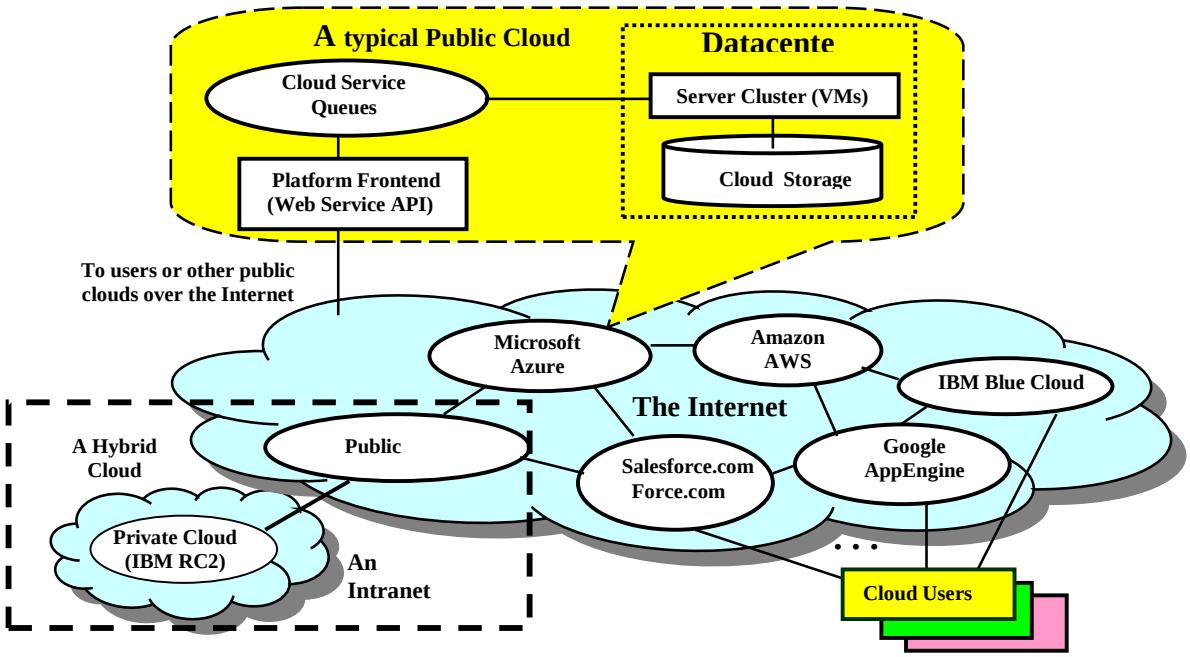


Figure 7.2 Public, private, and hybrid clouds over the Internet and intranets. The callout box shows the architecture of a typical public cloud. A private cloud is built within an intranet. A hybrid cloud involves both public and private clouds in its range. Users access the clouds from a web browser or through an special application programming interface (API).

For example, the email application can run in the service accessing get nodes and provide the user interface for outside users. The application can get the service from the internal cloud computing services e.g. the email storage service. There are also some service nodes for supporting the whole cloud computing cluster to run properly. These nodes are called as runtime supporting service nodes. For example, there might be distributed locking service for supporting some specific applications. Finally, it is possible that there are some independent service nodes. Those nodes provide independent service for other nodes in the cluster. For example, the news service might need the geographical information, there should be some services nodes providing such data.

With cost-effective performance as the key concept of clouds, we will consider public cloud, unless otherwise specified. Many executable application codes are much smaller than the web-scale datasets they process, Cloud computing avoids large data movement durin execution. This will result in less traffic on the Internet and better network utilization. Clouds also alleviate the Petascale I/O problem. The cloud performance and its QoS are yet to be proven in more real-life applications. We will model the performance of cloud computing in Chapter 7 along with data protection, security measures, service availability, fault-tolerance, and operating cost.

7.1.2 Cloud Ecosystem and Enabling Technologies

It was estimated by IBM that the worldwide cloud service market may reach $126 billion by 2012 including components, infrastructure services, and business services. Internet clouds work as service factories built around multiple datacenters. We introduce below the cloud ecosystems, cost modeling, and enabling technologies. These are important for our readers to understand the motivations behind cloud computing and the major barriers yet to be removed to make cloud computing services a reality.

Cloud Design Objectives: Despite the controversy surrounding the replacement of desktop or deskside computing by centralized computing and storage services at the datacenters or big IT companies, the cloud computing community has reached some consensus on what have to be done to make cloud computing universally acceptable. We list below six design objectives of cloud computing:

- S*hifting Computing from Desktops to Datacenters :* The shift of computer processing, storage, and software delivery away from desktop and local servers to datacenters over the Internet.
- *Service Provisioning and Cloud Economics:* Provider supply cloud services by signing SLAs with consumers and end users. The services must be resource economic with efficiency in computing, storage, and power consumption, etc. Pricing models are based on a *pay-as-you-go* policy.
- *Scalability in Performance:* The cloud platforms and software and infrastructure services must be able to scale in performance as the number of users mounting.
- *Data Privacy Protection:* Can you entrust the datacenters to handle your private data and records ? This concern must be addressed to make cloud successful as trusted services.
- *High Quality of Cloud Services:* The QoS of cloud computing must be standardized to remove doubt over services provided to users. Cloud interoperability is required across multiple providers.
- *New Standards and Interfaces:* This refers to solving the data lock-in problem associated with datacenters or cloud providers, Universally accepted APIs and access protocols are need to provide high portability and flexibility of virtualized applications.

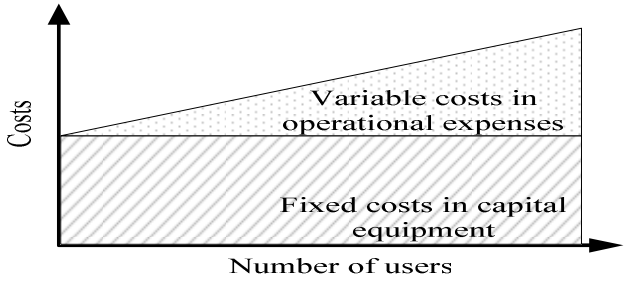
We will study most of the issues in this chapter and the remaining ones on security and performance in Chapter 7. Let us analyze below the cloud economics of scale as a starting point.

Cloud Ecosystem and Cost Model: In traditional IT computing, user must acquire their own computer and peripheral equipment as capital expenses. In addition, they have to face operational expenditure in operating and maintaining the computer systems, including the human and service costs. Figure 7.3(a) shows the adding of the variable operational costs on top of the fixed capital investment in traditional IT. Note that the fixed costs is a forefront costs which could be lowered slightly with increasing number of users. But the operational costs may increases sharply with larger number of users. Therefore, the total cost escalates quickly with massive number of users. On the other hand, Cloud computing applies a pay-per-use business model. User jobs are outsourced to the datacenters.

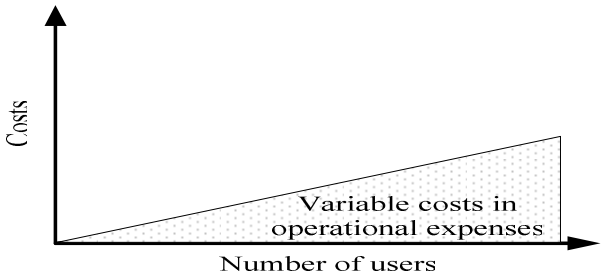
To use cloud, there is no out-front costs in acquiring heavy machines. Only variable costs are experienced by cloud users as demonstrated in Fig.7.3(b). Overall, cloud computing will reduce the computing costs significantly for both small users and large enterprises. Computing economics does shows

a big gap between traditional IC users and cloud users. The savings in acquiring no expensive computers outfront releasae a lot of burdens for startup companies. The fact that cloud users only pay for the operational expenses with no need to invest on permanent equipment is especially attractive to massive number of small users. This is a major driven force for cloud computing to become appealing to most enterprises and heavy computer users. In fact, any IT users whose capital expenses in under greater pressure than their operational expenses should consider sending their overflow work to utility computing or cloud service providers.

In general, private cloud clouds leverage existing IT infrastructure and personnel within an enterprise or government organization. Both public and private clouds handle workloads dynamically. However, public clouds should be designed to handle workloads without communication dependency. Both types of clouds distribute data and VM resources. However, private cloud can balance the workloads to exploit IT resources more efficiently within the same Intranet. Private cloud can also provide pre-production testing and enforce data privacy and security policies more effectively. In a public cloud, the surge workload is often off-loaded. The major advantage of public clouds lies in the avoidance of the capital expenses by users in IT investments in hardware, software, and personnel.



(a) Traditional IT cost model



(b) Cloud computing cost model

Figure 7.3 Computing economics between traditional IC users and cloud users, where traditional users must acquire expensive computers out front, while the cloud users only pay for the service provided with no major investment on large servers or expensive computer equipment.

Cloud Ecosystem for Building Private Clouds: With the emergence of various Internet clouds, an ecosystem of providers, users, and technologies has appeared. This ecosystem has evolved around public clouds. There exists strong interest is growing in open source cloud computing tools that let organizations build their own IaaS clouds using their internal infrastructures. Private and hybrid clouds are not exclusive, since public clouds are involved in both. A private/hybrid cloud can allow remote access to its resources over the Internet using remote web services interfaces like that used in Amazon EC2.

Figure 7.4 shows the ecosystem for building private clouds suggested by Sotomayer, et al [39]. They suggested 4 levels of ecosystem development in a private cloud: At the user end, consumers demand a flexible platform. At the cloud management level, cloud manger provides virtualized resources over an IaaS

platform. At the *virtual infrastructure* (VI) management level, the manager allocates VMs over multiple server clusters. Finally, at the VM management level, the VM managers handle VMs installed on individual host machines. An ecosystem of cloud tools attempt to span both cloud management and VI management. Integrating cloud management solutions with existing VI managers is complicated by the lack of open and standard interfaces between the two layers.

Many small cloud providers have appeared besides big IT industry, such as GoGrid, FlexiScale, and ElasticHosts. An increasing number of startup companies are now based their IT strategy on cloud resources, spending little or no capital to manage their own IT infrastructures. We desire a flexible and open architecture that organizations can build private/hybrid clouds. The VI management is aimed to this end. Example VI tools include Ovirt (*http://ovirt.org*), VMware vSphere (*www.vmware.com/products/vsphere/*), and the Platform VM Orchestrator (www.platform.com/Products/platform-vm-orchestrator). These tools support dynamic placement and VM management on a pool of physical resources, automatic load balancing, server consolidation, and dynamic infrastructure resizing and partitioning.

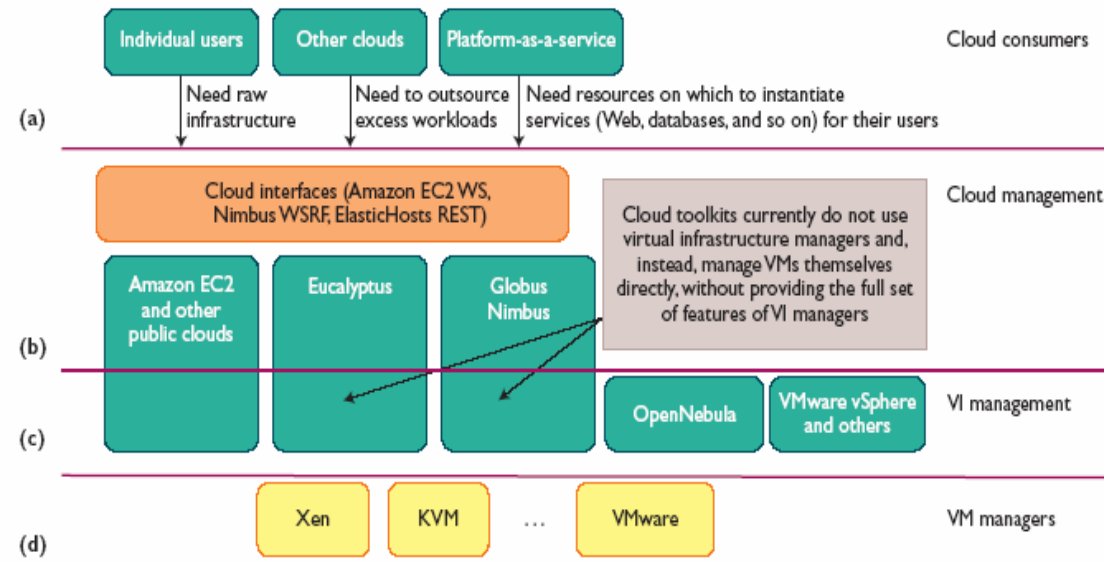


Figure 7.4 Cloud ecosystem for building private clouds. (a) Consumers demand a flexible platform. (b) Cloud manger provides virtualized resources over an IaaS platform. (c) Virtual infrastructure (VI) manager allocates VMs to server clusters. (d) The VM managers handle VMs installed on individual servers. (Courtesy of Sotomayor, Montero, and Foster, *IEEE Internet Computing,* Sept. 2009. [59])

7.1.3 Popular Cloud Service Models

Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers. The services provided over the cloud can be generally categorized into three different service models namely the IaaS, PaaS, and SaaS. These form the three pillars on top of which Cloud Computing solutions are delivered to end users. All the three models allow the user to access the services over the Internet, relying entirely on the infrastructures of the cloud service providers. These models are offered based on various SLAs between the providers and users. In a broad sense, the SLA for cloud computing is addressed in terms of the service availability performance and data protection and security aspects. Three cloud models are illustrated in Fig. 7.5 at different service levels of the cloud.

Infrastructure as a Service (IaaS): This model allows users to rent processing, storage, networks, and other resources. The user can deploy and run the guest OS and applications. The user does not manage or control the underlying cloud infrastructure but has control over OS, storage, deployed applications, and possibly select networking components. This IaaS model encompasses the storage as a service, computation resource as a service, and communication resource as a service. Example for this kind of service is: Amazon-S3 for storage, Amazon-EC2 for computation resources, and Amazon-SQS for communication resources. IaaS providers charge users based on the capability and capacity of requested infrastructure for a given duration. In case of Amazon IaaS environment, users can create, launch, and terminate server instances as needed, paying by the hour for active servers.

Platform as a Service (PaaS): Although one can develop, deploy, and manage execution of applications using basic capabilities offered under IaaS model, but it is very complex to do so due the lack of tools that enable rapid creation of applications and automated management and provisioning of resources depending on workload and users requirements. They requirements are met by PaaS, which offers the next-level of abstraction and is built using services offered by IaaS. The PaaS model provides the user to deploy user-built applications on top of the cloud infrastructure, that are built using the programming languages and software tools supported by the provider (e.g., Java, python, .Net).

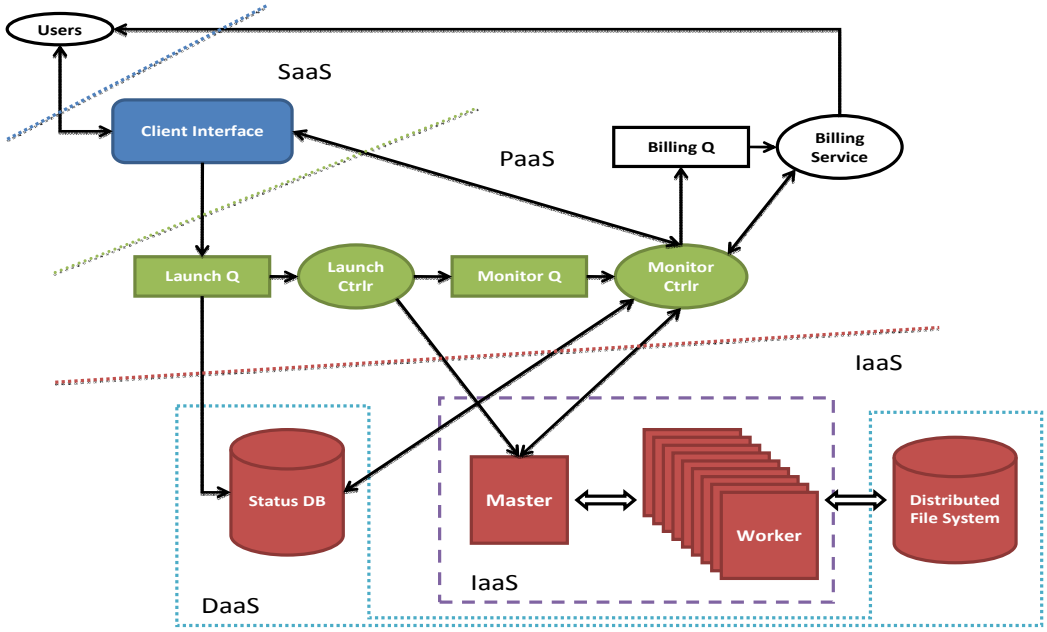


Figure 7.5: The IaaS provides virtualized infrastructure at user's costs. The PaaS is applied at the platform application level. The SaaS provides specific software support for users at web service level. DaaS (Data as a Service) applies the status database and distributed file system.

The user does not manage the underlying cloud infrastructure. The cloud provider facilitates to support the entire application development, testing and operation support on a well-defined service platform. This PaaS model enables the means to have a collaborated software development platform for developers from different parts of the world. Other service aspects in this mode include the third party to provide software management, integration and service monitoring solutions. Cloud services offered under PaaS model include: Google App Engine, Microsoft Azure, and Manjrasoft Aneka.

Software as a Service (SaaS): This refers to browser-initiated application software over thousands of cloud customers. Services and tools offered by PaaS are utilized in construction of applications and management of their deployment on resources offered by IaaS providers. SaaS model provides the software applications

as a service. As a result, on the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are rather low, compared with conventional hosting of user applications. The customer data is stored in the cloud that is either vendor proprietary or a publically hosted cloud supporting the PaaS and IaaS. Vast majority of the business logic software are delivered as a service. Microsoft online sharepoint and CRM software from Salesforce.com are good examples.

Providers such as Google and Microsoft offer integrated IaaS and PaaS services whereas others such as Amazon and GoGrid offer pure IaaS services and expect third parties PaaS providers such as Manjrasoft to offer application development and deployment services on top of their infrastructure services. To help our readers identify some cloud applications in enterprises, we share the following stories on three real-life cloud applications related to HTC, news media, and business transactions. The benefits of using cloud services are self-evident in these applications.

Customized Cloud Services: At present, public clouds are in use by growing number of users. Due to the lack of trust to leak sensitive data in the business world, more and more enterprises, organizations, and communitites are developing private clouds that demands deep customization. The concept is illustrated in Fig.7.6 for an enterprise cloud. This cloud will be used by multiple users within the organization. Each use needs to build strategic applications on the cloud. The user demands customized partition of the data, logic and database in the metadata representation. The user click the selection and enter their specific application code during the customization process. The blacken virtual machines at the upper right corner is chosen to form the coherent code base and managed infrastructure at the bottom. Furthermore, the user can upgrade its demand when convenient and also preserve the IP control for private usage of the provisioned cloud resources. We will see more and more of this kind of private clouds in the future.

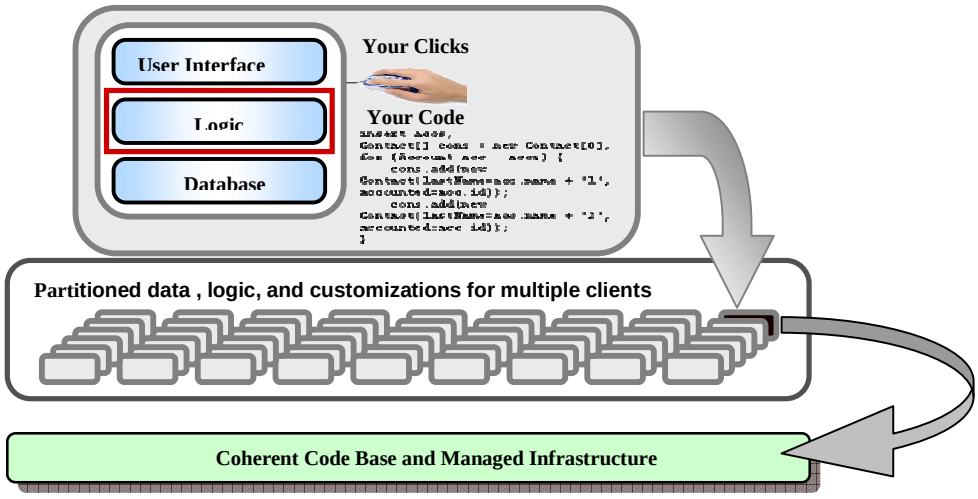


Figure 7.6 Enterprise clouds enable deep customization through metadata partitioning for multiple clients (Personal communication with Peter Coeffee, Salesforce.com., April 24, 2010)

Example 7.1: Some Success Stories on Cloud Service Applications

(1). To discover new drugs through DNA sequence analysis, Eli Lily Company has used Amazon's AWS platform with provisioned server and storage clusters to conduct high-performance biological sequence analysis without using an expensive supercomputer. The benefit of this IaaS application is a reduced drug deployment time with much lower cost.

(2). Another good example is New York Times applying Amazons, EC2 and S3 services to retrieve useful pictorial information quickly from millions of archival articles and news papers. The N.Y.

Times has significantly reduced their time and cost in getting job done more effectively.

(3). The third example is Pitney Bowes, an e-commerce Company, offers their clients the opportunity to perform B2B (*Business-to-Business*) transactions using Microsoft Azure platform, along with .net and SQL services. They end up with a significant increase in their client basis. ■

7.2 Datacenter Design and Interconnection Networks

We present below the basic architecture and design considerations of datacenters. A cloud architecture is built with commodity hardware and network devices. Almost all cloud platforms choose the popular x86 processors. The low-cost Terabyte disks and Gigabit Ethernet are used to build datacenters. Datacenter design emphasizes more on the performance/price ratio rather than the speed performance alone. The storage and energy efficiency are more important than the shear speed performance. Figure 7.7 shows the server growth and cost breakdown of datacenters over the past 15 years. Worldwide, there are about 43 millions of servers in use by 2010.

Datacenter Growth and Cost Breakdown : A large datacenter may be built with ten thousands or more servers. Smaller ones are built with hundreds or thousands of servers. The costs to build and maintain datacenter servers are increasing over the years. To keep a datacenter running well, typically, only 30% costs are due to purchase of IT equipments (such as servers and disks, etc), 33% costs are attributed to chiller, 18% on UPS (*uninterruptible power supply*), 9% on CRAC (*computer room air conditioning*), and the remaining 7% due to power distribution, lighting, and transformer costs. Thus the cost to run a datacenter is dominated by about 60% in management and maintenance costs. The server purchase cost did not increase much with time. The cost of electricity and cooling did increase from 5% to 14% in 15 years.

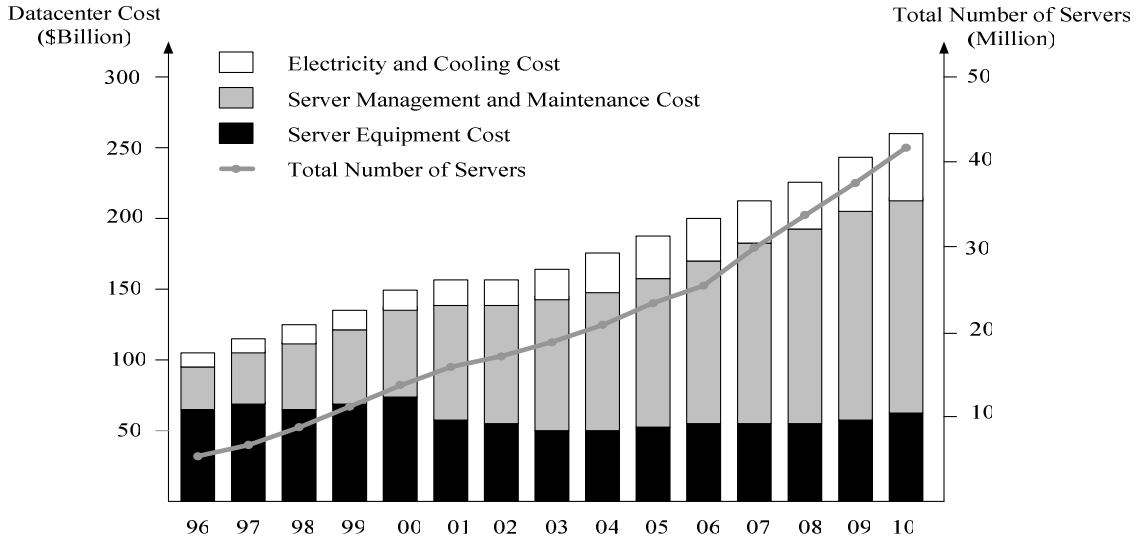


Figure 7.7 Growth and cost breakdown of datacenters over the years (Source: IDC Report 2009).

