<u>UNIT – III</u>

ARCHITECTURAL DESIGN OF COMPUTE AND STORAGE CLOUDS

A Generic Cloud Architecture Design

Cloud architecture is intended to process massive amounts of data with a high degree of parallelism. The following are the major design goals of a cloud architecture:

Scalability – cloud can be easily expanded by adding more servers and enlarging the network connectivity accordingly.

Virtualization - The cloud management software needs to support both physical and virtual machines

Efficiency - The hardware and software systems are combined to make it easy and efficient to operate.

Reliability - Data can be put into multiple locations. In such a situation, even if one of the data centers crashes, the user data is still accessible.

Security in shared resources and shared access of data centers also pose another design challenge.

Clouds support Web 2.0 applications. Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud.

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 system utilization via multiplexing, virtualization, and dynamic resource provisioning.

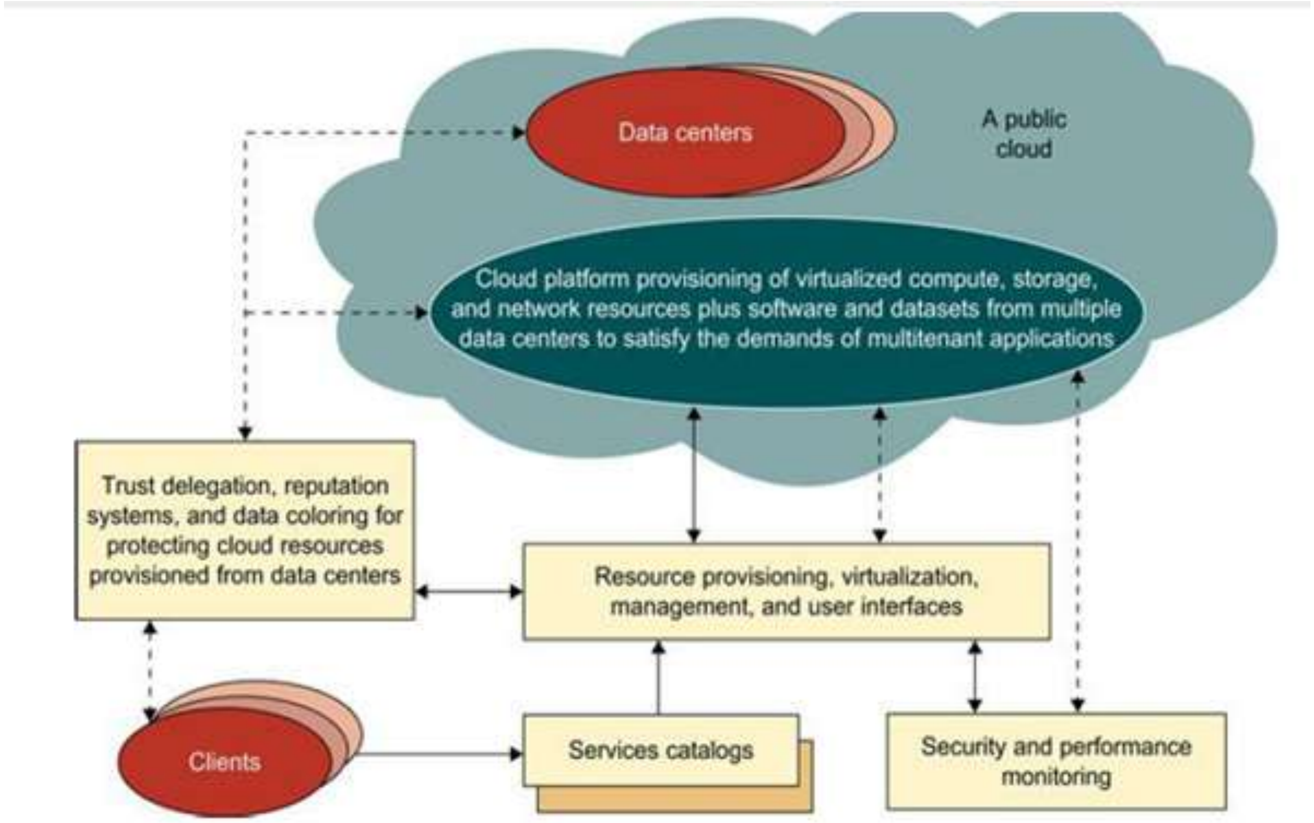
Cloud-Enabling Technologies in Hardware, Software, and Networking