Low-Cost Design Philosophy: High end switches or routers cost a lot of money. Thus using high-end network devices does not fit the economics of cloud computing. However, the cost only plays one side of the story. The other side is to provide more bandwidth. Given a fix number of budget, much more low-end commodity switches can be purchased than the high end devices. The large number of low-end commodity switches can provide network redundancies as well as much larger bandwidth. This is also the same story while choosing the large number of commodity x86 servers instead of small number of mainframes. While using the large number of low cost commodity switches, software developer should design the software

layer for handling the network traffic balancing, fault tolerant as well as expandability. This is also the same on the server side. The network topology design must face such situation. Currently, near all the cloud computing datacenters are using the Ethernet as the fundamental network technology.

7.2.1 Warehouse-Scale Datacenter Design

Figure 7.8 shows a programmer's view of storage hierarchy of a typical WSC. A server consists of a number of processor sockets, each with a multicore CPU and its internal cache hierarchy, local shared and coherent DRAM, and a number of directly attached disk drives. The DRAM and disk resources within the rack are accessible through the first-level rack switches (assuming some sort of remote procedure call API to them), and all resources in all racks are accessible via the cluster-level switch.

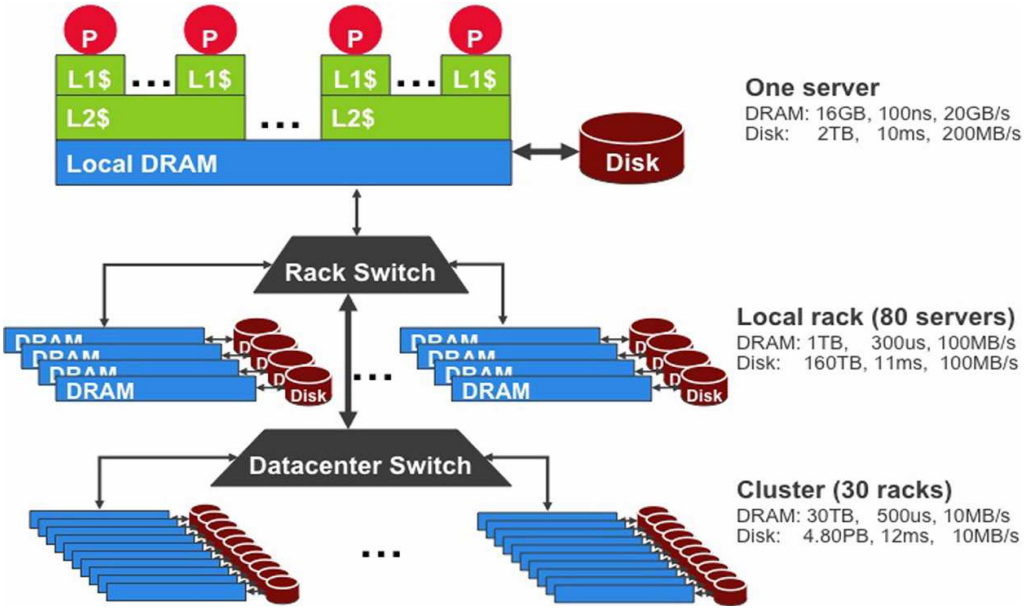


Figure 7.8 The architecture and storage hierarchy of a warehouse-scale datacenter. (Courtesy of Barroso and Holzle, *The Datacenter as A Computer,* Morgan Claypool Publisher, 2009 [7])

Consider a datacenter built with 2,000 servers, each with 8 GB of DRAM and four 1-TB disk drives. Each group of 40 servers is connected through a 1-Gbps link to a rack-level switch that has an additional eight 1-Gbps ports used for connecting the rack to the cluster-level switch. It was estimated by Barroso and Holzle [9] that the bandwidth available from local disks is 200 MB/s, whereas the bandwidth from off-rack disks is just 25 MB/s via the shared rack uplinks. On the other hand, total disk storage in the cluster is almost ten million times larger than local DRAM. A large application that requires many more servers than can fit on a single rack must deal with these large discrepancies in latency, bandwidth, and capacity.

In a large scale datacenter, each component is relatively cheap and easily obtained from the commercial market. The components used datacenters are very different from those in building supercomputer systems. With a scale of thousands of servers, concurrent failure, either hardware failure or software failure, of tens of nodes is common. There are many failures that can happen in hardware, for example CPU failure, disk IO failure, and network failure etc. It is even quite possible that the whole datacenter does not work while facing the situation of power crash. And also, some failures are brought by software. The service and data should not be lost in failure situation. Reliable can be achieved by redundant hardware. The software must keep multiple copies of data in different location and keep the data accessible while facing hardware or software errors.

Cooling System of a Datacenter Room : Figure 7.9 shows the layout and cooling facility of a warehouse a

datacenter. The datacenter room has raised floor for hiding cables, power lines, and cooling supplies. The cooling system is somewhat simpler than the power system.. The raised floor has a steel grid resting on stanchions about 2–4 ft above the concrete floor. The under-floor area is often used to route power cables to racks, but its primary use is to distribute cool air to the server rack. The CRAC units pressurize the raised floor plenum by blowing cold air into the plenum.

This cold air escapes from the plenum through perforated tiles that are placed in front of server racks. Racks are arranged in long aisles that alternate between cold aisles and hot aisles to avoid mixing hot and cold air. The hot air produced by the servers re-circulates back to the intakes of the CRAC units that cool it and then exhaust the cool air into the raised floor plenum again. Typically, the incoming coolant is at 12–14°C and the warm coolant returns to a chiller. Newer datacenters often insert a cooling tower to precool the condenser water loop fluid .Water-based free cooling uses cooling towers to dissipate heat. The cooling towers use a separate cooling loop in which water absorbs the coolant's heat in a heat exchanger.

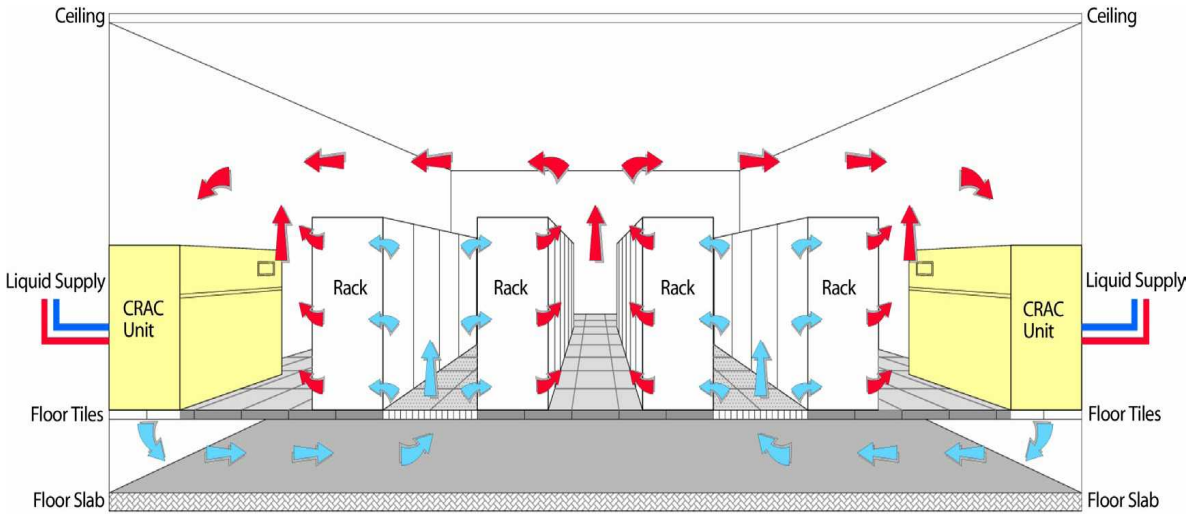


Figure 7.9 The cooling system in a raised-floor datacenter with hot-cold air circulation supported water heat exchange facilitiles (Courtesy of DLB Associates, D. Dyer, "Current Trends/Challenges in Datacenter Thermal Management, June 2006. [18]).

7.2.2 Datacenter Interconnection Networks

A critical core design of a datacenter is the interconnection network among all servers in the datacenter cluster. This network design must meet five special requirements: *low latency, high bandwidth, low cost, support MPI communications*, and *faul-tolearnt.* The design of an inter-server network must satisfy both point-to-point and collective communication patterns among all server nodes. Specific design considerations are given below:

Application Traffic Support: The network topology should support all MPI communication patterns.Both point-to-point and collective MPI communications must be supported. The network should have high bi-section bandwidth to meet this requirement. For example, one-to-many communications are used for supporting distruibute file accesses. One can use one or a few servers as metadata master servers which need to communicate with slave server nodes in the cluster. To support the MapReduce programming paradigm, the network must be designed to perform the map and reduce functions (to be treated in Chapter 7) in high speed. In other word, the underline network structure should support various network traffic patterns demanded by user applications.

Network Expandability : The interconnection network should be expandable. With thousands or even hundreds of thousands of server nodes, the cluster network interconnection should be allow to expand, once more servers are added into a datacenter. The network topology should be restructured while facing such an expected growth in the future. Also the network should be designed to support load balancing and data movement among the servers. None of the links should become a bottleneck to slow down the application performance. The topology of the interconnection should avoid such bottlenecks.

The fat-tree and crossbar, networks studied in Chapter 3 could be also implemented with low-cost Ethernet switches. However, the design could be very challenging when the number of the servers increases shapely. The most critical issue on the expandability is the support of modular network growth for building datacenter containers in Section 6.2.3. One single datacenter container contains hundreds of servers and is considered to be the building block of large-scale datacenters. The network interconnection among many container will be treated in Section 6.2.4. In other words, we design not only the cluster network for container datacenter, but also considering the cable connection among multiple datacenter containers.

Datacenters are not built by piling up servers in multiple racks now. Instead, the datacenter owners buy server containers while each container contains several hundred even thousands of server nodes. The owners can just plug-in the power supply, outside connection link as well as cooling water and the whole system can just go and work. This is quite efficient and reduces the cost of purchasing and maintaining of the servers. One approach is to establish the connection backbone first and them extend the backbone links to reach out the end servers. Another approach is to connect multiple containers through external switching and cabling as detailed in Section 6.2.4. .

Fault Tolerance and Graceful Degradation: The interconnection network should provide some mechanism to tolerate link or switch failures. Multiple paths should be established between any two server nodes in a datacenter. Fault tolerant of servers is achieved by replicating data and computing among redundant servers. Similar redundancy technology should apply to the network structure. Both software and hardware network redundancy apply to cope with potential failures. One the software side, the software layer should be aware of network failure. Packets forwarding should avoid using the broken links. The network support software drivers should handle this transparently without affecting the cloud operations.

In a datacenter network, failures and server corruptions are quite common. The network structure should degrade gracefully amid limited failures. Hot swappable components are desired. There should be no critical paths or critical points which may become a single point of failure that pull ,down the entire system. In the research frontier, efficient and dependable datacenter networks have been a hot topic in IEEE Infocom and Globecom Conference. Most design innovation lies in topology structure of the network. The network structure is often divided into two layers. The lower layer is close to the end servers. The upper layer establish the backbone connections among the server groups or subclusters. This hierarchical interconnection approach appeals to building datacenters with modular containers..

Switch-centric Datacenter Design : Currently, there are two approaches to building datacenter-scale networks : One is switch-centric and the other is server-centric. In a switch-centric network, the switches are used to connect the server nodes. The switch centric design does not affect the server side. No modifications to the servers are needed. The server-centric design does modify the operating system running on servers. Special drivers are designed for relaying the traffic. Switches still have to be organized for achieving the connections.

In Fig. 7.10, a fat-tree switch network design is presented for datacenter construction. The fat-tree topology is applied to interconnect the server nodes. The topology is organized in two layers. Server nodes are in the bottom layer. *Edge switches* are used to connect the nodes in the bottom layer. The upper layer aggregate the lower-layer edge switches. A group of aggregation switches, edge switches and their leaf nodes form a *pod*. On top of the pods are the *core switches*. Core switches provide paths among different

pods. The fat-tree structure provides multiple paths between any two server nodes in the datacenter. This provides fault-tolerant capability with alternat path in case of some isolated link failures.

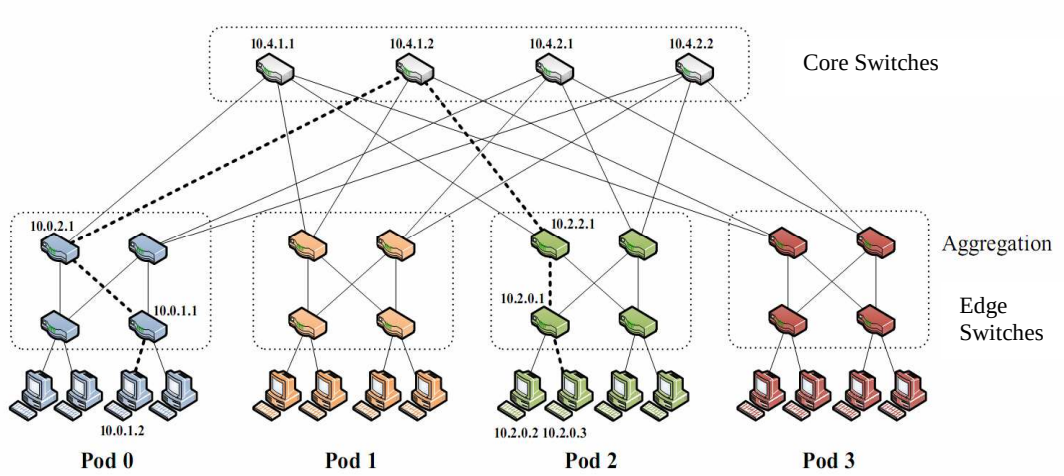


Figure 7.10 A fat-tree interconnection topology for scalable datacenter construction. (Courtesy of M. Al-Fares, et al, "A Scalable, Commodity Datacenter Network Architecture," *Proc. of the ACM SIGCOMM 2008 Conf. on Data Communication,* Seattle, WA, August 17–22, 2008 [2]).

As a matter of the fact, the failure of an aggregation switch and core switch will not affect the connectivity of the whole network. The failure of any edge switch can only affect a small number of end server nodes. The extra switches in a pod provide higher bandwidth to support cloud applications in massive data movement. The building blocks used in the fat-tree are the low-cost Ethernet switches. This reduces the cost quite a bit. However, traditional IP/Ethernet router switch only provides one single route from the source to destination. The design must overcome this difficulty by adding redundant switches in each pod. They have modified the routing table inside the switches to provide extra routing paths in case of switch or link failure. The modifications of routing table and routing algorithms are built inside the switches. The end server nodes in the datacenter are not affected during one switch failure, as long as one of the alternate routing path does not fail at the same time.

7.2.3 Modular Datacenters in Containers

A modern datacenter is structured as a shipyard of server clusters housed in truck-towed containers. Figure 7.11 shows the interior details of a truck container of server cluster. Inside the container, hundreds of blade servers are housed in racks surrounding the container walls. An array of fans forces the heated air generated by server racks to go through a heat exchanger, which cools the air for the next rack (detail in callout) on a continuous loop. A single container can house a datacenter with a capacity to process 7 Terabytes of data with 2 Petabytes of storage. Modern datacenters are becoming a shipping yard of container trucks

The modular datacenter in container trucks was motivated by the demand of lower power consumption, higher computer density, and mobility to relocate datacenters to better locations with lower cost in electricity, better cooling water supplies, and cheaper housing for maintenance engineers. Both chilled air circulation and cold water are flowing through the heat exchange pipes to keep the server racks cool and easy to repair. Datacenters usually are built on the ground where lease and utility for electricity is cheaper, and cooling is more efficient.

Both warehouse-scale and modular datacenters in containers are needed. In fact, the modular truck containers can be used to put together a large-scale datacenter like a container shipping yard. In addition to location selection and power saving in datacenter operations, one must consider the data integrity, server

monitory, and security management in datacenters. These problems are easier to handle if the datacenter is centralized in a single large building.

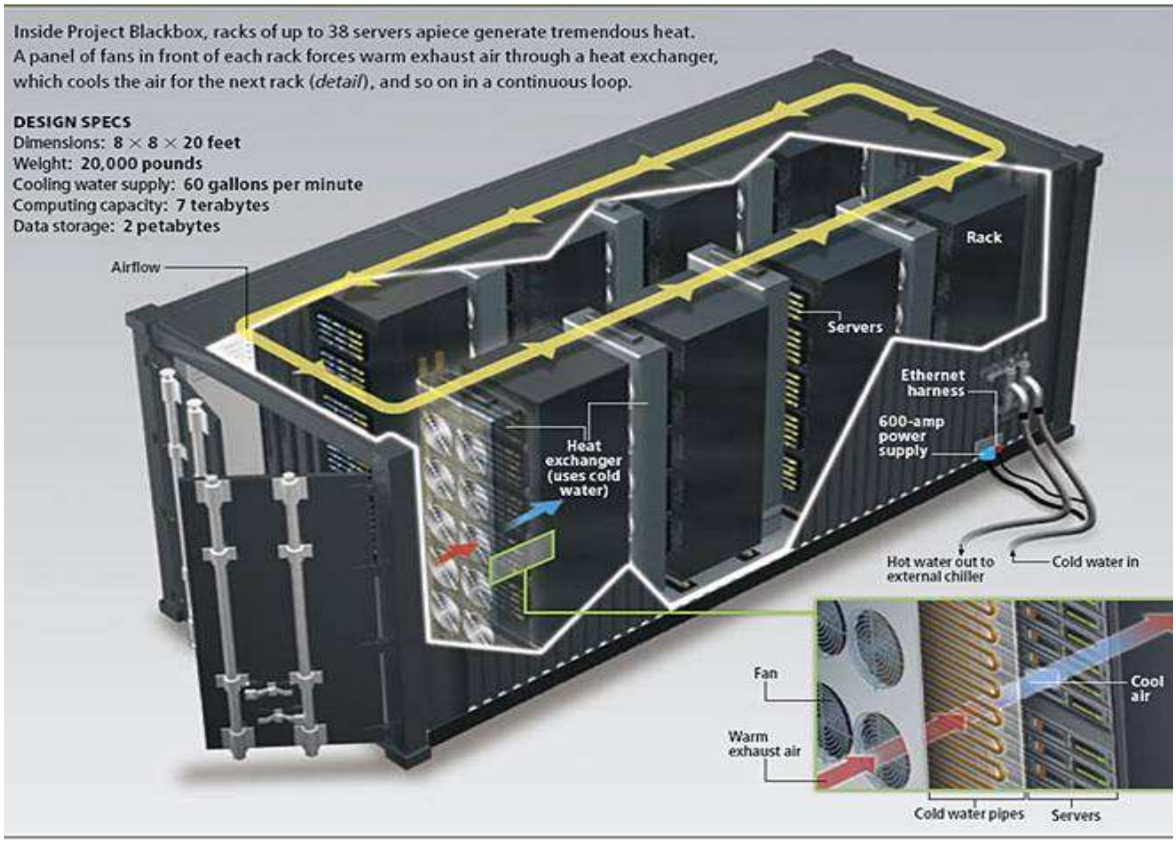


Figure 7.11 The layout of a datacenter built inside a truck container cooled by chilled air circulation with cold-water heat exchanges. (Courtesy of HP Project Blackbox, 2008).

Container Datacenter Construction: The datacenter module is housed in a container. The modular container design include the network gear, compute, storage, and cooling. Just plug-in power, network, and chilled water, the datacenter should work. One needs to increase the cooling efficiency by varying the water and air flow with better air flow management. Another concern is to meet the seasonal load requirements. The construction of the container-based dataceneter may start with one system (server), them move to rack system design and finally the container system. The staged development may take different amounts of time and demand an increasing cost. For example, building one server system may take a few hours in racking and networking. Building a rack of 40 servers may take a half day's effort.

Extending to a whole container system with multiple racks for 1,000 servers requires proper layout of the floor space with power, networking and cooling and complete testing. The container must be designed to be weatherproof and easy to transport. Datacenter construction and testing may take a few days to complete if all components are available and power and water supplies are handy. The regulatory approval of electrical and mechanical properties may take additional effort. The modular datacenter approach supports many cloud service applications. For example, the health-care industry will benefit more by installing their datacenter at all clinic sites. However, how to exchange information with the central database and maintain periodic consistency becomes a rather challenging design issue in a hierarchically structured datacenter. The security of co-location cloud services may demands multiple containers, which is much more complex than installing on a single container.

7.2.4 Interconnection among Modular Datacenters

In Fig.7.12, Guo, et al have developed a server-centric BCube network for interconnecting modular datacenters. The servers are represented by circles and switches by rectangles. The BCube provides a layered structure. The bottom layer contains all the server nodes and they form the level-0. Level-1 switches form the top layer of BCube0. BCube is a recursively constructed structure. The $BCube_0$ consists of n servers connecting to an n-port switch. The $BCube_k$ ($k \geq 1$) is structured from n $BCube_{k-1}$ with n^k n-port switches. The example of BCube1 is illustrated, where the connection rule is that the i-th server in the j-th $BCube_0$ connects to the j-th port of the i-th level-1 switch. The servers in the BCube have multiple ports attached. This allows extra devices to be used in the server. However, as the Ethernet switches continue lowered in price, this will not add too much additional cost.

The BCube provides multiple paths between any two nodes in the datacenter. In addition, the failure of any nodes (either server or switch) will not break the connectivity of the entire network. Multiple paths provide extra bandwidth to support various communication patterns in different cloud applications. The BCube provides a kernel module in the server operating system to perform the routing operations. The kernel module supports packet forwarding while the incoming packets are not destined at the current node. Such modification of kernel will not influence the upper layer applications. Thus, the cloud application can still run on top of the BCube network structure without any modification.

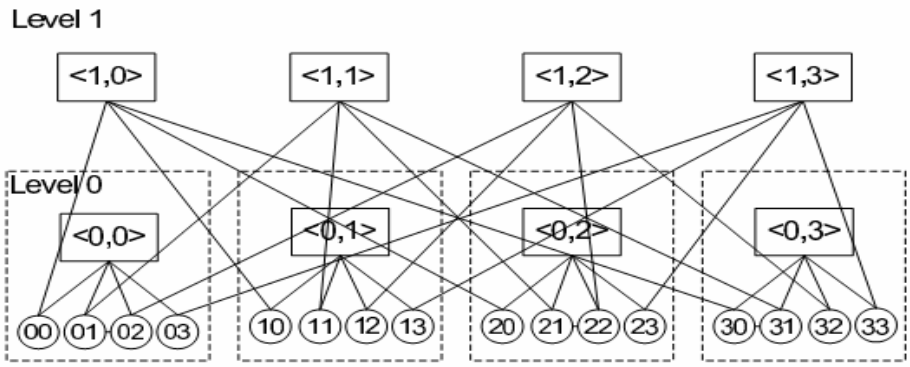


Figure 7.12 BCube: A High Performance, server-centric network for modular datacenters. (Courtesy of C. Guo, et al, "BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers," *ACM SIGCOMM Computer Communication Review, Oct. 2009.* [25]).

The BCube is commonly used inside a server container. The containers is considered as the building blocks for datacenters. Thus, despite the design of the inner container network, we need another level of networking among multiple containers in the datacenter. In Fig.7.13, Wu, et al [68] have proposed a network topology for inter-container connection using the above BCube network as building blocks. The proposed network was named as MDCube (*Modularized Datacenter Cube network*). This network connect multiple BCube containers by using a high-speed switches in BCube. Similarly, the MDCube is constructed by shuffle network with multiple containers.

Figure 6.13 shows how a 2-D MDCube is constructed from nine (9 =3x3) $BCube_1$ containers. In fact, there are many other ways of using MDCube to build the network. Essentially, this network architecture builds a virtual hypercube at the container level, in addition to the cube structure inside the container (BCube). With the server container built with the BCube network, the MDCube is used to build large-scale datacenter for supporting cloud application communication patterns**.** Readers are referred to the article [45] for detailed implementation and simulation results of this interconnection network over multiple modular datacenters built in containers.

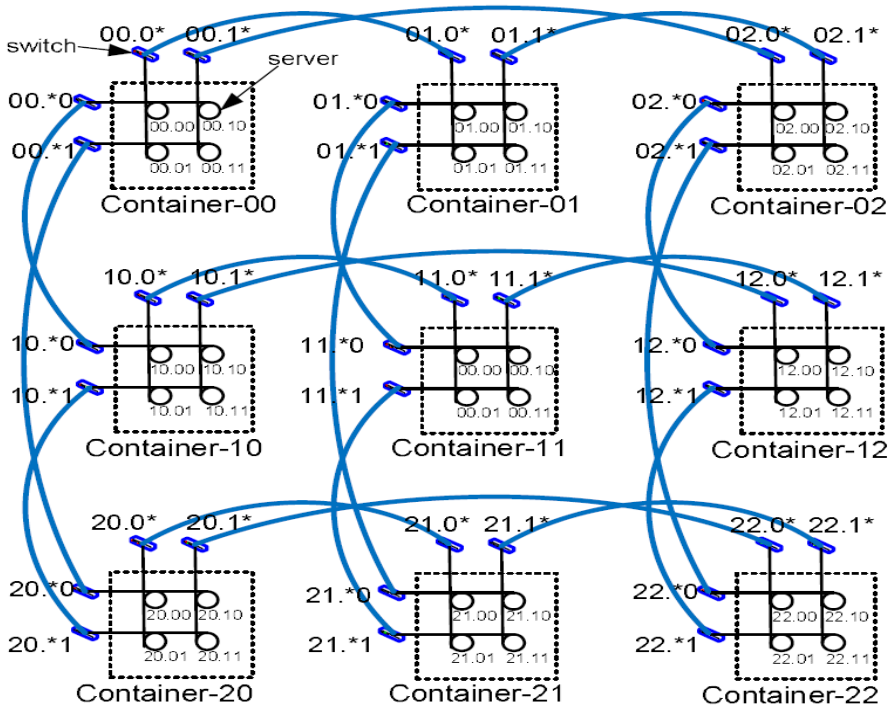


Figure 7.13 A 2-D MDCube is constructed from 9=3*3 BCube$_1$ Containers. (Courtesy of H. Wu, et al, "MDCube: A High Performance Network Structure for Modular ta Center Interconnection", *ACM CoNEXT'09*, Dec. 2009, Rome, Italy [68]).

7.2.5 Datacenter Management Issues

This involves the management of hardware, software, database, resources, and security of a datacenter. Listed below are basic requirements in managing a datacenter.

- ***Making the Common Users Happy:*** The system should be designed to provide quality service to the majority of the users for at least 30 years.
- ***Controlled Information Flows:*** Information flow should be streamlined. Sustained services and high availability are the primary goals.
- ***Multi-User Manageability :*** The system must be managed to support all functions of a datacenter, including traffic flows, database updating, server maintenance, etc.
- ***Scalability in Database Growth:*** The system should allow growth as workload increases. The storage, processing, I/O, power, and cooling subsystems should be all scalable.
- ***Reliability in Virtualized Infrastructure:*** Failover, fault-tolerance, and VM live migration should be integrated to enable recovery of critical data and applications from failures or disasters.
- ***Lowered Costs to both Users and Providers :*** Reducing the cost of both users and providers of the cloud system built over the datacenters including all operational costs.
- ***Security Enforcement and Data Protection:*** Data privacy and security defense mechanisms must deployed to protect the datacenter against network attacks and system interrupts and maintain data integrity from user abuses or network attacks.
- ***Green Information Technology :*** Saving power consumption and upgrading energy efficiency are very much in demand in designing and operating current and future datacenters

Example 7.2 Google Datacenter Health Monitoring Infrastructure

The Google *System Health* infrastructure is shown in Fig.7.14. This system monitors all servers in their configuration, activity, environmental, and error conditions. The system health module stores this information as a time series in a scalable repository. The MapReduce software is applied in various data analysis. For example, the MapReduce applies in an automated machine failure diagnosis. Machine learning methods are applied to suggest the most appropriate repairs action to take after some detected sever failures. With hundreds or thousands of servers in the datacenter. This monitoring system is itself a supercomputing system. ■

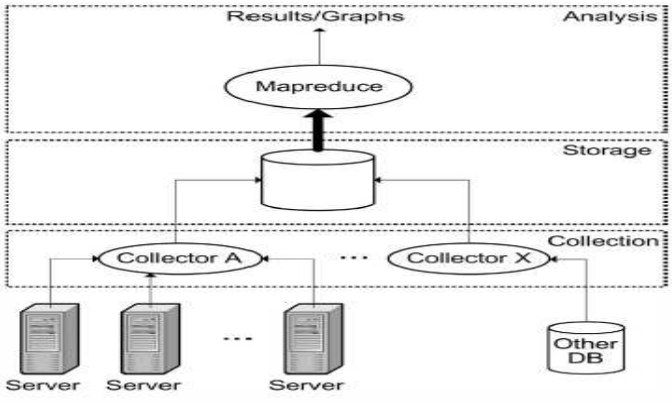


Figure 7.14 Google Datacenter health monitoring system (Courtesy of Barroso and Holzle, 2009 [7])

Marketplaces in Cloud computing Services : The container-based datacenter implemented can be done more efficiently with factory racking, stacking and packing One should avoid layers of packaging at customer site. However, the datacenters are still custom crafted rather than prefab units. The modular approach is more space efficient with power densities in excess of 1250 W/sq ft. Rooftop or parking lot installation is acceptable. One should leave sufficient redundancy to allow upgrades over time. Figure 7.15 shows the projected changes of datacenter cost from 2009 to 2013. In 2209, the global cloud service marketplace reached $17.4 billions. Acxcording to IDC estimate in 1010, this cloud-based economy may increase to $44.2 billions by 1013.

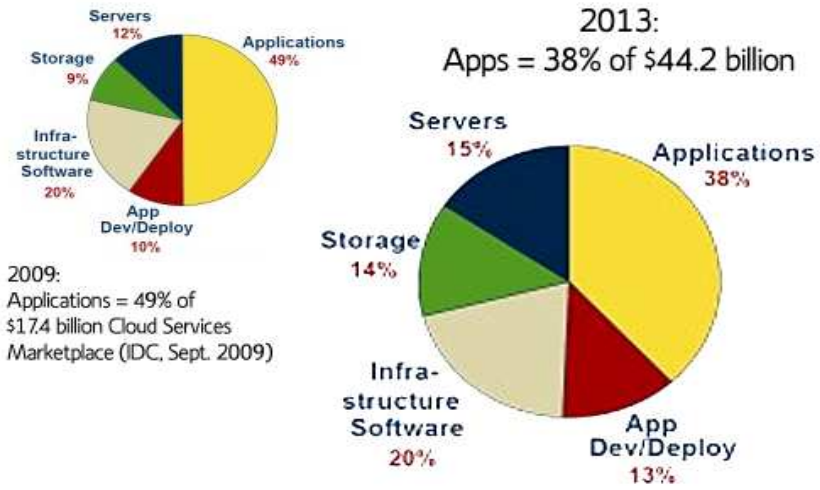


Figure 7.15 Projected growth of cloud service marketplace (IDC projection 2009)

7.3 Architectural Design of Computing Clouds

This section presents basic cloud design principles. We start with a basic cloud architecture to process massive data with a high-degree parallelism. Then we study virtualization support, resource provisioning, infrastructure management, and then performance modeling.

7.3.1 Cloud Architecture for Distributed Computing

An Internet cloud is envisioned as a public cluster of servers provisioned on demand to perform collective web services or distributed applications using the datacenter resources. The cloud design objectives are first specified below. Then we present a basic cloud architecture design..

Cloud Platform Design Goals: *Scalability, virtualization, efficiency,* and *reliability* are four major design goals of a cloud computing platform. Clouds support Web 2.0 applications. The cloud management receives the user request and then finds the correct resources, and then calls the provisioning services which invoke resources in the cloud. The cloud management software need to support both physical and virtual machines. Security in shared resources and shared access of datacenters also post another design challenge.

The platform needs to establish a very large-scale HPC infrastructure. The hardware and software systems are combined together to make it easy and efficient to operate. The system scalability can benefit from cluster architecture. If one service takes a lot of processing power or storage capacity or network traffic, it is simple to add more servers and bandwidth. The system reliability can benefit from this architecture. Data can be put into multiple locations. For example, the user email can be put in three disks which expand to different geographical separate data centers. In such situation, even one of the datacenters crashes, the user data is still accessible. The scale of cloud architecture can be easily expanded by adding more servers and enlarging the network connectivity accordingly.

Enabling Technologies for Clouds: The key driving forces behind cloud computing are the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software. Cloud users are able to demand more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers can increase the system utilization via multiplexing, virtualization, and dynamic resource provisioning. Clouds are enabled by the progress in hardware, software and networking technologies summarized in Table 7.1.

Table 7.1 Cloud Enabling Technologies in Hardware, Software, and Networking

Technology	Requirements and Benefits
Fast Platform Deployment	Fast, efficient, and flexible deployment of cloud resources to provide dynamic computing environment to users
Virtual Clusters on Demand	Virtualized cluster of VMs provisioned to satisfy user demand and virtual cluster reconfigured as workload changes.
Multi-Tenant Techniques	SaaS distributes software to a large number of users for their simultaneous uses and resource sharing if so desired.
Massive Data Processing	Internet search and web services often require massive data processing, especially to support personalized services
Web-Scale Communication	Support e-commerce, distance education, telemedicine, social networking, digital government, and digital entertainment, etc.
Distributed Storage	Large-scale storage of personal records and public archive information demand distributed storage over the clouds
Licensing and Billing Services	License management and billing services greatly benefit all types of cloud services in utility computing.

These technologies play instrumental roles to make cloud computing a reality. Most of these technologies are mature today to meet the increasing demand. In the hardware area, the rapid progress in multi-core CPUs, memory chips, and disk arrays has made it possible to build faster datacenters with huge storage space. Resource virtualization enables rapid cloud deployment faster and fast disaster recovery. *Service-oriented architecture* (SOA) also plays a vital role. The progress in providing *Software as a Service* (SaaS), Wed.2.0 standards, and Internet performance have all contributed to the emergence of cloud services. Today's clouds are designed to serve a large number of tenants over massive volume of data. The availability of large-scale, distributed storage systems lies the foundation of today's datacenters. Of course, cloud computing is greatly benefitted by the progress made in license management and automatic billing techniques in recent years

A Generic Cloud Architecture : A security-aware cloud architecture is shown in Fig.7.16. The Internet cloud is envisioned as a massive cluster of servers. These servers are provisioned on demand to perform collective web services or distributed applications using datacenter resources. Cloud platform is formed dynamically by provisioning or de-provisioning, of servers, software, and database resources. Servers in the cloud can be physical machines or virtual machines. User interfaces are applied to request services. The provisioning tool carves out the systems from the cloud to deliver on the requested service.

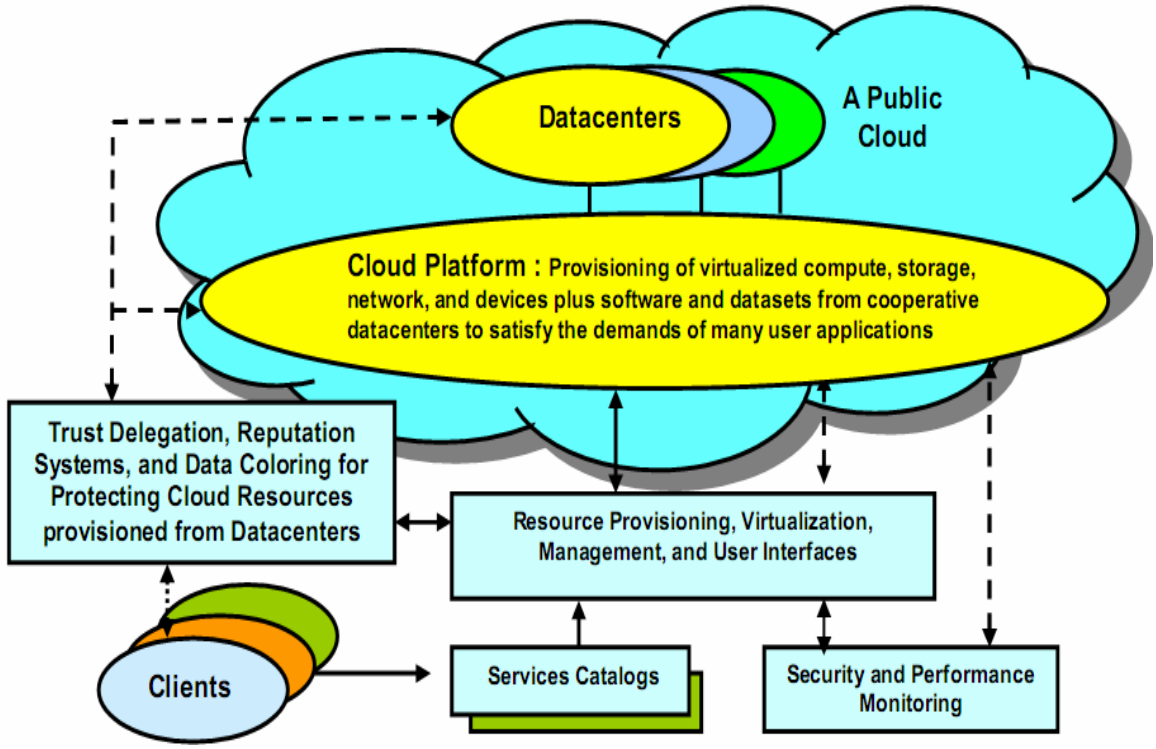


Figure 7.16 A security-aware cloud platform built with a virtual cluster of virtual machines, storage, and networking resources over the datacenter servers operated by providers.

In addition to building the server cluster, cloud platform demand distributed storage and accompanying services. The cloud computing resources are built in datacenters, which are typically owned and operated by a third-party provider. Consumers do not need to know the underlying technologies. In a cloud, software becomes a service. The cloud demands a high-degree of trust of massive data retrieved from large datacenters. We need to build a framework to process large scale data stored in the storage system. This demands a distributed file system over the database system. Other cloud resources are added into a

cloud platform including the storage area networks, database systems, firewalls and security devices. Web service providers offer special APIs that enable developers to exploit Internet clouds. Monitoring and metering units are used to track the usage and performance of resources provisioned

The software infrastructure of a cloud platform must handle all resource management and do most of the maintenance, automatically. Software must detect the status of each node, server joining and leaving and do the tasks accordingly. Cloud computing providers, like Google and Microsoft, have built a large number of datacenters all over the world. Each datacenter may have thousands of servers The location of the datacenter is chosen to reduce power and cooling costs. Thus, the datacenters are often built around hydroelectricity power stop. The cloud physical platform builder concerns more about the performance/price ratio and reliability issues than the shear speed performance.

In general, the private clouds are easier to manage. Public clouds are easier to access. The trends of cloud development is that more and more clouds will be hybrid. This is due to the fact that many cloud application must go beyond the boundary of an Intranet. One must learn how to create a private cloud and how to interact with the public clouds in the open Internet. Security becomes a critical issue in safeguard the operations of all cloud types. We will study the security and privacy issues of cloud services in Chapter 7.

7.3.2 Layered Cloud Architectural Development

The architecture of a cloud is developed at three layers: *infrastructure, platform,* and *application* as demonstrated in Fig.7.17. These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud. The services to public, private, and hybrid clouds are conveyed to users through the networking support over the Internet and intranets involved. It is clear that the infrastructure layer is deployed first to support IaaS type of services. This infrastructure layer serves as the foundation to build the platform layer of the cloud for supporting PaaS services. In turn, the platform layer is a foundation to implement the application layer for SaaS applications. Differnert types of cloud services demand to apply the resources at the , separately.

The infrastructure layer is built with virtualized compute, storage and network resources. The abstraction of these hardware resources is meant to provide the flexibility demanded by users. Internally, the virtualization realizes the automated provisioning of resources and optimizes the infrastructure management process. The platform layer is for general-purpose and repeated usage of the collection of software resources. This layer provides the users with an environment to develop their applications, to text the operation flows, and to monitor the execution results and performance. The platform should be able to assure the users with scalability, dependability, and security protection. In a way, the virtualized cloud platform serves as a "system middleware" between the infrastructure and application layers of the cloud

The application layer is formed with a collection of all needed software modules for SaaS applications. . Service applications in this layer include daily office management work, such as information retrieval, document , processing, and calendar and authentication services, etc. The application layer is also heavily used by enterprises in business marketing and sales, *consumer relationship management* (CRM), financial transactions, supply chain management, etc. It should be noted that not all cloud services are restricted to a single layer. Many applications may apply resources at mixed layers. After all, the three layers are built from bottom up with a dependence relationship.

From the provider's perspective, the services at various layers demand different amounts of function support and resource management by the providers. In general, the SaaS demands the most work from the provider, the PaaS in the middle, and IaaS the least. For an example, Amazon EC2 provides not only virtualized CPU resources to users but also the management of these provisioned resources. Services at the application layer demands more work from the providers. The best example is the Salesforce CRM service in which the provider supplies not only the hardware at the bottom layer and the software at the top layer,

but also provides the platform and software tools for user application development and monitory.

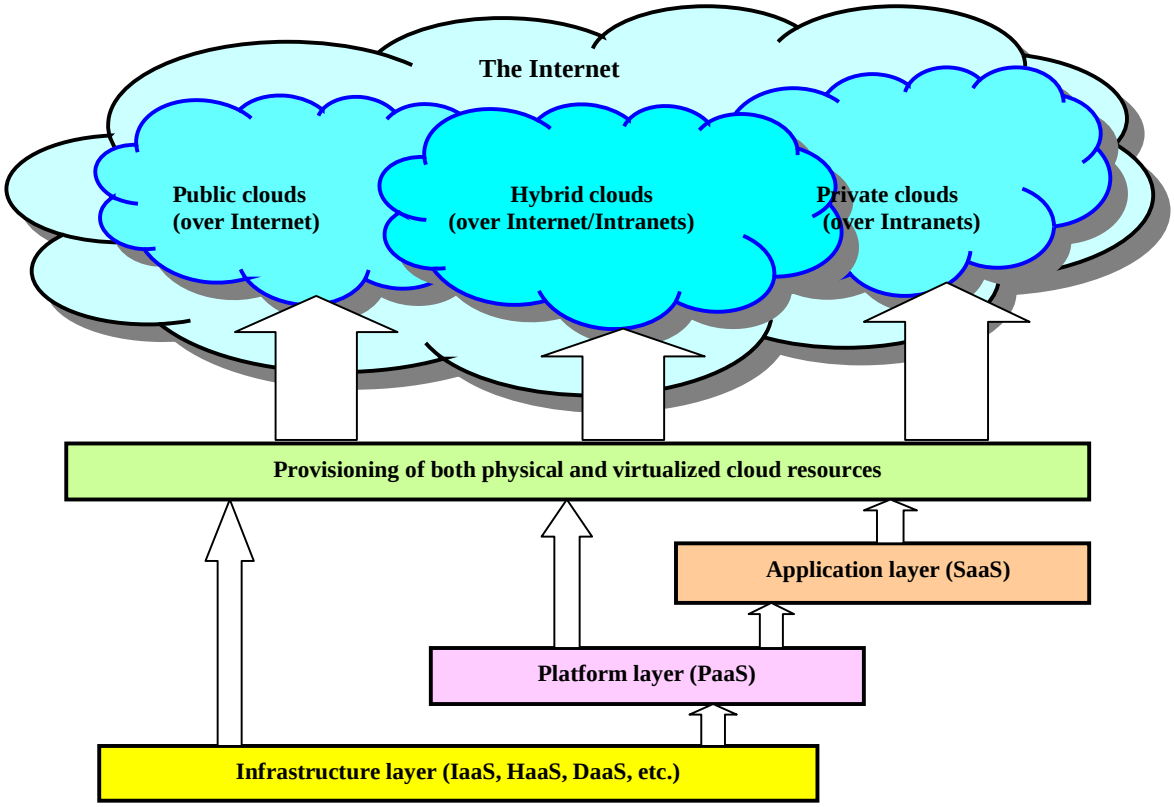


Figure 7.17 Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet and intranets.

7.3.3 Virtualization Support and Disaster Recovery

One very distinguish feature of cloud computing infrastructure is the use of system virtualization and the modification to provisioning tools. Virtualization of servers on a shared cluster can consolidate the web services. As the virtual machines are the container of cloud services, the provisioning tools will first find the corresponding physical machines and deploy the virtual machines to those nodes before scheduling the service to run on the virtual nodes. In cloud computing, virtualization is not only means the system virtualization platforms are uses but also some fundamental services are provided.

In addition, in cloud computing, virtualization also means the resources and fundamental infrastructure is virtualized. From the user point of view, the user will not care about the computing resources that are used for providing the services. Cloud users do not need to know and have no way to discover physical resources that are involved while processing a service request. And also, from the developer point of view, the application developers are not care about some infrastructure issues such as scalability, fault tolerant i.e. they are virtualized. Application developers focus on the service logic. Figure 6.14 shows the infrastructure needed to virtualize the servers in a datacenter for implementing specific cloud applications.

System Virtualization: In many cloud computing systems, system virtualization software is used. System virtualization software is a special kind of software which simulates the execution of hardware and run even the unmodified operating systems. Cloud computing systems use the virtualization software as the running environment for legacy software such as old operating systems and unusual applications. Virtualization software is also used as the platform for developing new cloud applications that the developers can use any

operating systems and programming environments as they like. The development environment and deployment environment can now be the same which eliminates some runtime troubles. The system virtualization supported is illustrated in Fig.7.18 with a virtualization infrastructure.