- Fast platform Deployment – flexible deployment of cloud resources
- Virtual clusters on demand – to be provided and cluster need to be reconfigured as workload changes
- Multitenant techniques –refers distributing software to a large no. of users simultaneously
- Massive data processing –internet search and web services required this process
- Web-scale communication – support for e-commerce, education, social networking, medical care etc.
- Distributed storage –large scale storage of data in distributed storage
- Licensing and billing services

Rapid progress in multicore CPUs, memory chips, and disk arrays has made it possible to build faster data centers with huge amounts of storage space. Resource virtualization enables rapid cloud deployment and disaster recovery. *Service-oriented architecture* (SOA) also plays a vital role. Today's clouds are designed to serve a large number of tenants over massive volumes of data. The availability of large-scale, distributed storage systems is the foundation of today's data centers. Of course, cloud computing is greatly benefited by the progress made in license management and automatic billing techniques in recent years.

A Generic Cloud Architecture

Figure shows security-aware cloud architecture. 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 data-center resources.


Data centers
A public cloud
Cloud platform provisioning of virtualized compute, storage, and network resources plus software and datasets from multiple data centers to satisfy the demands of multitenant applications
Trust delegation, reputation systems, and data coloring for protecting cloud resources provisioned from data centers
Resource provisioning, virtualization, management, and user interfaces
Clients
Services catalogs
Security and performance monitoring

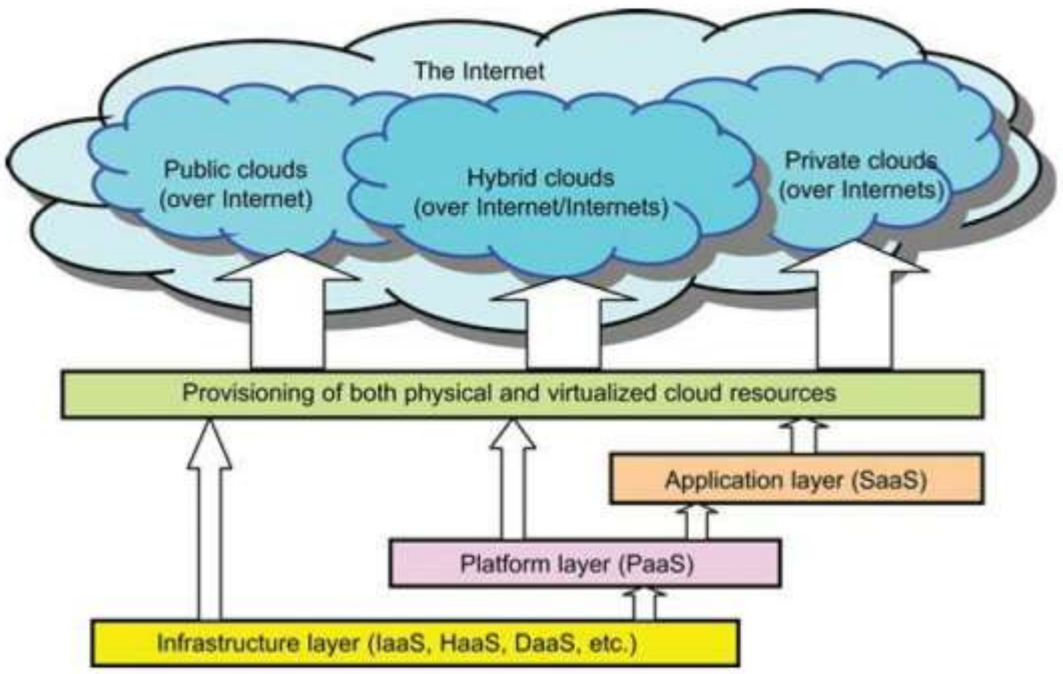
The cloud platform is formed dynamically by provisioning or deprovisioning servers, software, and database resources. Servers in the cloud can be physical machines or VMs. In addition to building the server cluster, the cloud platform demands distributed storage and accompanying services. The cloud computing resources are built into the data centers, which are typically owned and operated by a third-party provider.