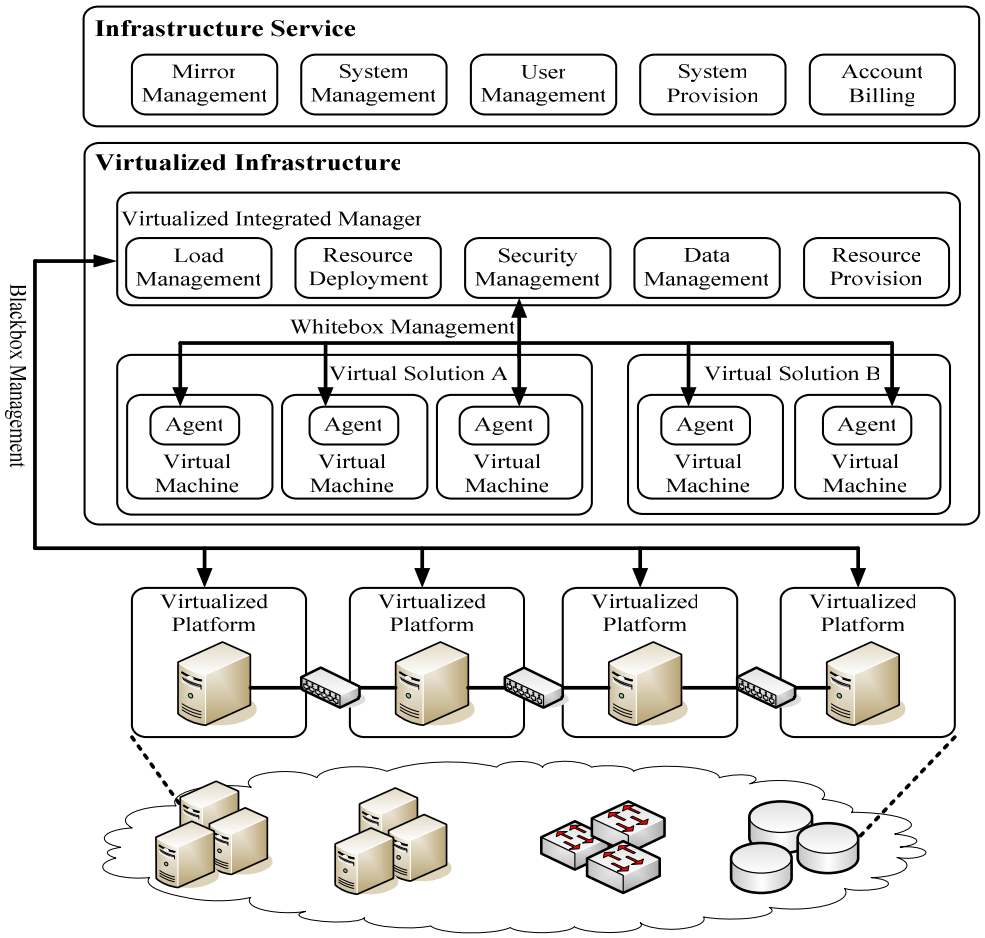


Figure 7.18 Virtualized servers, storage, and network for cloud platform construction

Some cloud computing providers have used the virtualization technology to provide the service for the developers. As mentioned before, the system virtualization software is considered as the hardware analogue mechanism to run unmodified operating system, usually run on bare hardware directly, on top of software. The current widely used system virtualization software is listed in Table 6.3. Currently, the VMs installed at a cloud computing platform are mainly used for hosting third-party programs. Virtual machines provide the flexible runtime services that the users do not have to worry about the whole system environment.

By using the virtual machines in cloud computing platform, extreme flexible can be brought to the users. As the computing resources are shared by many users, there needs a method to maximize the user's privilege and still keep them separate safely. Traditional sharing of cluster resources depends on the user and group mechanism on a system. Such sharing is not flexible. Users can not customize the system for their special purposes. Operating system cannot be changed. The separate is not complete. There is some affection among users. The environment meet one user's requirement often cannot satisfy another user. Virtualization provides the method to let the user have full privilege while keep them separately. Users have full access to their own VMs, while are completely separated from other user's VMs.

Multiple VMs are installed on a physical server. Different VMs may run with different OSs. We need also establish virtual disk storages and virtual networks needed by the VMs. The virtualized resources form a resource a pool. The virtualization is carried out by special servers dedicated to generate the virtualized resource pool. The virtualized infrastructure (blackbox in the middle) is built with many *virtualizing integration managers*. These managers handles load, resources, security, data, and provisioning functions. Two VM platforms are shown in Fig.6.18. Each platform carries out a virtual solution of a user job. All cloud services are managed at the top boxes

Virtualization Tools: In Table 7.2, we summarize some software tools for system virtualization. These tools are developed by three major software providers. The VMware tools apply to workstations, servers, and virtual infrastructure. The Microsoft tools are used on PCs and some special servers. The XenEnterprise tool applies only to XEN-based servers. Everyone is interested in the cloud, the entire IT industry is moving towards the vision of cloud. Virtualization, with its core benefits like High Availability, Disaster Recovery, Dynamic Load Leveling, On-the-fly resource configuration and rich provisioning support, is seen as the core backend infrastructure for the clouds. Cloud computing and utility computing will essentially leverage the benefits of virtualization to provide a more robust, scalable and autonomous computing environments.

Table 7.2 Some System Virtualization Software Tools

Provider	System Virtualization Software Name
VMware	VMware Workstation
	VMware Server
	VMware ESX Server (Virtual Infrastructure)
Microsoft	Virtual PC
	Hyper-V Server
XenEnterprise	XenServer

Storage Virtualization for Green Datacenters: IT power consumption in the US has more than doubled to 3% of the total energy consumed in the US. Large number of datacenters have contributed to a great extent of this energy crisis. Over half of the companies in the Fortune 500 are actively implementing new corporate energy policies. Recent Survey from, IDC and Garnter confirm the fact that the virtualization had great impact in cost reduction from reduced power consumption in physical computing systems. This alarming situation has waked up the IT Industry to become energy-aware. With little evolution of alternate energy resources, there is an imminent need to converser the power in all computers. Virtualization and server consolidation have already proven handy in this aspect. Green datacenters and benefits of storage virtualization are considered to further strengthen the synergy of green computing.

Virtualization for Supporting IaaS: VM technology increases ubiquity. This has enabled users to create customized environments atop physical infrastructure for cloud computing. Use of VMs in cloud has the following distinct benefits: (1). System administrators consolidate workloads of underutilized servers in a fewer servers. Machines. (2). VMs have the ability to run legacy code without interfering with other APIs. (3) VMs can be used to improve security through the creation of sandboxes for running applications with questionable reliability. (4). Virtualized cloud platform can apply performance isolation, letting providers offer some guarantees and better quality of service to customer applications.

VM Cloning for Disaster Recovery : *Virtual machine* (VM) technology requires advanced disaster recovery scheme. One scheme is to recover a *physical machine* (PM) by another PM. The second scheme is

to recover a VM by another VM. As shown in the top time row of Fig.7.19 the traditional disaster recovery from PM to PM is rather slow, complex, and expensive. The total recovery time is attributed to the hardware configuration, installing and configuring the OS, installing the backup agents, and the long time to restart the PM. To recover a VM platform, the installing and configuration times for OS and backup agents are eliminated. Therefore, we end up with a much shorter disaster recovery time, about 40% of that to recover the PMs. Virtualization aids to the fast disaster recovery by VM encapsulation.

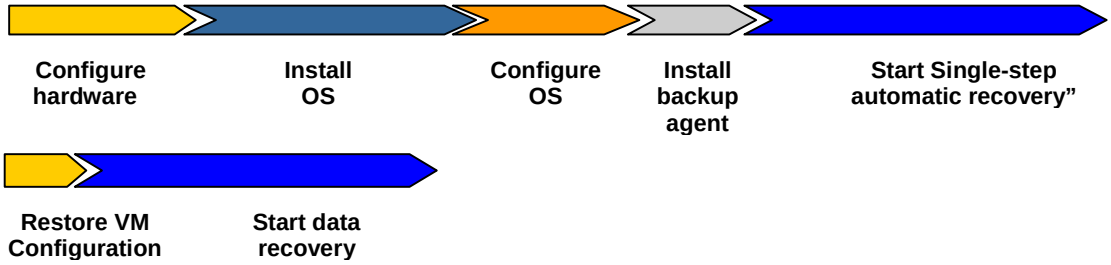


Figure 7.19 Recovery overhead of a conventional disaster recovery between physical machines, compared with that required to recover from live migration of virtual machines

We have learned the basics of disaster recovery in Chapters 2 and 3. The cloning of VMs offer an effective solution. The idea is to make a clone VM on a remote server for every running VM on a local server. Among all clone VMs, only one needs to be active. The remote VM should be in a suspended mode. A cloud control center should be able to activate this clone VM in case of failure of the original VM. Taking a snapshot of the VM to enable live migration with minimum time. The migrated VM run on a shared Internet connection. Only updated data and modified states are sent to the suspended VM to update its state. The RPO (*Recovery Property Objective*) and RTO (*Recovery Time Objective*) are affected by the amount of snapshots taken. Security of the VMs should be enforced during the live migration of VMs.

Virtualization Support in Public Clouds: Armbrust et al [2] have assessed in Table 7.3 three public clouds in the context of virtualization support: namely the *Amazon Web Service* (AWS), *Microsoft Azure,* and *Google App Engine* (GAE). The AWS provides extreme flexibility (virtual machines) for the users to execute their own applications. GAE provides limited application level virtualization for the users to build applications only based on the services that are created by Google. Microsoft Azure sits in the middle that it provides the programming level virtualization (.Net virtualization) for the users to build their applications. Thus the flexibility provided by Azure is also between the capability provided by Google and Amazon.

Table 7.3 Virtualized resources in Compute , Storage, and Network Clouds

(Courtesy of Armbrust, et al , "Above the Clouds: A Berkeley View of Cloud Computing [4].)

Provider	**Amazon Web Services (AWS)**	**Microsoft Azure**	**Google AppEngine (GAE)**
Compute Cloud with virtual cluster of servers	x86 instruction set, Xen VMs, resource elasticity allows scalability through virtual cluster, or a third party such as RightScale must provide the cluster.	Common language runtime VMs provisioned by declarative descriptions	Predefined application framework, handlers written in Python, Automatic scaling up and down, server failover in consistent with Web applications
Storage Cloud with virtual storage	Models for block store (EBS) and augmented key/blob store (SimpleDB), Automatic scaling varies from EBS to fully automatic (SimpleDB, S3)	SQL Data Services (restricted view of SQL Server), Azure storage service	MegaStore/BigTable
Network Cloud Services	Declarative IP level topology; placement details hidden, Security groups restricting communication, Availability zones isolate network failure, Elastic IP applied.	Automatic based on programmer's declarative descriptions of app components (roles)	Fixed topology to accommodate 3-tier Web app structure, Scaling up and down is automatic and programmer invisible

7.3.4 Data and Software Protection Techniques

In this section, we study a data coloring technique to preserve data integrity and user privacy. Then we show a watermarking approach to protect software files being largely distributed in a cloud environment.

Data Integrity and Privacy Protection : We desire a software environment that provides many useful tools to build cloud applications over large datasets. In addition to MapReduce, BigTable, EC2, 3S, Hadoop, AWS, AppEngine, and WebSphere2, we identify below some security and privacy features desired by cloud users.

- Special APIs for authenticating users and sending email using commercial accounts.
- Fine-grain access control is desired to protect data integrity and deter intruders or hackers.
- Shared datasets are protected from malicious alteration, deletion, or copyright violation
- Securing the ISP *or cloud service providers (*CSP) from invading user privacy.
- Personal firewalls at user ends. Keep shared datasets from Java, JavaScript, and ActiveX Applets
- Privacy policy consistent with CSP's policy. Protect against identity theft, spyware, and web bugs.
- VPN channels between resource sites to secure transmission of critical data objects.

Data Coloring and Cloud Watermarking : With shared files and datasets, privacy, security, and copyright could be compromised in a cloud-computing environment. We desire to work in a trusted software environment that provides useful tools to build cloud applications over protected datasets. In the past watermarking was mainly used for digital copyright management. Collberg and Thomborson [16] have suggested the use of watermarking to protect software. The "cloud" trust model proposed by Li et al [37] offers Type-2 fuzzy membership. We apply this model to generate data coloring to protect large datasets in the cloud. Readers should not be confused with the membership "clouds" from the term "cloud computing". Forward and backward cloud generation processes are illustrated in Fig.7.20 based on type-2 fuzzy logic. The cloud drops (data color) are added in the left photo and removed to restore the original photo on the right. This process is called data coloring or data watermarking.

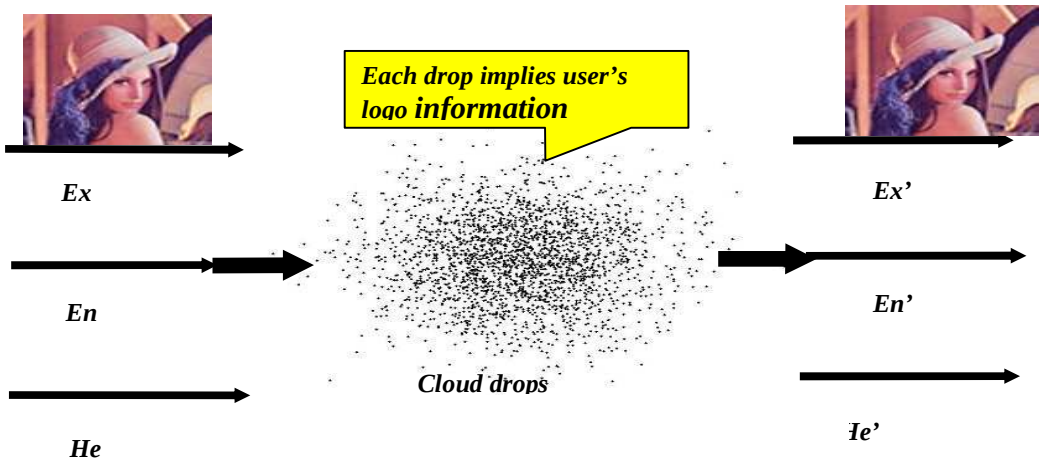


Figure 7. 20 The concept of data coloring or data watermarking by adding unique log cloud drops into a data file.

As a matter of fact, we suggest to merge these two concepts for data protection in the cloud. Cloud security is indeed a community property by combining cloud storage and watermarking. The cloud watermark generator employs three Type-2 fuzzy parameters: *expected value* (Ex), *entropy* (En), and *hyper-entropy* (He). As shown in Fig.7.21, these are used to generate special color for each data object.

Data coloring means labeling each data object by a unique color. Differently colored data objects are thus distinguishable. The user identification is also colored and matched with the data colors to initiate different trust management events. Cloud storage provides a process for the generation, embedding, and extraction of the watermarks.

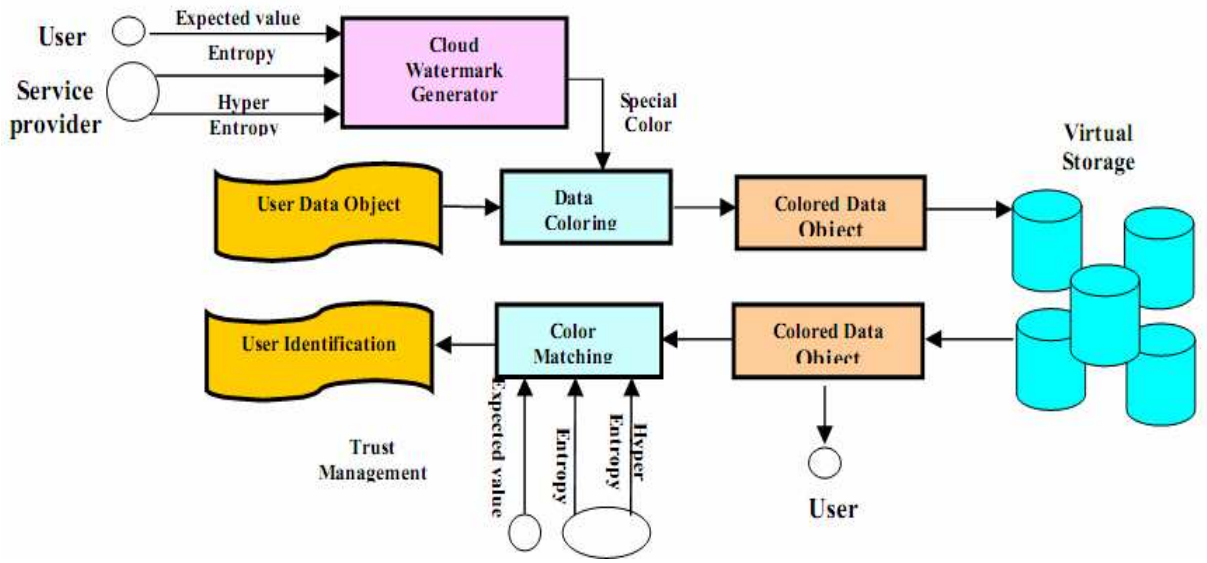


Figure 7.21. Data coloring with cloud watermarking for trust management at various security clearance levels in the datacenters (Courtesy of Hwang and Li, "Security, Privacy, and Data protection for Trusted Cloud Computing", *IEEE Internet Computing,* Sept. 2010 [30])

Data Lock-in Problem and Proactive Solutions : Cloud computing moves both the computation and data to the server clusters maintained by cloud service providers. Once the data is moved into the cloud, users cannot easily extract their data and programs from cloud servers to run on another platform. This leads to a data lock-in problem. This has hindered the use of cloud computing. The data lock-in is attributed to two causes. (1) *Lack of interoperability:* Each cloud vendor has their proprietary API that limits users to extract data once submitted. (2) Lack of a*pplication compatibility:* Most computing clouds expect user to write new applications from scratch, when they switch the cloud platforms.

One possible solution to data lock-in is the use of standardized cloud APIs. This requires building standardized virtual platforms that adhere to *Open Virtual Format* (OVF) – a platform-independent, efficient, extensible and open format for virtual machines. This will enable efficient, security software distribution, facilitating the mobility of virtual machines. Using OVF, one can move the data from one application to another. This will enhance the QoS and thus enable cross-cloud applications, allowing workload migration among datacenters to user-specific storage. By deploying applications without the need of rewriting per cloud, we can access and inter-mix the applications across different cloud services.

7.4 Cloud Platforms and Services Models

This section characterizes the various cloud service models and their extensions. The cloud service trends are reviewed. Then we prepare the readers with an overview of various network threats to cloud platforms and the services they provide. The defense against these threats and trust management will be treated in Section 6.6 .

7.4.1 Major Cloud Providers and Service Offerings

Cloud services are demanded by computing and IT administrators, software vendors, and end users. Figure 7.22 introduces five levels of cloud players in industry. At the top level, individual users and organizational users demand very different services. The application providers at SaaS level serve mainly the individual users. Most business organizations are serviced by IaaS and PaaS providers. The infrastructure services (IaaS) provide compute, storage and communication resources to both applications and organizational users. The cloud environment is defined by the PaaS or platform providers. The cloud platform is built on top of hardware and software infrastructure. Hence, the hardware and software providers feed the platform builders. Note that the platform providers support both infrastructure services and organization users directly.

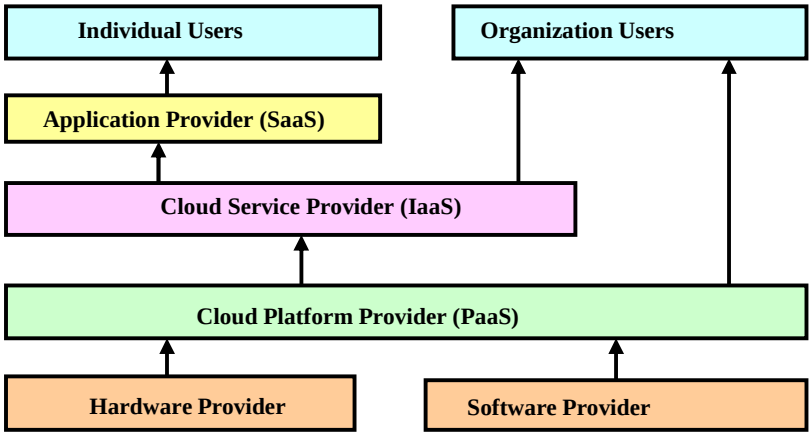


Figure 7.22 Roles of various cloud players or providers in cloud computing industry

Cloud computing services rely on new advances in machine virtualization, service-oriented architecture, grid infrastructure management, power efficiency, etc. Consumers purchase such services in the form of IaaS, PaaS, or SaaS as described above. There are also many cloud entrepreneurs selling value-added utility services to massive users. The cloud industry leverages the growing demand by many enterprises and business users to outsource their computing and storage jobs to professional providers. The provider service charges are often much less than user replacement of their obsolete servers frequently. In the future, a few cloud providers may satisfy massive number of users more cost-effectively. In Table 7.4, we summarizes the profiles of 5 major cloud providers by 2010 standard.

The programming and service offerings will be exemplified in details in Chapter 7. Amazon pioneered the IaaS business in supporting e-commerce and cloud applications by millions of customers simultaneously. The elasticity in Amazon cloud comes from the flexibility provided by the hardware and software services. The EC2 provides an environment for running virtual servers on demand. The S3 provides unlimited online storage space. Both EC2 and S3 are supported in *Amazon Web Services* (AWS) platform. Microsoft offer the Windows Azure platform for cloud applications. They have also supported the .NET service, dynamic CRM, hotmail, and SQL applications. Salsforce.com offers extensive SaaS applications for on-line CRM applications using their own Force.com platforms.

All IaaS, PaaS, and SaaS models allow the user to access the services over the Internet, relying entirely on the infrastructures of the cloud service providers. These models are offered based on various *service-level agreements* (SLAs) between the providers and users. SLAs are more common in network services as they account to the "QoS" characteristics of the network services. For cloud computing services, it is difficult to find a reasonable precedent for negotiating an SLA. In a broader sense, the SLAs for cloud computing addresses the service availability, data integrity, privacy and security protections. The blank

Table 7.4 : Major Cloud Providers, Platforms, and Their Service Offerings in 2010 [30]

Model/ Features	IBM	Amazon	Google	Microsoft	Salesforce.com
PaaS	BlueCloud, WCA, RC2,		App Engine (GAE)	Windows Azure	Force.com
IaaS	Ensembles	AWS			
SaaS	Lotus Live		Gmail, Docs	.NET service, Dynamic CRM,	Online CRM, Gifttag
Service Offerings	SOA, B2, TSAM, RAD, Web 2.0	EC2, S3, SQS, SimpleDB	GFS, Chubby, BigTable, MapReduce	Live, SQL Hotmail	Apex, visual force, Record-security
Security Features	WebSphere2 and PowerVM tuned for protection	PKI and VPN for security, EBS to recover from failure	Chubby locks for security enforcemnt	Replicated Data, rule-based access control	Adm./Record security, Use Metadata API

Note: WCA: Websphere CloudBurst Appliance, RC2: Research Compute Cloud, RAD: Rational Application Developer, SOA: Service Oriented Architecture, TSAM: Tivoli Service Automation Manager, EC2: Elastic Compute Cloud. S3: Simple Storage Service, SQS: Simple Queue Service, GAE: Google AppEngine, AWS: Amazon Web Services, SQL: Structured Query Language, EBS: Elastic Block Store, CRM: Consumer Relationship Management.

7.4.2 Cloud Service Models and Extensions

Figure 7.23 shows 6 layers of cloud services, ranging from hardware, network, and co-location, to infrastructure, platform, and software applications. We have already introduced the top 3 service layers as SaaS, PaaS, and IaaS, respectively. The cloud platform provides the PaaS, which sits on top of the IaaS infrastructure. On the top layer offers software application services (SaaS). These must be implemented on the cloud platforms provided. Although, the three basic models are dissimilar in usage as shown in Table 6.8, they are built one on top of another, clearly. The implication is that one cannot launch SaaS applications with a cloud platform. The cloud platform cannot be built if compute and storage infrastructures are not there.