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 storage area networks (SANs), 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 provisioned resources.

The software infrastructure of a cloud platform must handle all resource management and do most of the maintenance automatically. So�ware must detect the status of each node server joining and leaving, and perform relevant tasks accordingly. In general, private clouds are easier to manage, and public clouds are easier to access. The trends in cloud development are that more and more clouds will be hybrid.

Layered Cloud Architectural Development

The architecture of a cloud is developed at three layers: infrastructure, platform, and application, as demonstrated in figure.


The Internet
Public clouds (over Internet)
Hybrid clouds (over Internet/Internets)
Private clouds (over Internets)
Provisioning of both physical and virtualized cloud resources
Application layer (SaaS)
Platform layer (PaaS)
Infrastructure layer (IaaS, HaaS, DaaS, etc.)

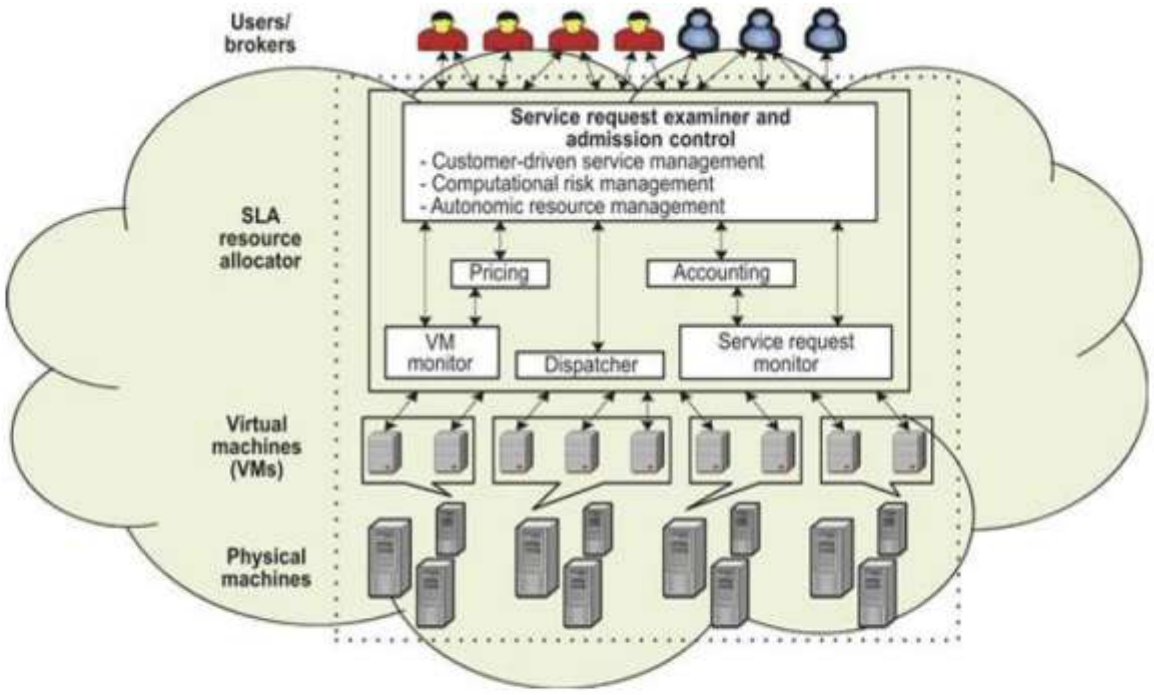
These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud. The services to public, private, and hybrid clouds are conveyed to users through networking support over the Internet and intranets involved. It is clear that the infrastructure layer is deployed first to support IaaS services. This infrastructure layer serves as the foundation for building the platform layer of the cloud for supporting PaaS services. In turn, the platform layer is a foundation for implementing the application layer for SaaS applications. Different types of cloud services demand application of these resources separately.