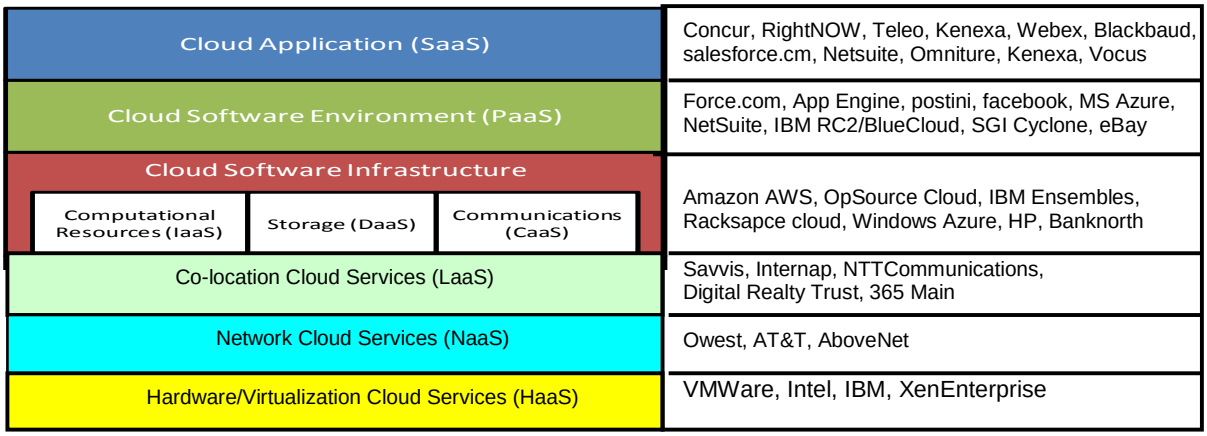


Figure 7.23 A stack of six layers of cloud services and their major providers, ranging from hardware, middleware and network services to location, infrastructure, platform, and software services. (Courtesy: partially from T. Chou, *Introduction To Cloud Computing,* 2010 [12])

The 3 bottom layers are more related to physical requirements. The bottom layer provides *Hardware as a Service* (HaaS). The next layer is to interconnect all the hardware components together, simply called *Network as a Service* (NaaS). *Virtual local-are network* (VLAN) falls within the scope of NaaS. The next higher layer offers *Loaction as a Service* (LaaS), which provide a co-location service to house, power, and secure all the physical hardware and network resources. Some authors call this layer to provide *Security as a Service* ("SaaS"). The cloud infrastructure layer can be further subdivided as *Data as a Service* (DaaS), and *Communication as a Service* (CaaS) in addition to compute and storage in IaaS.

We will examine the commercial trends in cloud services in subsequent sections. We cover mainlt the top 3 layers with some success stories of cloud computing. As shown in Table 7.5, we divide cloud players into three classes: (i) cloud service providers and IT administrators, (ii) software developers or vendors and (iii) end users or business users. These cloud players varies in their roles under the IaaS, PaaS, and SaaS models. The table entries distinguish the three cloud models as viewed by different players. From software vendors' perspective, the application performance on a given cloud platform is most important. From the perspective of cloud providers, mainly the cloud infrastructure performance is important. From end-user's perspective, they concern mainly the quality of services rather the infrastructure performance issues.

Table 7.5 Model Differences from The Perspectives of providers, Vendors, and Users

Cloud Users	IaaS	PaaS	SaaS
IT Administrators/ Cloud Providers	Monitor SLAs	Monitor SLAs and enable Service Platforms	Monitor SLAs and deploy software
Software Developers (Vendors)	To deploy and store data	Enabling Platforms via configurator and APIs	Develop and Deploy Software
End Users or Business Users	To deploy and store data	To develop and test web software	Use Business Software

Runtime Support Services: Just similar as in a cluster environment, there are also some runtime supporting services in the cloud computing environment. Cluster monitoring is use to collect runtime status of the whole cluster. One of the most important facilities is the cluster job management system introduced in Chapter 3. The scheduler queues the tasks submitted to the whole cluster and assign the tasks to the processing nodes according to the node availability. The distributed scheduler for the cloud application has special characteristics that can sport the cloud applications such as scheduling the programs written in MapReduce style. The runtime supporting services are fundamental infrastructure to keep the cloud cluster working properly.

This refers to browser-initiated application software over thousands of cloud customers. SaaS model provides the software applications as a service, rather than letting the users to purchase the software packages. As a result, on the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are rather low, compared with conventional hosting of user applications. The customer data is stored in the cloud that is either vendor proprietary or a publically hosted cloud supporting the PaaS and IaaS.

Software Stack for cloud Computing: Despite the various types of nodes in the cloud computing cluster, the overall software stacks are built from the scratch to meet the rigorous goals. Developers have to consider on how to design the system to meet the critical requirement such as high throughput, high availability, and fault tolerant. Even the operating system might be modified to meet the special requirement of cloud data processing. Based on the observation of some typical cloud computing instances such as Google, Microsoft, Yahoo, etc, the overall software stack structure of cloud computing software can be viewed as layers. Each layer has its own purpose and provides the interface for the upper layer just as the traditional software stack. However, the lower layer is not completely transparent to the upper layer. The software stack for cloud computing services is shown in Fig.7.24.

. The platform for running cloud computing services can be either physical servers or virtual servers. By using the virtual machines, the platform can be flexible i.e. the running services are now not bound to specific hardware platforms. This brings the flexibility to the cloud computing platforms. The software layer on top of the platform is the platform is the layer for storing massive data. This layer acts similar as the file system in the traditional single machine. Other layers running on top of the file systems are the layers for executing the cloud computing applications. They include the layer of database storage system, programming for the large scale cluster and data query language support. Next layer are the components in the software stack.

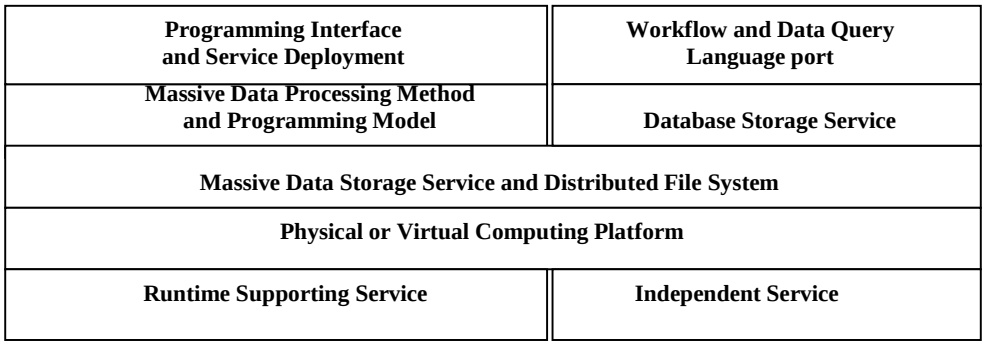


Figure 7.24 Software stack for cloud computing services

7.4,3 Trends in Cloud Service Applications

Currently, major cloud applications include upgraded web services, distributed data storage, raw supercomputing, and access to specialized Grid or P2P services. Web services have been upgraded to Web 2.0. The Web 2.0 services emphasizes interactive and personalized services. User can talk to the web applications through web browsers more effectively and often get their response quickly. Thus the web is no longer an one-way download services. The Web 2.0 services include Facebook, Youtube, Flickr, etc. Cloud computing applies interactive web interface or GUI in these applications.

In cloud computing, performance is not the only objective function, but one of the attributes that will it to attract users. Addressing the cloud performance in an end-to-end fashion includes the performance of a cloud application, run over a cloud infrastructure, delivered via the Internet or Web2.0 that brings about the network performance aspects. In evaluating the performance, the point of measurement plays an important role. As the cloud services are delivered over the Internet, from the end user's perspective the quality of service could be defined by the combined performance in networked platform, application results, the infrastructure utilization, etc.

Cloud Service Stack : Cloud services are introduced in Fig.6.25 in 5 layers. The top layer is for SaaS applications as further subdivided into 5 application areas in Table 7.6, mostly for business applications. For an example, *consumer relationship management* (CRM) is heavily practiced in business promotion, direct sales, and marketing services. CRM offered the first SaaS on the cloud successfully. The approach is to widen the market coverage by investigating customer behaviors and revealing opportunities by statistical analysis. SaaS tools also apply to distributed collaboration, financial and human resource management. These cloud services are growing rapidly in recent years.

PaaS is provided by Google, Salesforce, facvebook, etc as detailed in Chapters 8 and 9. IaaS is provided by Amazon, Windows Asure, RackRack, etc. The co-location services involve multiple cloud providers to work together such as supporting supply chains in manufacturing. The network cloud services provide communications such as those by AT&T, Qwest, AboveNet, etc. Details cab e found in Clou's introductory book on business clouds [12]. The vertical cloud services in Table 7.6 refer to a sequence of cloud services that are mutually supportive. Often cloud meshup is practiced in vertical cloud applications.

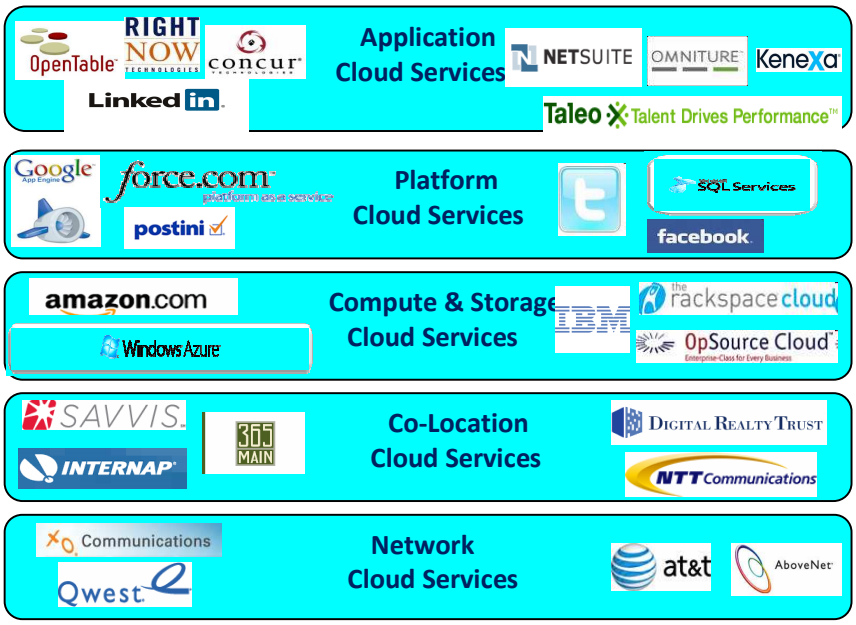


Figure 7.25 Cloud srvice stack ranging from application, platform, infrastructure to co-location and network services in 5 layers (Courtesy of T. Clou, *Introduction To Computing Computing: Business and Technology*, Active Book Press, 2010 [12])

Table 7.6 Application Cloud Services Tasks

Application Services	Typical Application Service Tasks
Collaboration	Email Management, Mailroom Management, Panel Management, Survey, Web Collaboration, Web Conferencing, Training
Financials	Accounting, Accounts Payable, Accounts Receivable, Billing and Invoicing, Budgeting, Tax Planning. Business Performance Management, Compliance, Expense Report, Financial Reporting, Fixed Asset Management, Purchasing, Enterprise Resource Management, Project Management
Sales	CRM, Sales Performance Management, Contact Management, Lead Management, Proposal Management, Quoting, Sales Force Automation, Call Center, Customer Service Satisfaction, Help Desk, Predictive Dialer
Human Resources	Talent Management, Payroll, Performance Appraisal, Applicant Tracking, Recruiting, Benefits Administration, Human Resource, Workforce Management
Marketing	Advertisement, Brand Management, Business Graphics, Campaign, Competitive Intelligence, Email Marketing, Marketing Automation, Public Relations, Search Marketing Automation
Vertical applications	Reservation Management, Transportation & Logistics, Government, Banking & Finance, Education & Teaching, Energy, Entertainment, Healthcare & Medicine, Hospitality & Travel, Legal & Law Enforcement, Manufacturing, Media, Nonprofit, Real Estate, Telecommunications

7.5 Resource Management and Design Challenges

In this section, we study the method for resource provisioning in a cloud. Then we examine cloud infrastructures and study some design challenges and explore plausible solutions to these problems.

7.5.1 Resource Provisioning and Platform Deployment

The emergence of computing clouds suggests fundamental changes in software and hardware architecture. Cloud architecture emphasizes more on the number of processor cores or VM instances. Parallelism is exploited at the cluster node level. In this section, we first study techniques to provision computer resources or VMs. Then we discuss the storage allocation schemes and techniques to interconnect distributed computing infrastructures by harnessing the VMs dynamically.

Provisioning of Compute Resources (VMs) : Providers supply cloud services by signing *service-level agreements* (SLAs) with the end-users. The SLAs must commit sufficient resources such as CPU, memory, and bandwidth that the use can use for a preset time period. Under-provisioning of resources would lead to broken SLAs and penalties. Over-provisioning of resources would lead to resource underutilization and, consequently, a decrease in revenue for the provider. Deploying an autonomous system to efficiently provision resources to users is indeed a very challenging problem. The difficulty comes from the unpredictability of consumer demand, software and hardware failures, heterogeneity of services, power management, and conflicts in signed SLAs between consumers and service providers.

Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures. The resource provisioning schemes demands also fast discovery of services and data in cloud computing infrastructures. In a virtualized cluster of servers, this demands efficient installation of the VMs, live VM migration, and fast recovery from failures. To deploy VMs, users treat them as physical hosts with customized operating system for specific applications. For example, Amazon's EC2 (*Elastic Computing Cloud*) uses the XEN as the *virtual machine monitor* (VMM). The same VMM is used in IBM's BlueCloud.

In EC2 platform, some predefined VM templates are also provided. Users can choose different kinds of VMs from the templates. IBM's BlueCloud does not provide any VM templates. In general, any type of VM can run on top of XEN. Microsoft also applies virtualization in its Azure cloud platform. The provider should offer resource-economic services. Power-efficient schemes for caching, query processing, and thermal management are mandatory due to increasing energy waste by heat dissipation from datacenters. Public or private clouds promise to streamline the on-demand provisioning of software, hardware, and data as a service, achieving the economies of scale in IT deployment and operation.

Dynamic Resource Deployment: The cloud uses virtual machines as building blocks to create an execution environment across multiple resource sites. We use the InterGrid-managed infrastructure developed by the Melbourne University group [19] to explain how dynamic resource deployment can be implemented to achieve scalability in performance. The InterGrid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources.

Peering arrangements established between gateways enables the allocation of resources from multiple grids to establish the execution environment. Figure 7.26 illustrates a scenario in which an IGG allocates resources from a local cluster to deploy applications in 3 steps: (1) requesting the VMs, (2) enactment of the leases, and (3) delpoyment of the VMs as requested. Under peak demand, this IGG interacts with another that can allocate resources from a cloud computing provider.

A grid has predefined peering arrangements with other grids, which the *inter-grid gateway* (IGG) manages. Through multiple IGGs, the system coordinates the use of InterGrid resources. An IGG is aware of the peering terms with other grids, selects suitable grids that can provide the required resources, and replies to requests from other IGGs. Request redirection policies determine which peering grid InterGrid selects to process a request and a price for which that grid will perform it. An IGG can also allocate

resources from a cloud provider. The cloud system creates a virtual environment to help users deploy their applications. These applications use the distributed grid resources.

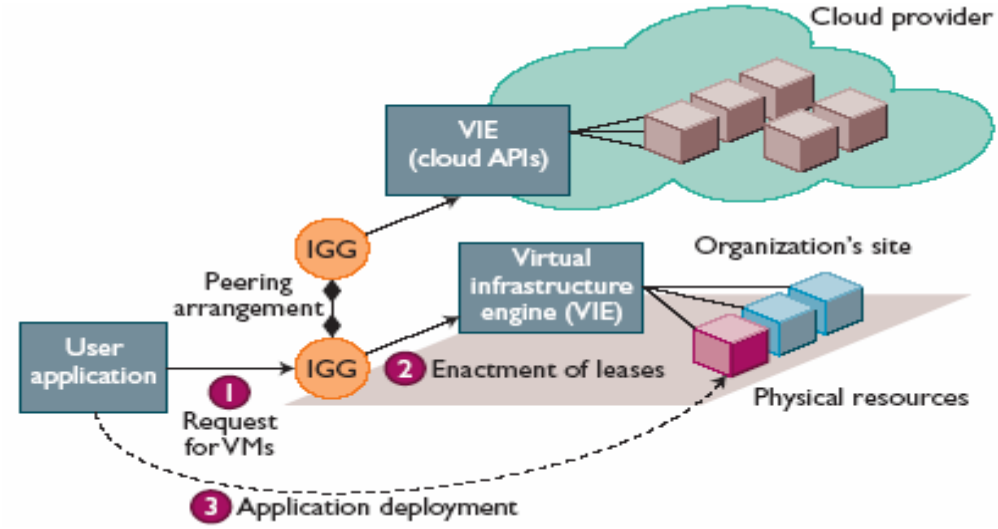


Figure 7.26 Cloud resource deployment using an InterGrid gateway (IGG) to allocate the virtual machines (VMs) from a local cluster to interact with the IGG of a public cloud provider.(Courtesy of Constanzo, Assuncao, and Buyya, *IEEE Internet Computing*, Sept. 2009 [17])

The InterGrid allocates and provides as a *distributed virtual environment* (DVE). This is a virtual cluster of VMs that runs isolated from other virtual clusters. A component, called the *DVE manager*, performs resource allocation and management on behaff of specific user applications. The core component of the IGG is a scheduler to implement the provisioning policies and peering with other gateways. The communication component provides an asynchronous message-passing mechanism. Received messages are handled in parallel by a thread pool. For more details, readers are referred to the cited article [19].

Provisioning of Storage Resources : The data storage layer is built on top of the physical or virtual servers. As the cloud computing applications often provide service to users, it is unavoidable that the data are stored in the clusters of cloud provider. The service can be accessed through out of the world which means the scale of the data will be quite large. One example is the email systems. A typical large email system might have millions of users and each user can have thousands of emails and consumes multiple GB of disk space. Another example is the web searching application. In storage technologies, hard disk drives may be augmented with solid-state drives in the future. This will provide reliable and high-performance data storage. The biggest barriers to adopting flash memories in datacenters have been price, capacity, and, to some extent, the lack of sophisticated query-processing techniques. However, this is about to change as the I/O bandwidth of solid-state drives become too impressive to ignore.

Distributed file system is a very important form for storing large scale data. However, there are still some other forms of data storage. Some data do not need the namespace of tree structure file system. Database is the storage not using files. In cloud computing, there are some other forms of data storage such as (Key,Value) pairs. Beside the common file system interface, storage can also be expressed as some other accessing methods. Amazon has developed a service called S3 (Simple Storage Service) which uses SOAP (*Simple Object Access Protocol*) for accessing the objects stored in the cloud. Table 7.7 summarized three cloud storage services provided by Google, Hadoop, and Amazon.

Table 7.7 Storage Services in Various Cloud Computing Systems

Storage System	Function Features
GFS: Google File System	GFS features with very large sustainable reading and writing bandwidth, mostly continuous accessing instead of random accessing. The programming interface is similar to that of POSIX file system accessing interface.
HDFS: Hadoop Distributed File System	The open source clone of GFS. Written in Java. The programming interfaces are similar as POSIX but not identical.
Amazon S3 and EBS	S3 used for retrieving and storing data from/to remote servers. EBS is built on top of S3 for using virtual disks in running EC2 instances.

The storage system might have the whole Internet data. Thus, many cloud computing companies have developed large scale data storage system to keep copies of the data. For example, Google has developed GFS (*Google File System*) to store the web data and some other data like geographical data for Google Earth. Similar system has been built in the open source community like *Hadoop Distributed File System* (HDFS) for apache. Hadoop is the open source implementation of Google's cloud computing infrastructure such as Google file system and MapReduce framework. Similar systems include Microsoft file system Cosmos for the cloud.

Despite the storage service or distributed file system that can be accessed directly, similar as in traditional database, cloud computing does provide some forms of structure or semi-structure database processing capability. For example, applications might want to process the information contained in a web page. Web pages are the example of semi-structural data with the format of HTML. If some forms of database capability can be used, application developers will construct their application logic more easily. Another reason for building the database like service in cloud computing is that it will be quite convenient for traditional application developers to code for the cloud platform. Database is quite common as the underline storage device for many applications.

Thus, such developers can think in a familiar way as they've done for traditional software development. Hence, in cloud computing, it is necessary to build database like large scale system based on data storage or distributed file system. The scale of such database might be quite large for processing huge amount of data. The main purpose is to store the data in structural or semi-structural ways that the application developer can use easily and build their applications rapidly. As for the strong consistent issue, traditional database will meet the performance bottleneck while the system is expanded to large scale. However, some real applications do not need such strong consistency. The scale of such database can be quite large. Typical cloud database include BigTable from Google, SimpleDB from Amazon and SQL service from Microsoft Azure.

Resource Provisioning Strategies : Three cases of cloud resource provisioning without elasticity are shown in Fig.7.27. In case (a), over provisioning with the peak load cases heavy resource waste (shaded area). In case (b), under provisioning (along the capacity line) of resources results in losses by both user and provider in that paid demand by users (shaded area above the capacity) is not served and wasted resources still exists for those undemanded areas below the provisioned capacity. In case (c), we see a fixed provisioning of a constant capacity for a declining user demand of resources could result in even large resource waste. User may give up the service by canceling the demand meaning reduced revenue by the provider. Both users and provider may be losers in resource provisioning without elasticity.

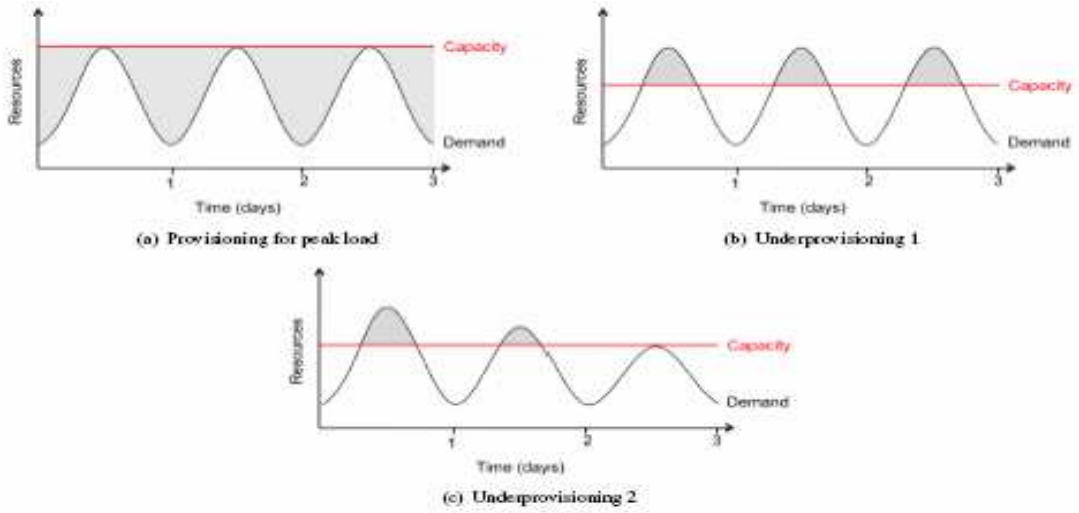


Figure 7.27 Three cases of cloud resource provisioning without elasticity. Heavy waste in case (a) by over provisioning resources to satisfy peak load. Under provisioning results in user's paid service disabled in case (b). Static provisioning results in heavy waste in case (c), if the resource demand declines rapidly in time. (Courtesy of Berkeley Report by Armbrust, et al, Feb.2009 [4].)

7.5.2 Virtual Machine Creation and Management

In this section, we consider several issues for cloud infrastructure management. First, we consider the resource management of independent service jobs. Then we consider how to execute third-party cloud applications. We use the cloud loading experiments by Melbourne research group on the French Grid'5000 system to illustrate the VM creation and management. This case-study example will reveal major VM management issues and suggest some plausible solutions of workload-balanced execution. Figure 7.28 shows the Interactions among VM managers for cloud creation and management. The managers provides a public API for users to submit and control the VMs.