The infrastructure layer is built with virtualized compute, storage, and network resources. The platform layer is for general-purpose and repeated usage of the collection of software resources. This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance. 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. The application layer is also heavily used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transactions, and supply chain management. In general, SaaS demands the most work from the provider, PaaS is in the middle, and IaaS demands the least.

Market-Oriented Cloud Architecture

Cloud providers consider and meet the different QoS parameters of each individual consumer as negotiated in specific SLAs. Market-oriented resource management is necessary to regulate the supply and demand of cloud resources. The designer needs to provide feedback on economic incentives for both consumers and providers. The purpose is to promote QoS-based resource allocation mechanisms. Figure shows the high level architecture for supporting market-oriented resource allocation in a cloud computing environment.


Users/ brokers
Service request examiner and admission control
- Customer-driven service management
- Computational risk management
- Autonomic resource management
SLA resource allocator
Pricing
Accounting
VM monitor
Dispatcher
Service request monitor
Virtual machines (VMs)
Physical machines

Users or brokers submit service requests to the data center and cloud to be processed. The SLA resource allocator acts as the interface between the data center/cloud service provider and external users/brokers. When a service request is first submitted the service request examiner interprets the request for QoS requirements before determining whether to accept or reject the request.

The request examiner ensures that there is no overloading of resources, after that it assigns requests to VMs and determines resource entitlements for allocated VMs. The Pricing mechanism decides how service requests are charged. Pricing serves as a basis for managing the supply and demand of computing resources within the data center and facilitates in prioritizing resource allocations effectively. The Accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to users. In addition, the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.

The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements. The Dispatcher mechanism starts the execution of accepted service requests on allocated VMs. The Service Request Monitor mechanism keeps track of the execution progress of service requests. Multiple VMs can be started and stopped on demand on a single physical machine to meet accepted service requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service request.

Quality of Service Factors

There are critical QoS parameters to consider in a service request, such as time, cost, reliability, and trust/security. QoS requirements cannot be static and may change over time due to continuing changes in business operations. Negotiation mechanisms are needed to respond to alternate offers protocol for establishing SLAs.

Commercial cloud offerings must be able to support customer-driven service management based on customer profiles and requested service requirements. Commercial clouds define computational risk management tactics to identify, assess, and manage risks involved in the execution of applications with regard to service requirements and customer needs. The system incorporates autonomic resource management models that effectively self-manage changes in service requirements to satisfy both new service demands and existing service obligations, and leverage VM technology to dynamically assign resource shares according to service requirements.

Architectural Design Challenges

We will identify six open challenges in cloud architecture development.

1—Service Availability and Data Lock-in Problem

The management of a cloud service by a single company is source of single points of failure. Multiple cloud providers are to be used to achieve HA. Even if a company has multiple data centers 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 which make services unavailable to intended users. Some utility computing services offer SaaS providers the opportunity to defend against DDoS attacks by using quick scale-ups.

Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary. The obvious solution is to standardize the APIs so that a SaaS developer can 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 data center of a private cloud.

2—Data Privacy and Security Concerns

Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks. Many obstacles can be overcome immediately with well understood technologies such as encrypted storage, virtual LANs, and network middleboxes (e.g., firewalls, packet filters). Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries.

Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms. In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits. Another type of attack is the man-in-the-middle attack for VM migrations. In general, passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

3—Unpredictable Performance and Bottlenecks

Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic. It is required to improve 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. Data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed for minimizing the cost.