Independent Service Management: Independent services request facilities to execute many unrelated tasks. Commonly, the APIs provided are some web services that the developer can use conveniently. In Amazon cloud computing infrastructure, SQS (S*imple Queue Service*) is constructed for providing the reliable communication service between different providers. Even endpoint does not run while another entity has posted a message in the SQS. By using independent service providers, the cloud applications can run different services at the same time. Some other services are used for providing data other than the compute or storage services.

Running Third Party Applications: Cloud platforms have to provide the support for building applications that are constructed by third party application providers or programmers. As current web applications are often provided by using Web 2.0 form (interactive applications with Ajax), the programming interfaces are different from the traditional programming interfaces such as functions in runtime libraries. The APIs are often in the form of services. Web service application engines are often used by the programmers for building applications. The web browsers are the user interface for end users.

In addition to the gateway applications, cloud computing platform gives extra capabilities of accessing the backend services or underlying data. As examples, Google AppEngine and Microsoft Azure apply their own cloud APIs to get special cloud services. WebSphere application engine is deployed in IBM for the BlueCloud. It can be used to develop any kind of web applications written in Java. In EC2, users can use any kind of application engine that can run in VM instances.

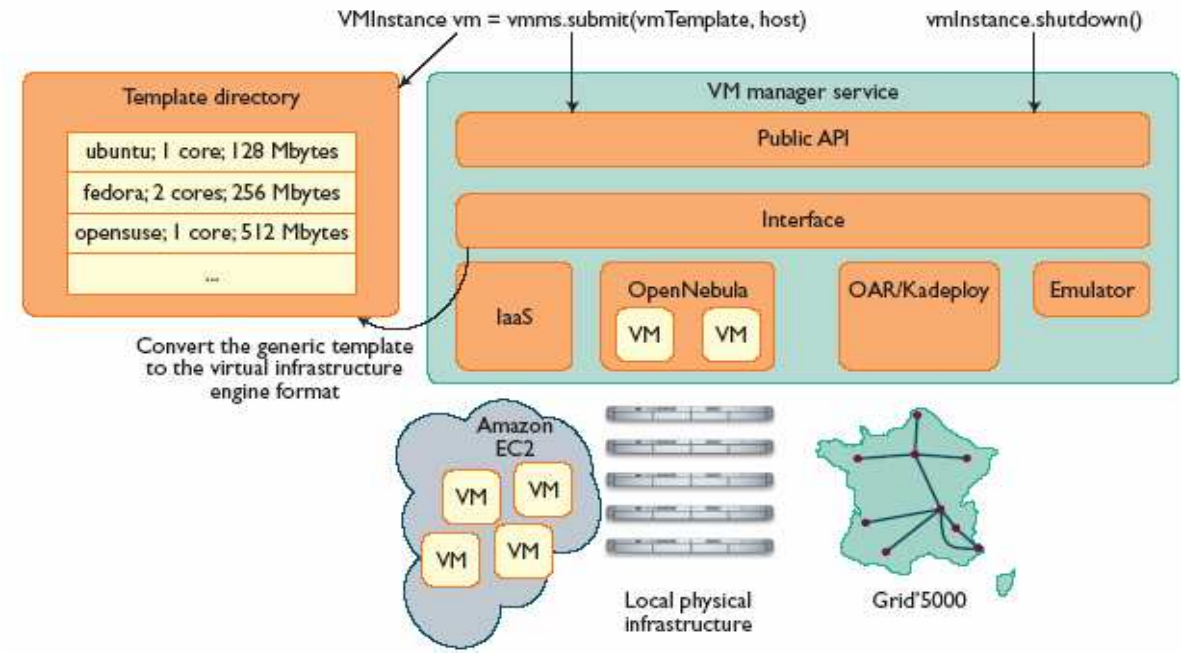


Figure 7.28 Interactions among VM managers for cloud creation and management. The managers provides a public API for users to submit and control the VMs. (Courtesy of Constanzo, Assuncao, and Buyya, *IEEE Internet Computing*, Sept. 2009 [17])

Virtual Machine Manager : The VM manger is the link between gateway and resources. The gateway doesn't share physical resources directly but relies on virtualization technology for abstracting them. Hence, the actual resources it uses are VMs. The manager relies on a VIE to manage VMs on a set of physical resources. VM manager implementation is generic so that it can connect with different VIEs. Typically, VIEs can create and stop VMs on a physical cluster. The Melbourne group has developed managers for OpenNebula, Amazon EC2, and French Grid'5000. In what follows, the manager for using OpenNebula (www.opennebula.org) is assumed as a VIE for deploying VMs on a local cluster.

OpenNebula recognizes. OpenNebula runs as a daemon service on a master node, so the VMM works as a remote user. Users submit VMs on physical machines using different kinds of hypervisors, such as Xen (www.xen.org), which enables running several operating systems on the same host concurrently. The VMM also manages VM deployment on grids and IaaS providers. The InterGrid supports Amazon EC2. The connector is a wrapper for the command-line tool Amazon provides. The VM manager for Grid'5000 is also a wrapper for its command-line tools. To deploy a VM, the manager needs to use its template.

Virtual Machine Templates : A *VM template* is analogous to a computer's configuration and contains a description for a VM with the following static information:

- the number of cores or processors to be assigned to the VM,
- the amount of memory the VM requires,
- the kernel used to boot the VM's operating system,
- the disk image containing the VM's file system, and
- the price per hour of using a VM.

The gateway administrator provides the VM template information when the infrastructure is set up. The administrator can update, add, and delete templates at any time. In addition, each gateway in the InterGrid network must agree on the templates to provide the same configuration on each site. To deploy an instance of a given VM, the VMM generates a descriptor from the template. This descriptor contains the same fields

as the template and additional information related to a specific VM instance. Typically the extra information include :

- the disk image that contains the VM's file system,
- the address of the physical machine hosting the VM,
- the VM's network configuration, and
- the required information for deployment on an IaaS provider.

Before starting an instance, the scheduler gives the network configuration and the host's address; it then allocates MAC and IP addresses for that instance. The template specifies the disk image field. To deploy several instances of the same VM template in parallel, each instance uses a temporary copy of the disk image. Hence, the descriptor contains the path to the copied disk image. The descriptor's fields are different for deploying a VM on an IaaS provider. Network information is not needed, because Amazon because EC2 automatically assigns a public IP to the instances. The IGG works with a repository of VM templates , called the *VM template directory*.

Distributed VM Management : Figure 6.24 shows the interactions between InterGrid's components. A distributed VM manager makes requests for VMs and querying their status. This manager requests VMs from the gateway on behalf of the user application. The manager obtains the list of requested VMs from the gateway. This list contains a tuple of public IP/ private IP for each VM with Secure Shell [SSH] tunnels. Users must specify which VM template they want to use and the number of VM instances needed, the deadline, the wall time, and the address for an alternative gateway.

The local gateway tries to obtain resources from the underlying VIEs. When this is impossible, the local gateway starts a negotiation with any remote gateways to fulfill the request. When a gateway schedules the VMs, it sends the VM access information to the requester gateway. Finally, the manager configures the VM, sets up SSH tunnels, and executes the tasks on the VM. Under the peering policy, each gateway's scheduler uses conservative backfilling to schedule requests. When the scheduler can't start a request immediately using local resources, then a redirection algorithm will be initiated..

Example 7.4: Benchmark Experiments on an InterGrid Testbed over The Grid'5000

The Melbourne group conducted two experiments to evaluate the InterGrid architecture. The first one evaluates the performance of allocation decisions by measuring how the IGG manages load via peering arrangements. The second considers its effectiveness in deploying a bag-of-tasks application. The experiment was conducted on the French experimental grid platform: Grid'5000. Grid'5000 comprises 4,792 processor cores on nine grid sites across France. Each gateway represents one Grid'5000 site shown Fig.7.29.

To prevent the gateways from interfering with real Grid'5000 users, emulated VM managers were implemented to instantiate fictitious VMs. The number of emulated hosts is limited by the core number at each site. A balanced workload was configured among the sites. The maximum VMs requested do not exceed cores in any site. Figure 7.30 shows the load characteristics under a 4-gateway scenario. The teal bars indicate each grid site's load. The magenta bars show the load when gateways redirect requests to one another. The green bars correspond to the amount of load each gateway accepts from other gateways. The brown bars represent the amount of load redirected. The results show that the loading policy applied can balance the load across the 9 sites. Rennes, a site with a heavy load, benefits from peering with other gateways as the gateway redirects a great share of its load to other sites. ■

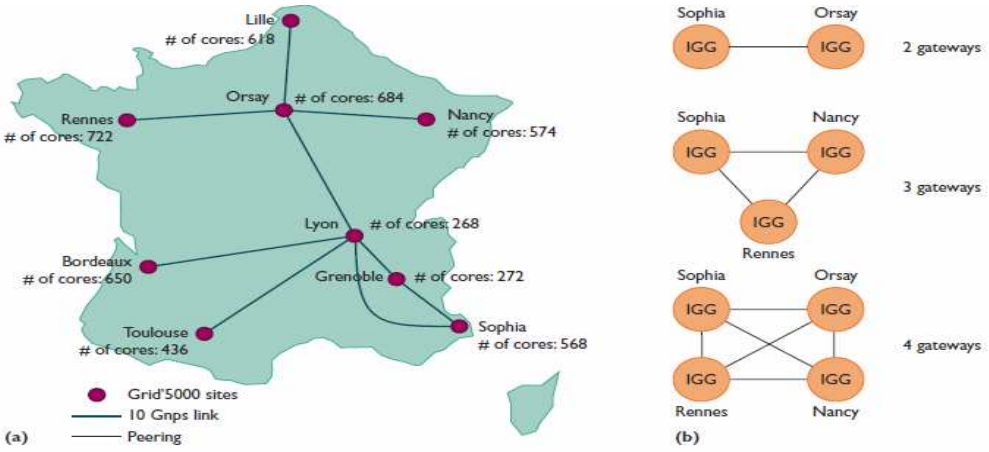


Figure 7.29 The InterGrid testbed over the French Grid'5000 located in 9 cities across France . (Courtesy of Constanzo, Assuncao, and Buyya, *IEEE Internet Computing*, Sept. 2009 [17])

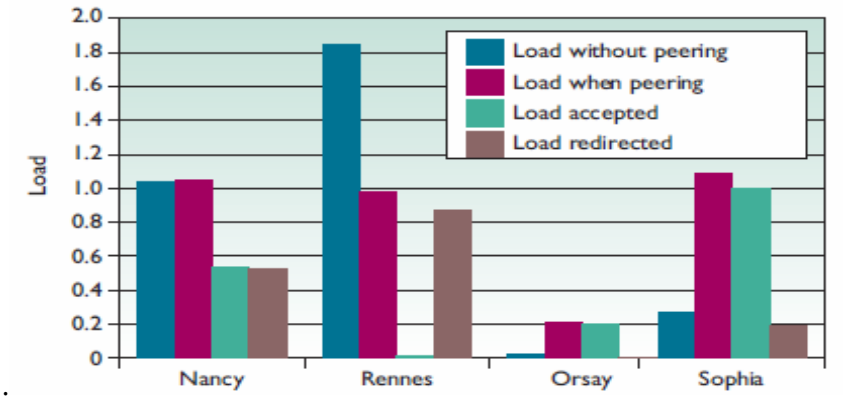


Figure 7.30 Cloud loading results at 4 gateways at resource sites in Grid'5000 system. The bars indicate the load with and witout peering and the effect of load migration. (Courtesy of Constanzo, Assuncao, and Buyya, *IEEE Internet Computing*, Sept. 2009 [17])

7.5.3 Cloud Architectural Design Challenges

We identify below six open challenges in cloud architecture development. Armbrust, et al [4] have advocated that some of these are obstacles to cloud development as well as offering new opportunities for cloud system designers. Plausible approaches to meet these challenges are discussed below. Some solutions are already implemented in commercial cloud platforms to be presented in Chapter 8. Other new ideas are still under research and development by the cloud computing communities.

***Challenge (1) -- Service Availability and Data Lock-in Problem*:** The management of a cloud service by a single company is often the source of single-point of failures. To achieve high availability, one can consider using multiple cloud providers. Even if the company has multiple datacenters located in different geographic regions, it may have common software infrastructure and accounting systems. Therefore, using multiple cloud providers may provide more protection from failures. Another availability obstacle is *distributed denial of service* (DDoS) attacks. Criminals threaten to cut off the incomes of SaaS providers by making their service unavailable. Some utility computing services offer SaaS providers the opportunity to defend against DdoS attacks by using quick scale-up.

Software stacks have improved interoperability among different cloud platforms, but the APIs itself

are still proprietary. Thus, customers cannot easily extract their data and programs from one site to run on another. The obvious solution is to standardize the APIs so that an SaaS developer could deploy services and data across multiple cloud providers. This will rescue the loss of all data due to the failure of a single company. In addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the same software infrastructure can be used in both public and private clouds. Such an option could enable "Surge Computing," in which the public cloud is used to capture the extra tasks that cannot be easily run in the datacenter of a private cloud.

Challenge (2) Data Privacy and Security Concerns **:** Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks. Many of the obstacles can be overcome immediately with well understood technologies such as encrypted storage, virtual local area networks, and network middleboxes (e.g. firewalls, packet filters). For example, encrypting data before placing it in a cloud. Many nations have laws requiring the SaaS providers to keep customer data and copyrighted material within the national boundaries.

Traditional network attacks include buffer overflows, DoS attacks, DoS), spyware, malwares, rootkits, Trojans, worms. In a cloud environment, newer attacks may be resulted from hypervisor malware, guest hopping and hijacking, or VMRootKits. Another one is the man-in-the-middle attack for VM Migrations. In general, passive attacks steal sensitive data or passwords. The active attacks may manipulate the kernel data structures which will cause major damages to cloud servers. All of these security and privacy problems on clouds will be studied in Section 6.5.

Challenge (3) —Unpredictable Performance and Bottlenecks**:** Multiple VMs can share CPUs and main memory in cloud computing, but that I/O sharing is more problematic. For example, to run 75 EC2 instances with the STREAM benchmark [??], the mean bandwidth is 1,355 MB/s. However, for each of the 75 EC2 instances to write 1 GB files to local disk, the mean disk write bandwidth is only 55 MB/s. This demonstrates the problem of I/O interference between VMs. One opportunity is to improve the I/O architectures and operating systems to efficiently virtualize interrupts and I/O channels.

Internet applications continue to become more data-intensive. If we assume applications to be "pulled apart" across the boundaries of clouds, this may complicate data placement and transport. Cloud users and providers have to think about the implications of placement and traffic at every level of the system, if they want to minimize the cost. This kind of reasoning can be seen in Amazon's development of their new Cloudfront service. Therefore data transfer bottleneck must be removed. The bottleneck links must be widened. The weak servers should be removed. Performance issue will be studied in Chapter 8.

Challenge (4) -- Distributed Storage and Wide-Spread Software Bugs**:** The database is always growing in cloud applications. The opportunity is to create a storage system that would not only meet the growth but also combine them with the cloud advantages of scaling arbitrarily up and down on-demand. This demands the design of efficient distributed *storage-area network* (SAN). The datacenters must meet programmer's expectations in scalability, data durability, and high availability. Data consistence checking in SAN-connected datacenters is a major challenge in cloud computing.

Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production datacenters. No datacenter will provide such a convenience. One opportunity may be the reliance on using VMs in cloud computing. The level of virtualization may make it possible to capture valuable information in ways that are implausible without using VMs. Debugging over simulators is also another approach to attack the problem, if the simulator is well designed.

Challenge (5) — Cloud Scalability, Interoperability, and Standardization **:** The pay-as-you-go model applies to storage and network bandwidth, both are counted by the bytes used. Computation is different depending on the virtualization level. Google App Engine automatically scales in response to load increases

and decreases. Users are charged by the cycles used. The AWS charges by the hour for the number of VM instances used, even if the machine is idle. The opportunity is to scale quickly up and down in response to load variation, in order to save money, but without violating the SLAs.

Open virtualization format (OVF) describes an open, secure, portable, efficient and extensible format for the packaging and distribution of virtual machines. It also defines a format for distributing software to be deployed in VMs. This VM format does not rely on the use of a specific host platform, virtualization platform, or guest operating system. The approach is to addresses virtual platform agnostic packaging with certification and integrity of packaged software. The package supports virtual Appliances to span more than one VM's.

The OVF also defines transport mechanism for VM templates. It an apply to different virtualization platforms with different levels of virtualization. Towards cloud standardization, we suggest to enable virtual appliances to run on any virtual platforms. We need also to enable VM's to run on heterogeneous hardware platform hypervisors. This require to enable hypervisor agnostic virtual machines. We need to realize cross platform live migration between x86 Intel and AMD and support legacy hardware for load balancing. All these issue are wide open for further _esearch.

Challenge (6)—Software Licensing and Reputation Sharing: Many cloud computing providers originally relied on open-source software, because the licensing model for commercial software is not ideal for utility computing. The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing. One can consider using both pay-for-use and bulk use licensing schemes to wide the business coverage.

Reputations do not virtualize well. One customer's bad behavior can affect the reputation of the entire cloud. For instance, blacklisting of EC2 IP addresses by spam-prevention services may limit the VM installation smoothly. An opportunity would be to create reputation-guarding services similar to the "trusted email" services currently offered (for a fee) to services hosted on smaller ISP's. Another legal issue is the question of transfer of legal liability. Cloud providers want legal liability to remain with the customer and vice versa. This problem must be solved at the SLA level. We will study reputation systems for protecting datacenters in the next section.

7.6 Cloud Security and Trust Management

Lacking trust between service providers and cloud users has hindered the universal acceptance of cloud computing as a service on demand. In the past, trust models have been developed to protect mainly e-commerce and on-line shopping provided by eBay and Amazon. For web and cloud services, trust and security become even more demanding, because leaving user applications completely to the cloud providers has faced strong resistance by most PC and server users. Cloud platforms become worrisome to some users for lack of privacy protection, security assurance, and copyright protection. Trust is a social problem, not a pure technical issue. However, the social problem can be solved with a technical approaches.

By common sense, technology can enhance trust, justice, reputation, credit and assurance in Internet applications. As a virtual environment, cloud poses new security threats that are more difficulty to contain than the traditional client and server configurations. To solve these trust problems, we present a new data-protection model based on data watermarking. Virtual resources and datacenters are facing many operational uncertainties. How to manage the uncertainties effectively is the most difficult challenge in trusted cloud computing. One can extend from the trust models previously proposed for *peer-to-peer* (P2P) networks and for grid systems. The ultimate goal is to secure the cloud computing environments.

7.6.1 Cloud Security Defense Strategies

We desire a healthy cloud ecosystem that is free from abuses, violence, cheating, hacking, viruses, rumors, pornography, spasm, privacy and copyright violations. We assess the security demands of three cloud service models: IaaS, PaaS, and SaaS that have used in cloud practices. These models are based on various *service level agreements* (SLAs) between providers and users.

Basic Cloud Security: Three basic cloud security enforcements are expected: (1) Facility security in datacenters demand on-side security all year round. Biometric readers, CCTV (close-circuit TV), motion detection, and man traps are often deployed. (2) Network security demands fault-tolerant external firewalls, intrusion detection systems, and third-party vulnerability assessment, and (3) Platform security demands SSL and data decryption, strict password policies, system trust certification, etc.

Servers in the cloud can be physical machines or virtual machines. User interfaces are applied to request services. The provisioning tool carves out the systems from the cloud to satisfy the requested service. A security-aware cloud architecture demands security enforcement. Malware-based attacks like worms, viruses and DdoS exploit the system vulnerabilities. These attacks compromise the system functionalities or provide the intruders an unauthorized access to critical information. Thus, security defense is needed to protect all cluster servers and datacenters. . Figure 7.31 shows s the mapping of cloud models to various security measures needed at cloud operational levels.

Listed below are some cloud components that demand special security protection

- Protection of servers from malicious software attacks like worms, viruses, and malwares.
- Protection of hypervisors or VM monitors from software based attacks and vulnerabilities.
- Protection of VMs and monitors from service disruption and denial of service attacks.
- Protection of data and information from theft, corruption and natural disasters.
- Providing authenticated and authorized access to critical data and services

Security Challenges in Virtual Machines: Traditional network attacks include buffer overflows, DoS attacks, DoS), spyware, malwares, rootkits, Trojans, worms. In a cloud environment, newer attacks may be resulted from hypervisor malware, guest hopping and hijacking, or VMRootKits. Another one is the man-in-the-middle attack for VM Migrations. In general, passive attacks attempt to steal sensitive data or passwords. The active attacks manipulate the kernel data structures. *Intrusion detection systems* (IDS) can be network-based (NIDS) or host-based (HIDS)..Program shepherding can be applied in control and verify the code execution. Other defense technology include using RIO dynamic optimization infrastructure, or Vmware vSafe and vShield tools, security compliance for hypervisors, and using Intel vPro Technology. Others apply a separate hardened closed OS environment or use isolated execution and sandboxing

Cloud Defense Methods: Virtualization enhances cloud security. But *virtual machines* (VMs) add an additional layer of software that could become a single-point of failure. With virtualization, a single physical machine can be divided or partitioned into multiple VMs (e.g. server consolidation). This provides each VM with better security isolation and each partition is protected from DoS attacks by other partitions Security attacks in one VM are isolated and contained from affecting the other VMs. VM failures do not propagate to other VMs. Hypervisor provides the visibility of the guest OS, with complete guest isolation. Fault containment and failure isolation of VMs provide a more secure and robust environment.

In Table 7.8, eight protection schemes are listed to secure public clouds and datacenters. Malicious intrusions may destroy valuable hosts, network, and storage resources. Internet anomalies found in routers, gateways, and distributed hosts may stop cloud services. Trust negotiation is often done at the SLA level. *Public Key Infrastructure* (PKI) service could be augmented with datacenter reputation systems. Worm and DdoS attacks must be contained. It is harder to establish security in the cloud due to the fact all data and software are shared by default. However, it is possible to add social tools like reputation systems to prove any trust model chosen to deploy.

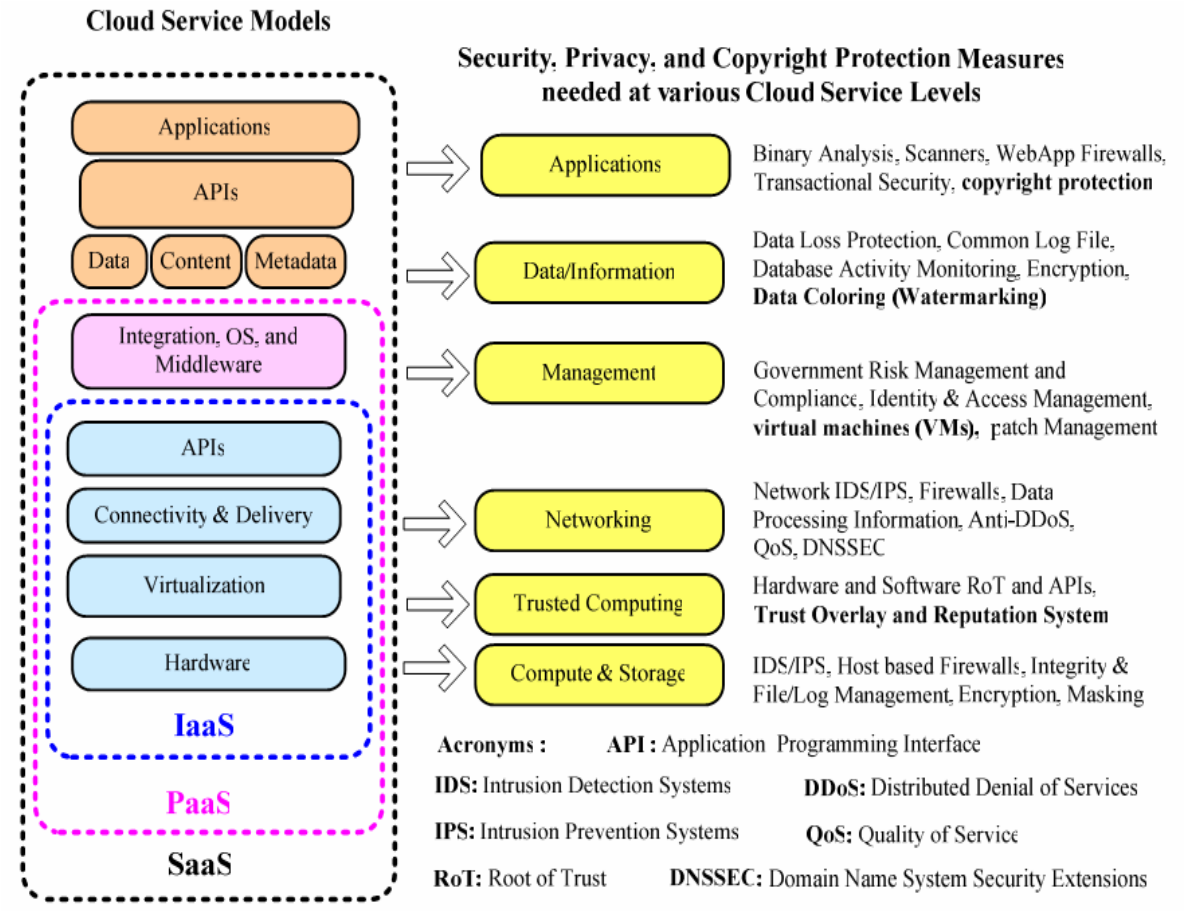


Figure 7.321: Cloud service models on the left and corresponding security measures on the right: The IaaS is at the inner-most level, PaaS at the mid-level, and SaaS at the widest level including all resources. (Courtesy of Hwang and Li, *IEEE Internet Cloud*, September 2010, [30])

In virtual machine safe cloning, we anticipate an attack by knowing when an attack has occurred. The snapshot control is based on the defined RPO. We need new security mechanisms to protect cloud. For example, one can apply secured information logging, migrate over secured VLAN, and apply ECC based encryption for secured migration. The sandbox provides a safe execution platform for running the programs. Further, Sandbox can provide a tightly controlled set of resources for the guest operating systems, which allows a security testbed to run the untested code and programs from the untrusted third party vendors.

Defense with Virtualization : The VM is decoupled from the physical hardware. The entire VM can be represented as a software component and can be regarded as a binary or digital data. VM can be saved, cloned, encrypted, moved, or restored with ease. VMs enable a higher availability and faster disaster recovery. We suggest live migration of VMs specifically designed for building *distributed intrusion detection system* (DIDS). Multiple IDS virtual machines can be deployed at various resource sites including the datacenters. DIDS design demands trust negation among PKI domains. Security policy conflicts must be resolved at design time and updated periodically.

Table 7.8: Physical and Cyber Security Protection at Cloud/Datacenters

Protection Schemes	Brief Description and Deployment Suggestions
Secure datacenters and computer buildings	Chose hazard-free location, enforce building safety. Avoid windows, keep buffer zone around the site, bomb detection, camera surveillance, earthquake-proof, etc.
Use redundant utilities at multiple sites	Multiple power and supplies, alternate network connections, multiple databases at separated sites, data consistency, data watermarking, user authentication, etc.
Trust delegation and Negotiation	Cross certificates must be used to delegate trust across PKI domains for various data- centers. Trust negotiation among thet CAs need to resolve policy conflicts
Worm containment and DdoS Defense	Internet worm containment and distributed defense against DdoS attacks are necessary to secure all datacenters and cloud platforms.
Reputation system for Datacenters	Reputation system could be built with P2P technology. One can build a hierarchy of reputation systems from datacenters to distributed file systems.
Fine-grain file access control	This refers to fine-grain access control at the file or object level. This adds up the security protection beyond firewalls and intrusion detection systems.
Copyright Protection and Piracy prevention	Piracy prevention achieved with peer collusion prevention, filtering of poisoned contents, non-destructive read, alteration detection, etc.
User and data privacy Protection	Use double authentication, biometric identification, intrusion detection and disaster recovery, privacy enforcement by data watermarking, data classification, etc

Example 7.5: Cloud Security Deployment by Vordel Company

Figure 7.32 shows the security defense system deployed by Vordel for protecting the access of some commercial clouds that are widely to general public. The firewall provides an external shielding. The Vordel XML gateway secures the application server, message queue, database, web service client and browser with HTTP, JMS, SQL, XML and SSL security protocols, respectively**.** Defense scheme is needed to protect user data from server attacks. The user private data must not be leaked to other users without permission. ■

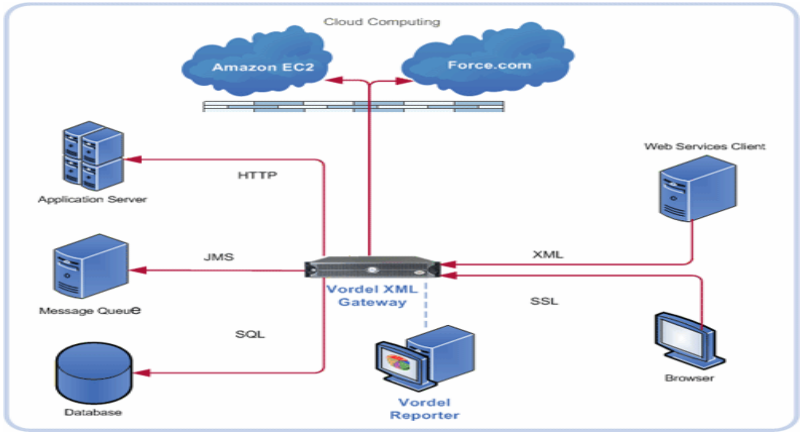


Figure 7.32 Vordel installed to secured XML gateway plus external firewall to safeguard the access of the commercial Amazon EC2 and Forece.com cloud platforms.

Privacy and Copyright Protection: The user gets predictable configuration before actual system integration. Yahoo's Pipes is a good example of lightweight cloud platforms. With shared files and datasets, privacy, security, and copyright could be compromised in a cloud-computing

environment. We desire to work in a software environment that provides many useful tools to build cloud applications over large data sets. Google platform essentially applies in-house software to protect resources. The Amazon EC2 applies HMEC and X.509 certificates in securing resources. We want to protect browser-initiated application software in the cloud environment. We identify below several security features that are desired in the cloud.

- Dynamic web services with full support from secure web technologies.
- Establish the trust between users and providers through SLA and reputation systems
- Effective user identity management and data-access management
- Single sign-on and single sign off to reduce security enforcement overhead
- Auditing and copyright compliance through proactive enforcement
- Shifting the control of data operations from client environment to cloud providers
- Protection of sensitive and regulated information in a shared externally managed environment

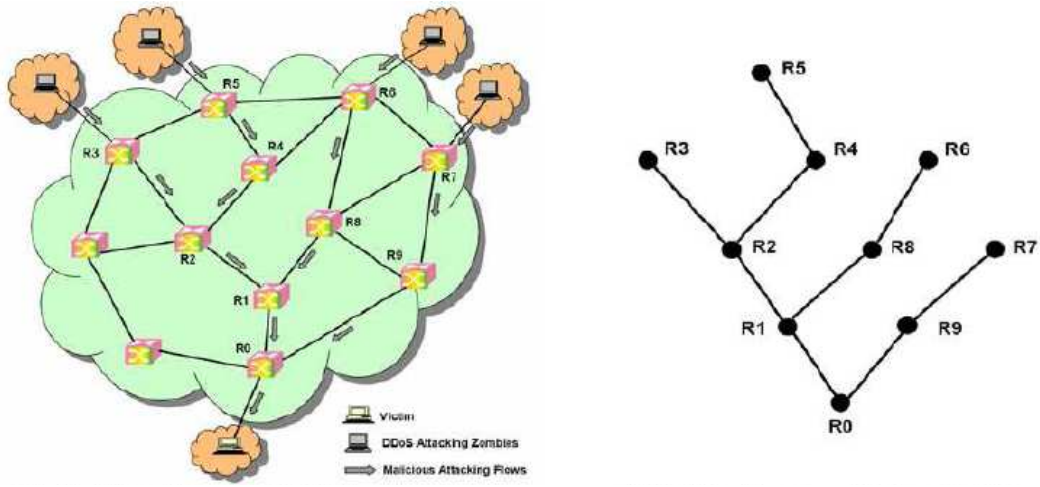
7.6.2 Distributed Intrusion and Anomaly Detection

The data security is the weakest link in all cloud models. We need new cloud security standards to apply common API tools to cope with the data lock-in problem and various network attacks or abuses. Extensive benchmark experiments are needed to validate the design for security enforcement. The IaaS model represented by Amazon Elastic cloud is most sensitive to external attacks. The role-based interface tools alleviate the complexity of the provisioning system. For example, The IBM BlueCloud provisions through a role-based web portal. A SaaS bureau may order secretarial services from a common cloud platform. Many IT companies are now offering cloud services with almost no guaranteed security or trustworthiness.

Security threats may be aimed at VMs, guest OS, and software running on top of the cloud. *Intrusion detection system* (IDS) attempts to stop the attacks before they take effect. Both signature matching and anomaly detection can be implemented on VMs dedicated for building IDS. Signature-matching IDS is more mature to build, but require frequent update of the signature databases. Network anomaly detection reveal abnormal traffic patterns, such as unauthorized episode of TCP connection sequences, against the normal traffic patterns. Distributed IDSs are needed to combat both types of intrusions. .

Distributed Defense against DDoS Flooding Attacks : A DDoS defense system must be designed to cover multiple network domains spanned by a given cloud platform. These network domains cover the edge networks where the protected cloud resources are physically connected. DDoS attacks come with widespread worms. The flooding traffic is large enough to crash the victim server by buffer overflow, disk exhaustion, or connection saturation, etc. Figure 7.33(a) shows a flooding attack pattern launched from 4 zombies towards a victim server attached at the bottom router R_0. The flooding traffic flows essentially form a tree pattern as shown in Fig,7.33(b). Successive attack-transit routers along the tree detect the abnormal surge of traffic at their I/O ports. This DDoS defense system is based on change-point detection along all attack-transit routers. The defense scheme works across multiple network domains.

In a DDoS flooding attack, the attacker often recruits many zombies over the Internet. The flooding traffic flows through multiple AS (*autonomous system*) domains before reaching the edge network where the victim machine is attached. Routers at the upstream domains observe the suspicious traffic flows ahead of routers at the downstream domains. Figure 6.34 illustrates an example network covering six AS domains. The victim server is located in the AS1 domain. Zombies are scattered widely in the Internet. By detecting abnormal traffic changes, the detection server at each domain creates a traffic flow subtree locally. Multiple traffic subtrees are merged to form the global traffic flow tree. Once the global tree is detected at the end router, all servers in the covered domains will be alerted with the eminant DDoS attacks. The Packet dropping could stop attack from the suspected upstream routers beamed towards the victim.



(a) Traffic flow pattern of a DDoS attack **(b) The attack traffic flow tree along 10 routers**

Figure 7.33 DDoS attacks and defense by change-point detection at all routers on the flooding tree (Courtesy of Chen, Hwang and Ku, *IEEE TPDS,* Dec. 2007 [11])

To sum up, the above defense system detects abnormal traffic changes at attack-transit routers. Based on the anomaly pattern detected in covered network domains, the scheme detects a DDoS attack before the victim is overwhelmed. The detection scheme is suitable for deployment at the cloud provider core networks. The provider-level cooperation eliminates the need of intervention by edge networks. Experimental results reported in [13] prove that 4 to 8 domains are sufficient to yield a 98% detection success rate of TCP SYN and UDP flooding attacks. Based on Internet AS domain distribution, the defense scheme can scale well to protect almost one hundred AS domains in a real-life Internet environment.

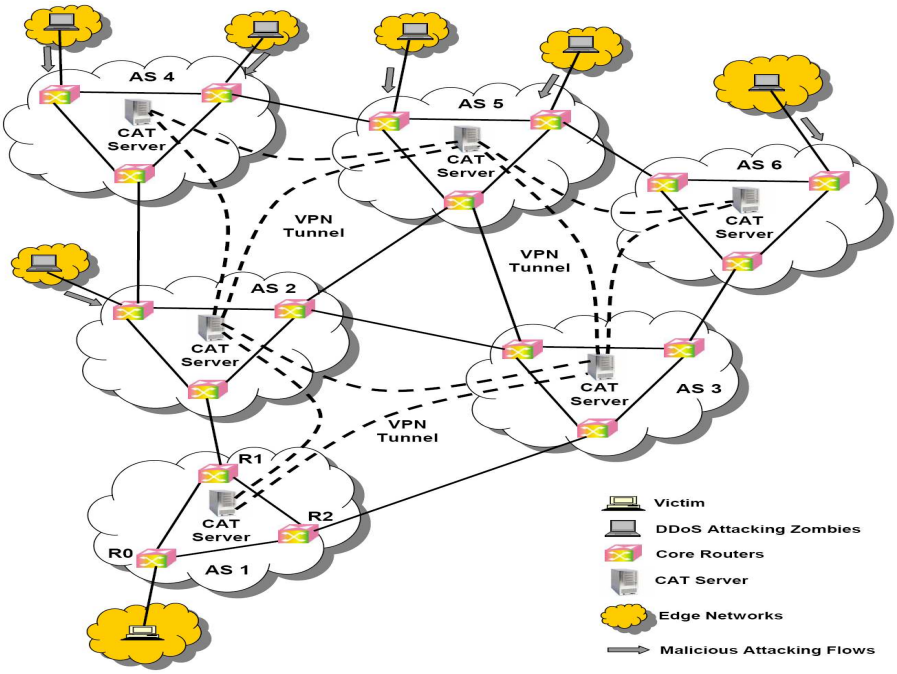


Figure 7.34 Distributed defense against DDoS attacks over multiple network domains (Courtesy of Chen, Hwang, and Ku, *IEEE Trans.of Parallel and Distributed Systems,* Dec. 2007 [13])

Man-in-The–Middle Attacks: The virtual machine migration from physical host machine VMM -A to the host machine VMM-B, via a security vulnerable network. Man In the Middle attack over the network can view the contents getting migrated, steal sensitive data or even modify the VM specific contents including the OS and application states. Figure 7.35 shows the VM migration from one physical host machine to another host. An attacker posing the man-in-the-middle attack can launch active attack to insert a VM-based rootkit (VMBR) into the migrating VM, which can subvert the entire operation of the migration process without the knowledge of the guest OS and embedded application.

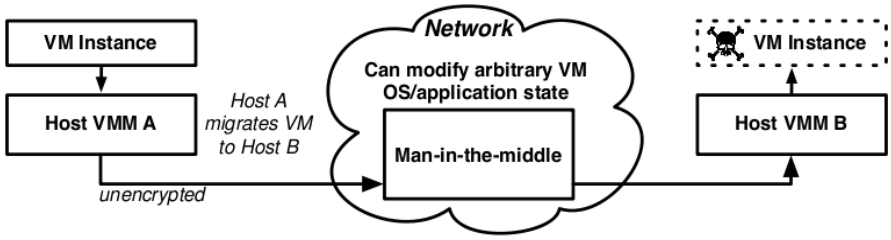


Figure 7.35 A virtual machine (VM) migrating from a host A to a host B through a vulnerable network threatened by man-in-the-middle attack to modify the VM template and OS state.

7.6.3 Reputation-Guided Protection of Datacenters

Trust is a personal opinion, which is very subjective and often biased. Trust can be transitive but not necessarily symmetric between two parties. Reputation is a public opinion, which is more objective and often relies on a large opinion aggregation process to evaluate. Reputation may change or decay over time. Recent reputation should be given more preference than the old images. In this section, we assess reputation systems for protecting datacenters or cloud user communities.

Reputation System Design Options: Figure 7.36 gives an overview of design options of reputation systems. The public opinion on the character or standing (such as honest behavior or reliability) of an entity could be the reputation of a person, an agent, a product, or a service. It represents a collective evaluation by a group of people/agents and resource owners. Many reputation systems have been proposed in the past mainly for P2P, multi-agent, or e-commerce systems.

To address the reputation systems for cloud services, a systematic approach is based on the design criteria and administration of the reputation systems. Figure 7.37 shows a two-tier classification of existing reputation systems that have been proposed in recent years. Most of them were designed for peer-to-peer or social networks. These reputation systems can be converted for protecting cloud computing applications. In general, the reputation systems are classified as *centralized* or *distributed* depending on how they are implemented. In a centralized system, a single central authority is responsible for managing the reputation system, while the distributed model involves multiple control centers working collectively. Reputation-based trust management and techniques for securing P2P and social networks could be merged to defend datacenters and cloud platforms against attacks from the open network.

A centralized reputation system is easier to implement, but demand more powerful and reliable server resources; while a distributed reputation system is much more complex to build. Distributed systems are more scalable and reliable to handle failures. At the second tier, reputation systems are further classified by the scope of reputation evaluation. The *user-oriented* reputation systems focus on individual users or agents. Most P2P reputation system belongs to this category. In datacenters, the reputation is modeled for the resource site as a whole. This reputation applies to products or services offered by the cloud. Ccommercial reputation systems have been built by e-Bay, Google, and Amazon in connection with the services they provided. These are centralized reputation systems.

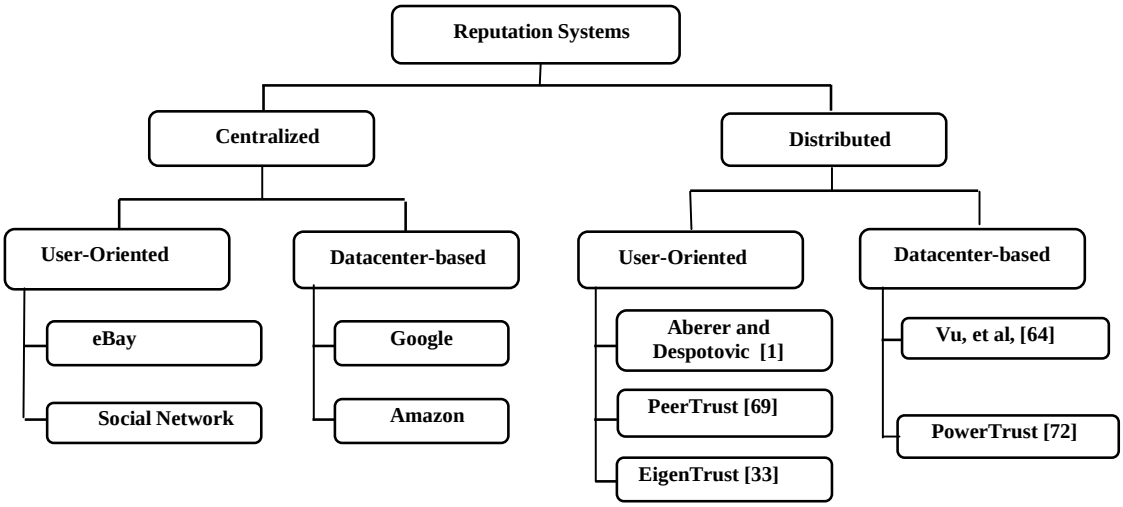


Figure 7.36 Design options of reputation systems for social networks and cloud platforms.

The distributed reputation systems are most developed by the academic research communities. Aberer and Despotovic have propose a model to mange trust in P2P systems. The Eigentrust reputation system was developed at Stanford University using trust matrix approach. The PeerTrust system was developed at Geogia Institute of Technology for supporting e-commerce applications. The PowerTrust system was developed at University of Southern California based on Power law characteristic of Internet traffic for P2P applications. Vu, et al proposed a QoS-based ranking system for P2P transactions.

To redesign the above reputation systems for protecting datacenters offers new opportunities for their expanded applications beyond the P2P networks. Data consistency is checked across multiple databases. Copyright protection secures wide-area content distributions. To separate user data from specific SaaS programs, the providers take the most responsibility in maintaining data integrity and consistency. Users can switch among different services using their own data. Only the users have the keys to access the requested data. The data objects must be uniquely named to ensure global consistency. To ensure data consistency, unauthorized updates of data objects by other cloud users are prohibited

The reputation system can be implemented with a trust overlay network. A hierarchy of P2P reputation systems is suggested to protect cloud resources at the site level and data objects at the file level. This demands both coarse-grain and fine-grained access control of shared resources. These reputation systems keep track of security breaches at all levels. The reputation system must be designed to benefit both cloud users and the datacenters. Data objects used in cloud computing reside in multiple datacenters over a *storage-area network* (SAN).

In the past, most reputation systems were designed for peer-to-peer social networking or for on-line shopping services. These reputation systems can be converted to protect cloud platform resources or user applications on the cloud. A centralized reputation system is easier to implement, but demand more powerful and reliable server resources. Distributed reputation systems are more scalable and reliable to handle failures. The five security mechanisms presented earlier can be greatly assisted by using the reputation system specifically designed for datacenters.

Trust Overlay Networks: Reputation represents a collective evaluation by users and resource owners. Many reputation systems have been proposed in the past for P2P, multi-agent, or e-commerce systems. To support trusted cloud services, we suggest to build a *trust overlay network* to model the trust relationships among datacenter modules. This trust overlay could be structured with DHT (*distributed hash table*) to

achieve fast aggregation of the global reputations out of a large number of local reputation scores. This trust overlay design was first introduced in [12]. Here, we suggest to extend the design to have two layers for fast reputation aggregation, updating, and dissemination to all users.

Figure 7.37 shows the construction of the two layers of the trust overlay network for this purpose. At the bottom layer is the trust overlay for distributed trust negotiation and reputation aggregation over multiple resource sites. This layer handles user/server authentication, access authorization, trust delegation, and data integrity control. At the top layer is an overlay for fast virus/worm signature generation and dissemination and for piracy detection. This overlay facilitates worm containment and intrusion detection systems against virus, worm, and DDoS attacks. The content poisoning technique presented by Lou and Hwang [6] is also reputation-based. That protection scheme can be easily extended to stop copyright violations in a cloud environment surrounding the datacenters.

The reputation system enables trusted interactions between cloud users and datacenter owners. Privacy is enforced by matching colored user identification with the colored data objects. We suggest the use of content poisoning to protect copyright of digital contents. The security-aware cloud architecture (Fig.7.16) is specially tailored to protect virtualized cloud infrastructure. The trust of provided cloud platforms comes from not only SLA (service-level agreements), but also effective enforcement of security policies and deployment of countermeasures to defend against network attacks. By varying the security control standards, one can cope with the dynamic variation of the cloud operating conditions. We aim at a trusted cloud environment to assure high-quality services with guaranteed throughput at the task level.