4—Distributed Storage and Widespread Software Bugs

The database is always growing in cloud applications. The opportunity is to create a storage system that will not only meet this growth, but also combine it with the cloud advantage of scaling arbitrarily up and down on demand. This demands the design of efficient distributed SANs. Data centers must meet programmers' expectations in terms of scalability, data durability, and HA. Data consistence checking in SAN-connected data centers 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 data centers. No data center will provide such a convenience. One solution may be a reliance on using VMs in cloud computing. The level of virtualization may make it possible to capture valuable information in ways that are impossible without using VMs. Debugging over simulators is another approach to attacking the problem, if the simulator is well designed.

5—Cloud Scalability, Interoperability, and Standardization: GAE automatically scales in response to load increases and decreases; users are charged by the cycles used. AWS charges by the hour for the number of VM instances used, even if the machine is idle. In order to save the money, scale up and down must happen quickly. Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of VMs.

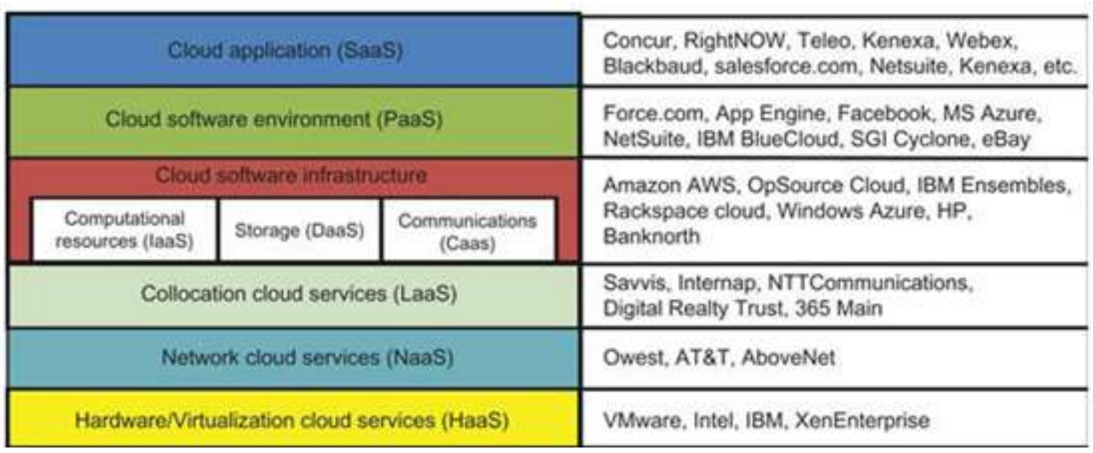
6—Software Licensing and Reputation Sharing: Many cloud computing providers originally relied on open source software because the licensing model for commercial software is not ideal for utility computing. An opportunity would be to create reputation-guarding services similar to the "trusted e-mail" services currently offered (for a fee) to services hosted on smaller ISPs. Cloud providers want legal liability to remain with the customer, and vice versa. This problem must be solved at the SLA level.

Inter-cloud Resource Management

Cloud resource management and intercloud resource exchange schemes are reviewed in this section.

Extended Cloud Computing Services

Figure shows six layers of cloud services, ranging from hardware, network, and collocation to infrastructure, platform, and software applications. The cloud platform provides PaaS, which sits on top of the IaaS infrastructure. The top layer offers SaaS. These must be implemented on the cloud platforms provided.


Cloud application (SaaS)
Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.
Cloud software environment (PaaS)
Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay
Cloud software infrastructure
Computational resources (IaaS)
Storage (DaaS)
Communications (Caas)
Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth
Collocation cloud services (LaaS)
Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main
Network cloud services (NaaS)
Owest, AT&T, AboveNet
Hardware/Virtualization cloud services (HaaS)
VMware, Intel, IBM, XenEnterprise

The bottom three layers are more related to physical requirements. The bottommost layer provides Hardware as a Service (HaaS). The next layer is for interconnecting all the hardware components, and is simply called Network as a Service (NaaS). Virtual LANs fall within the scope of NaaS. The next layer up offers Location as a Service (LaaS), which provides a collocation service to house, power, and secure all the physical hardware and network resources. Some authors say this layer provides Security as a Service. 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.