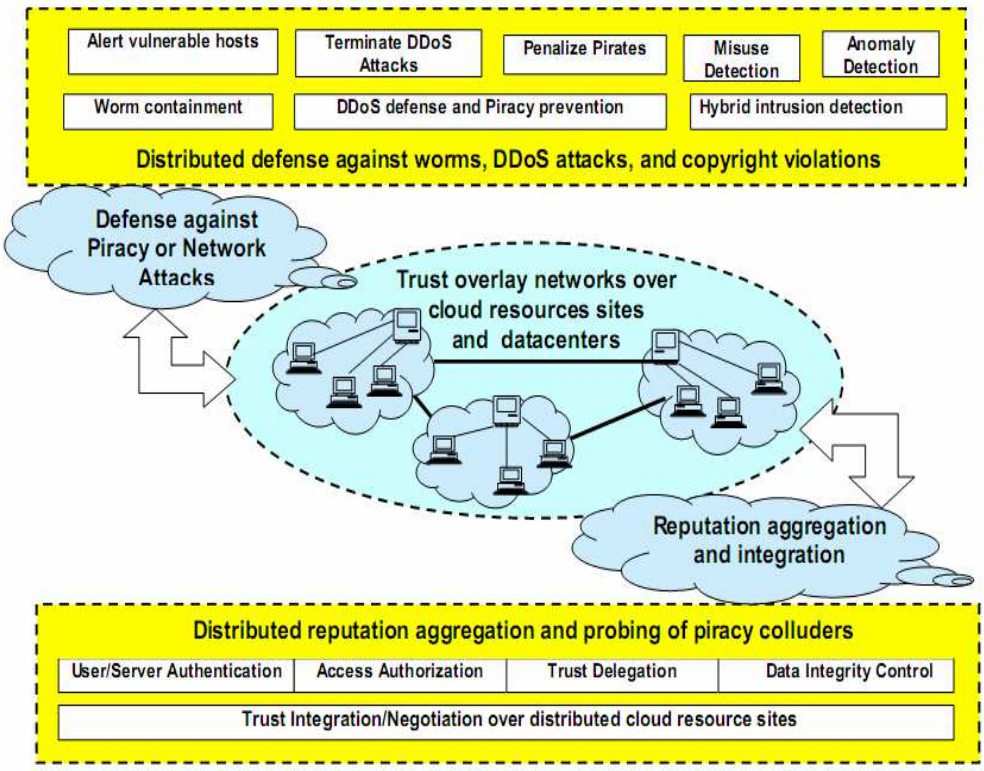


Figure 7.37 DHT-based trust overlay networks built over cloud resources provisioned from multiple datacenters for trust management and distributed security enforcement

The development trend is to apply virtualization support for security enforcement in cloud or datacenter environments. Both reputation systems and data watermarking mechanisms are designed to protect datacenter access at coarse-grain level and to limit the data access at the fine-grain file level. In the long run, we expect a new *Security as a Service* (SaaS) to become available. This "SaaS" is crucial to the universal acceptance of web-scale cloud computing in personal, business, community, and government applications. Internet clouds are certainly in line with IT globalization and efficient computing outsourcing. However, the interoperability among different clouds relies on a common operational standard by building a healthy cloud ecosystem.

7.7 References and Homework Problems

Tutorials of cloud computing can be found in [14] and Wikipedia [65]. Datacenters are introduced in Wikipedia [67]. Three recent books on cloud computing by Linthicum [39], Rittinghouse and Ransome [54], and Velte, et al [61] address cloud architecture, technologies, and implementations. Warehouse-scale datacenters are treated by Barroso and Holzle [7]. Articles related to datacenter architecture and implementations can be found in Al-Fares, et al [2], Greenberg, et al [24], Guo, et al [25], Nelson, et al [45], and Wu, et al [68]. Green IT is treated in [17, 22, 23, 52].

Articles arguing about the cloud definition can be found in [20, 41, 71]. Benefits and opportunities of cloud computing are treated in [26, 36, 44, 53]. Pros and cons of cloud computing coexist in the service industry. Some argued that the cloud is nothing but a big rental station in the sky. Others think that the cloud offers the efficiency and flexibility in making desktop computing centralized. We argue that cloud computing can consolidate the advantages from both centralized and distributed computing [47].

Cloud computing infrastructures are reported by both academia [4, 17, 34, 47, 56, 59] and by IT industry or enterprises [3, 6, 8, 21, 32, 35, 57, 60]. Cloud technology for HPC or HTC was studied in [27, 48, 55]. Virtualization support for cloud computing is studied in [10, 16, 59, 63]. Live VM migration and disaster recovery have been treated in [13, 46, 49, 63]. Cloud security is treated in the book by Mather, et al [43], comprehensively. Business clouds are introduced by Chou in a 2010 book [12] with a lot of case studies of success and failure stories,

Reputation systems studied in [10, 33, 50, 62, 66, 69, 72] could be modified for protecting datacenters and cloud resource sites. Data coloring and cloud watermarking are studied in [16, 37, 38, 70]. Trust models are proposed in [1, 19, 28, 42, 58, 64] for cloud, pervasive computing, and e-commerce applications. Security and privacy issues are discussed in [9, 11, 15, 29, 30, 31, 40, 51]. A global cloud computing testbed, called *Open Circus,* was recently reported in [5]. A cloud simulator, CloudSim, is available from the University of Melbourne [56].

References:

1. K. Aberer and Z. Despotovic, "Managing Trust in a Peer-to-Peer Information System", *ACM CIKM International Conference on Information and Knowledge Management,* 2001.

2. M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Datacenter Network Architecture," *Proc. of the ACM SIGCOMM 2008 Conference on Data Communication,* Seattle, WA, August 17–22, 2008.`

3. Amazon EC2 and S3, "Elastic Compute Cloud (EC2) and Simple Scalable Storage (S3)", http://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud, http://spatten_presentations.s3.amazonaws.com/s3-on-rails.pdf

4. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, R. Karp, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", *Technical Report* No. UCB/EECS-2009-28, University of California at Berkley, USA, Feb. 10, 2009.

5. I. Arutyun, et al , "Open Circus: A Global Cloud Computing Testbed", *IEEE Computer Magazine,* April 2010, pp35-43.

6. L. Barroso, J. Dean, and U. Holzle, "Web search for a planet: the architecture of the Google cluster," *IEEE Micro,* April 2003. doi:10.1109/MM.2003.1196112

7. L. Barroso and U. Holzle, *The Datacenter as A Computer: An Introduction to The Design of Warehouse-Scale Machines,* Morgan Claypool Publisher, 2009

8. G. Boss, P. Mllladi, et al, "Cloud Computing- The BlueCloud Project ", www.ibm.com/ developerworks/ websphere/zones/hipods/, Oct. 2007.

9. M. Cai, K. Hwang, J. Pan and C. Papadupolous, "WormShield: Fast Worm Signature Generation with Distributed Fingerprint Aggregation", *IEEE Trans. of Dependable and Secure Computing* (TDSC), Vol.4, No. 2, April/June 2007, pp.88-104.

10. Chase, et al, "Dynamic Virtual Clusters in a Grid Site Manager*", IEEE 12-th Symposium on High-Performance Distributed Computing (HPDC)* , 2003.

11. Y. Chen, K. Hwang, and W. S. Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains", *IEEE Trans. on Parallel and Distributed Systems* , Vol. 18, No.12, Dec. 2007, pp.1649-1662.

12. T. Clou, *Introduction To Computing Computing: Business and Technology,* Active Book Press, 2010.

13. C. Clark, K. Fraser, J. Hansen, E. Jul, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines", *Proc. of Symp. on Networked Systems Design and Implementation.* Boston, May 2, 2005. 273 – 286.

14. Cloud Computing Tutorial, http://www.thecloudtutorial.com. Jan. 2010.

15. Cloud Security Alliance, "Security guidance for Critical Areas of Focus in Cloud Computing", April 2009

16. C. Collberg and C. Thomborson, "Watermarking, Temper-Proofing, and Obfuscation-Tools for Software Protection", *IEEE Trans. Software Engineering,* Vol.28, 2002, pp.735-746

17. A. Costanzo, M. Assuncao, and R. Buyya, "Harnessing Cloud Technologies for a Virtualized Distributed Computing Infrastructure", *IEEE Internet Computing,* Sept. 2009.

18. D. Dyer, "Current Trends/Challenges in Datacenter Thermal Management—a Facilities Perspective," *presentation at ITHERM,* San Diego, CA, June 1, 2006.

19. Q. Feng, K. Hwang, and Y. Dai, *"Rainbow Product ranking for Upgrading e-Commerce", IEEE Internet Computing,* Sept. 2009.

20. I. Foster, Y. Zhao, J.Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," *Grid Computing Environments Workshop,* 12-16 Nov. 2008.

21. Google Inc., "Efficient Data Center Summit, April 2009". Available at http://www.google.com/corporate/green/datacenters/summit.html.

22. Green Grid, "Quantitative analysis of power distribution configurations for datacenters". Available at http://www.thegreengrid.org/gg_content/.

23. The Green 500, Available at http://www.green500.org.

24. A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, " The Cost of a Cloud: Research Problems in Datacenter Networks", *ACM SIGCOMM Computer Communication Review,* Vol.39, No.1, Jan. 2009.

25. C. Guo, G. Lu, et al, "BCube: A High-Performance Server-Centric Network architecture for Modular Datacenters", *ACM SIGCOMM Computer Communication Review,* Vol.39, No.44, Oct. 2009.

26. E. Hakan, "Cloud Computing: Does Nirvana Hide behind the Nebula?," *IEEE Software,* March 2009.

27. C. Hoffa, et al., "On the Use of Cloud Computing for Scientific Workflows," *IEEE Fourth Int'l Conf. on eScience,*Dec. 2008

28. R. He, J. Hu , J. Niu, and M. Yuan, "A Novel Cloud-Based Trust Model for Pervasive Computing", *Fourth Int'l Conf. on Computer and Information Technology,* Sept. 14-16 2004, pp. 693 - 700

29. K. Hwang, S. Kulkarni, and Y. Hu, "Cloud Security with Virtualized Defense and Reputation-based Trust Management*", IEEE Int'l Conf. on Dependable, Autonomic, and Secure Computing* (DASC09), Chengdu, China, Dec.12-14, 2009.

30. K. Hwang and D. Li, ""Security, Privacy, and Data protection for Trusted Cloud Computing", *IEEE Internet Computing,* Sept. 2010.

31. K. Hwang, M. Cai, Y. Chen, and M. Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes", *IEEE Trans. on Dependable and Secure Computing,* Vol.4, No.1, Jan-March, 2007, pp.41-55.

32. V. Jinesh , "Cloud Architectures," White Paper, Amazon, http://aws.amazon.com/about-aws/whats-new/2008/07/16/cloud-architectures-white-paper/

33. D. Kamvar, T. Schlosser and H. Garcia-Molina," The EigenTrust Algorithm for Reputation Management in P2P Networks", *Proc. of the 12th Int'l Conf. on World Wide Web,* 2003.

34. K. Keahey, M. Tsugawa, and A. Matsunaga, and j. Fortes, "Sky Computing", *IEEE Internet Computing,* Sept. 2009.

35. G. Lakshmanan, "Cloud Computing : Relevance to Enterprise", Infosys Technologies, Inc. 2009.

36. N. Leavitt, et al, "Is Cloud Computing Really Ready for Prime Time*?," IEEE Computer* , vol.42, no.1, pp.15-20, Jan. 2009

37. D. Li, H. Meng, and X. Shi, "Membership Clouds and Membership Cloud Generator", *Journal of Computer Research and Development,* Vol.32, No.6, 1995, pp.15-20.

38. D. Li, C. Liu, and W. Gan, "A New Cognitive Model: Cloud Model", *Int'l Journal of Intelligent Systems,* March 2009.

39. D. Linthicum, *Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide,* Addison Wesley Professional, ISBN 0=-13-600922-0, 2009.

40. X. Lou and K. Hwang, "Collusive Piracy Prevention in P2P Content Delivery Networks", *IEEE Transactions. on Computers,* July 2009.

41. M. Luis, Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner , "A Break in the Clouds: Towards a Cloud Definition". *ACM SIGCOMM Computer Communication Review Archive,* Jan. 2009.

42. D. Manchala, "E-Commerce Trust Metrics and Models", *IEEE Internet Computing,* April 2000.

43. T. Mather, et al, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance,* O'Reilly Media, Inc. 2009.

44. L. Mei, W. Chan, and T. Tse, "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues," IEEE *Asia-Pacific Services Computing Conference,* Dec. 2008.

45. D. Nelson, M. Ryan, S. DeVito, K. V. Ramesh, P. Vlasaty, B. Rucker, B. Da y Nelson, et al., "The Role of Modularity in Datacenter design". Sun BluePrints Online, http://www.sun.com/storagetek/docs/EED.pdf.

46. M Nelson, BH Lim, G Hutchins. "Fast Transparent Migration for Virtual Machines", *Proceedings of the USENIX 2005 Annual Technical Conference.* Anaheim, CA, April 10-15, 2005. 391–394.

47. D. Nurmi, Rich Wolski, et al, "Eucalyptus: An Elastic Utility Computing Architecture Linking Your Programs to Useful Systems," *UCSB Computer Science Technical Report* No.2008-10. Aug. 2008

48. W. Norman, M. Paton, T. de Aragao, K. Lee, A. Alvaro . Fernandes, and R. Sakellarious. "Optimizing Utility in Cloud Computing through Autonomic Workload Execution" *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 2009.

49. D. A. Patterson, et al, "Recovery-oriented computing (ROC): Motivation, Definition, Techniques, and Case Studies". UC Berkeley CS *Technical Report UCB//CSD-02-1175,* March 15, 2002.

50. M. Pujol, et al, "Extracting Reputation in Multi-agent Systems by Means of Social Network Topology", *Proc. of Int'l Conf. on Autonomous Agents and Multi-Agents Systems,* 2002, .

51. S. Roschke, F. Cheng, and C. Meinel, "Intrusion Detection in The Cloud", *IEEE Int'l Conf on Dependable, Autonomic, and Secure Computing* (DASC 09) , Dec.13, 2009.

52. R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No 'power' struggles: coordinated multi-level power management for the datacenter', *Proc. of the ACM Int'l Conf. on Architectural Support for Programming Languages and Operating Systems,* Seattle, WA, March 2008.

53. D. Reed, "Clouds, Clusters and Many Core: The Revolution Ahead," *Cluster Computing, 2008 IEEE International Conference on* , vol., no., pp.1-1, Sept. 29 2008-Oct. 1 2008

54. J. Rittinghouse and J. Ransome, *Cloud Computing: Implementation, Management and Security,* CRC Publisher, 2010.

55. B. Rochwerger, D. Breitgand, E. Levy A. et al, "The RESERVOIR Model and Architecture for Open Federated Cloud Computing". In *IBM Systems Journal,* 2008

56. N. Rodrigo, Calheiros, R. Ranjan, A. Cesar , De Rose, and R. Buyya, " CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services", *Technical Report,* GRIDS-TR-2009-1, University of Melbourne, Australia, March 13, 2009.

57. Salesforce.com, http://en.wikipedia.org/wiki/Salesforce.com / , Jan. 2010

58. S. Song, K. Hwang, R. Zhou, and Y. Kwok, "Trusted P2P Transactions with Fuzzy Reputation Aggregation", *IEEE Internet Computing,* Special Issue on Security for P2P and Ad Hoc Networks, Nov. 2005, pp. 24-34.

59. B. Sotomayor, R. Montero, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds", *IEEE Internet Computing,* Sept. 2009.

60. C. Vecchiola, X. Chu, and R. Buyya, "Aneka: A Software Platform for .NET-based Cloud Computing", *High Speed and Large Scale Scientific Computing,* pp. 267-295 in Gentzsch, et al. (Eds.), ISBN: 978-1-60750-073-5, IOS Press, Amsterdam, Netherlands, 2009.

61. T. Velte, A. Velite, and R. Elsenpeter, *Cloud Computing, A Practical Approach,* MsGraw-Hill Osborne Media, ISBN 0-07-162694-8, 2010.

62. K. Vlitalo and Y. Kortesniemi, "Privacy in Distributed Reputation Management", *Workshop of the 1st Int'l Conf. on Security and Privacy for Emerging Areas in Comm. Networks,* 2005. Sept. 2005.

63. VMWare, "Disaster Recovery Solutions from VMWare", White Paper, 2007

64. L. Vu, M. Hauswirth, and K. Aberer, "QoS-based Service Selection and Ranking with Trust and Reputation Management", *Proc. of On The Move Conference,* (OTM'05), LNCS 3760, 2005.

65. Wikipedia, "Cloud Computing", http://en.wikipedia.org/wiki/Cloud_computing, Jan.26, 2010.

66. Y. Wang and J. Vassileva, "Toward Trust and Reputation Based Web Service Selection: A Survey", *Journal of Multi-agent and Grid Systems* (MAGS)", special Issue on "New tendencies on Web Services and Multi-agent Systems (WS-MAS)", Jan 2007.

67. Wikipedia, "Data Center", http://en.wikipedia.org/wiki/Data_center, Jan.26, 2010.

68. H. Wu, G. Lu, D. Li, C. Guo and Y. Zhang. "MDCube: A High Performance Network Structure for Modular Data Center Interconnection", *ACM CoNEXT'09,* December 1–4, 2009, Rome, Italy

69. L. Xiong and L. Liu, " PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities",. *IEEE Transactions on Knowledge and Data Engineering,* July 2004, pp. 843-857.

70. J. Yang, J. Wang; C. Wang, and D. Li, "A Novel Scheme for Watermarking Natural Language Text," *Third Int'l Conf. on Intelligent Information Hiding and Multimedia Signal Processing,* 2007, pp.481-484.

71. L. Youseff, M. Butrico, D. Maria; and D. Silva, "Toward a Unified Ontology of Cloud Computing," *Grid Computing Environments Workshop,* 2008. GCE '08 , Nov. 2008, pp.1-10.

72. R. Zhou, and K. Hwang, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing", *IEEE Trans. Parallel and Distributed Systems,* April 2007.

Homework Problems :

Problem 7.1 Compile a table to compare public clouds and private clouds in each of the following aspects. Also identify their differences, advantages, and shortcomings in terms of development efforts and application flexibility. Give several example platforms that you know of under each cloud class.

- **(a)** Technology leverage and IT resources ownership
- **(b)** Management of provisioned resources including data, virtual machines, and computing environments.
- **(c)** Workload distribution methods and loading policies
- **(d)** Security precautions and data privacy enforcement

Problem 7.2 Describe the following techniques or terminologies used in cloud computing and cloud services. Dig out the a concrete example cloud or case study behind the addressed technology.

- **(a)** Service-level agreement (SLA)
- **(b)** Virtualized cloud infrastructure
- **(c)** Green information technology
- **(d)** Multi-tenant technique
- **(e)** Web-scale communication

Problem 7.3 Describe the following technology terms, functionality, and cloud services associated with the Google cloud platform.

- **(a)** Google file system (GFS)

(b) BigTable

(c) MapReduce

(d) Chubby

Problem 7.4 Check the Amazon Web Service (AWS) cloud web site. Plan a real computing application using the Elastic compute cloud (EC2). You must specify the resources requested and figure out the costs\to be charged by Amazon. If you have access of the AWS platform, carry out the EC2 experiments and report the performance measured against the cost of paid services.

Problem 7.5 Check the Amazon Web Service (AWS) cloud web site. Plan a real storage application using the Simple storage service (S3). You must specify the resources requested and figure out the costs to be charged by Amazon. If you have access of the AWS platform, carry out the S3 experiments and report the performance measured against the cost of paid services.

Problem 7.6 Check the Amazon Web Service (AWS) cloud web site. Plan a real storage application using the Simple queue service (SQS). You must specify the resources requested and figure out the costs to be charged by Amazon. If you have access of the AWS platform, carry out the S3 experiments and report the performance measured against the cost of paid services.

Problem 7.7 Describe the following technology terms, functional modules, and cloud services associated the IBM Blue Cloud and RC2 Cloud. You may need to dig our more information from IBM web sites or *IBM Journal of Research and Development* for reported achievements.

(a) IBM WebSphere

(b) Service-oriented architecture (SOA)

(c) Tivoli service automation manager (TSAM)

(d) IBM Research Compute Cloud (RC2)

(e) Tivoli provisioning manager

(f) IBM x-server architecture

(g) Linux with XEN

Problem 7.8 Consider two cloud systems: Google File System and Amazon S3. Explain how they achieve their design goals to secure data integrity and to maintain data consistency while facing the problems of hardware failure, especially concurrent hardware failures

Problem 7.9 Conduct a Google search for technical specification of the following two private clouds. Describe what you have found out from reliable sources of information. Try to reveal the cloud platform architecture and its infrastructure bases and cope of major applications.

(a) The private cloud under development by NASA at the USA

(b) The private cloud under development by CERN in Europe

Problem 7.10 Elaborate four major advantages of using virtualized resources in cloud computing applications. Your discussion should address the resource management issues from provider's perspectives and the application flexibility, cost-effectiveness, and dependability concerns by cloud users.

Problem 7.11 Answer the following questions on the InterGrid Gateway (IGG) experiments. Read the original paper [5] for more details beyond the material in Section 6.3.3-4 on cloud creation and load peering

experiments in a distributed virtualized cloud infrastructure built on top of the French Grid'5000 system.

(a) Study the details of the IGG software components and its usage to link resource sites

(b) Repeat the IGG experiments in a local small-scale network or grid environment by getting the IGG software from University of Melbourne.

(c) Using the cloud simulator: CloudSim from University of Melbourne or write your own Simulator to repeat the same experiments in Part (b).

Problem 7.12 Explain the following terminologies associated with datacenter design and managements :.

(a) Differences in warehouse-scale datacenters and modular datacenters

(b) Study three datacenter architecture papers in [2, 24, 25, 45, 68] and report on their contributions to advance datacenter performance and dependability.

(c) Discuss the scalability of those datacenter architectures you have studied in Part (b).

(d) Discuss how system virtualization can enhance the reliability of datacenter operations.

Problem 7.13 Explain the differences in the following two machine recovery schemes. Comment on their implementation requirements, advantages and shortcomings, and application potentials.

(a) Recovery of a physical machine failure by another physical machine

(b) Recovery of a virtual machine failure by another virtual machine

Problem 7.14 You have studied the basic concepts of live migration of virtual machines and disaster recovery in Section 3.5.2 and Section 6.3.3. Read the related articles in [10, 16, 59, 63] and answer the following questions based on your research findings :

(a) Why VM live migration is important in disaster recovery of datacenter or cloud operations ?

(b) What virtualization support is needed to achieve fast cloning of VMs ? Explain how VM cloning can enable fast recovery ?

(c) What are RTO (recovery property objective) and RTO (recovery time objective) in the design of disaster recovery scheme ? Explain the role of snapshots in disaster recovery schemes.

Problem 7.15 Briefly describe the following technology terms, functional modules, and cloud services associated with the SGI Cyclone cloud engine :

(a) High-performance computing cloud

(b) Scale-up, scale-out, and hybrid management of cloud servers

(c) SGI Altix servers and ICE clusters

(d) Reported Cyclone HPC applications

Problem 7.16 Briefly describe the following technology terms, functional modules, and cloud services associated the Google App Engine (GAE). Visit the following GAE web site for lot more details.

Google App Engine Front Page: http://code.google.com/appengine/

(a) Datastore for data storage service

(b) Applications runtime environment

(c) Software development kit (SDK)

(d) Compute cloud versus storage cloud

Problem 7.17 This is an extended research problem on an efficient datacenter network by Al-Fare, et al [1], that we have studied in Fig.6.10 in Section 6.2.2. The scalable datacenter architecture was proposed to extend the fat tree concept. You are asked to perform the following tasks:

(a) Study these two papers and justify the claimed network features on scalability, efficient routing, and compatibility with the Ethernet, IP, and TCP protocols.

(b) Can you suggest means to improve the network in fault tolerance, cross-section bandwidth, and implementation efficiency based on today's technology.

Problem 7.18 This is also an extended research problem on efficient datacenter networks. In the papers by Guo, et al [25] and Wu, et al [68], an interconnection network, called MDCube, was proposed to connect many modular datacenter containers for building mega datacenters as shown in Fig.6.13 in Section 6.2.4.

(a) Discuss the advantages of using this network to improve the inter-container bandwidth, reducing cost of interconnection structure, and cabling complexity.

(b) Justify the claimed low diameter, high capacity, and fault tolerance of the MDCube network design for interconnecting datacenter containers.