The cloud players are divided into three classes: (1) cloud service providers and IT administrators, (2) software developers or vendors, and (3) end users or business users. These cloud players vary in their roles under the IaaS, PaaS, and SaaS models. From the software vendors' perspective, application performance on a given cloud platform is most important. From the providers' perspective, cloud infrastructure performance is the primary concern. From the end users' perspective, the quality of services, including security, is the most important.

Cloud Service Tasks and Trends

The top layer is for SaaS applications, mostly used for business applications. The approach is to widen market coverage by investigating customer behaviors and revealing opportunities by statistical analysis. SaaS tools also apply to distributed collaboration, and financial and human resources management.

PaaS is provided by Google, Salesforce.com, and Facebook, among others. IaaS is provided by Amazon, Windows Azure, and RackRack, among others. Collocation services require multiple cloud providers to work together to support supply chains in manufacturing. Network cloud services provide communications such as those by AT&T, Qwest, and AboveNet. Details can be found in Clou's introductory book on business clouds.

Software Stack for Cloud Computing

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 layers just as the traditional software stack does. However, the lower layers are not completely transparent to the upper layers.

The platform for running cloud computing services can be either physical servers or virtual servers. By using VMs, the platform can be flexible, that is, the running services are not bound to specific hardware platforms. This brings flexibility to cloud computing platforms. The software layer on top of the platform is the layer for storing massive amounts of data. This layer acts like the file system in a traditional single machine. Other layers running on top of the file system are the layers for executing cloud computing applications. They include the database storage system, programming for large-scale clusters, and data query language support. The next layers are the components in the software stack.

Runtime Support Services

As in a cluster environment, there are also some runtime supporting services in the cloud computing environment. Cluster monitoring is used to collect the runtime status of the entire cluster. One of the most important facilities is the cluster job management system. The scheduler queues the tasks submitted to the whole cluster and assigns the tasks to the processing nodes according to node availability. The distributed scheduler for the cloud application has special characteristics that can support cloud applications, such as scheduling the programs written in MapReduce style. The runtime support system keeps the cloud cluster working properly with high efficiency. Runtime support is software needed in browser-initiated applications applied by thousands of cloud customers. The SaaS model provides the software applications as a service, rather than letting users purchase the software

Resource Provisioning and Platform Deployment

In this section, we will discuss techniques to provision computer resources or VMs and storage allocation schemes.

Provisioning of Compute Resources

Providers offer cloud services by signing SLAs with end users. The SLAs must commit sufficient resources such as CPU, memory, and bandwidth that the user can use for a preset period. Underprovisioning of resources will lead to broken SLAs and penalties. Overprovisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider. Deploying an autonomous system to efficiently provision resources to users is a 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. Resource provisioning schemes also demand fast discovery of services and data in cloud computing infrastructures. In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration, and fast recovery from failures. To deploy VMs, users treat them as physical hosts with customized operating systems for specific applications. 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 data centers.

Resource Provisioning Methods

There are three cases of static cloud resource provisioning policies. In case (a),overprovisioning with the peak load causes heavy resource waste (shaded area). In case(b), underprovisioning (along the capacity line) of resources results in losses by both user and provider In case (c), the constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste. Three resource-provisioning methods are presented here:

- *demand-driven method*
- *event-driven method*
- *popularity-driven method*

Demand-driven method

This method adds or removes computing instances based on the current utilization level of the allocated resources. In general, when a resource has surpassed a threshold for a certain amount of time, the scheme increases that resource based on demand. When a resource is below a threshold for a certain amount of time,that resource could be decreased accordingly. Amazon implements such an auto-scale feature in its EC2 platform.

Event-driven method

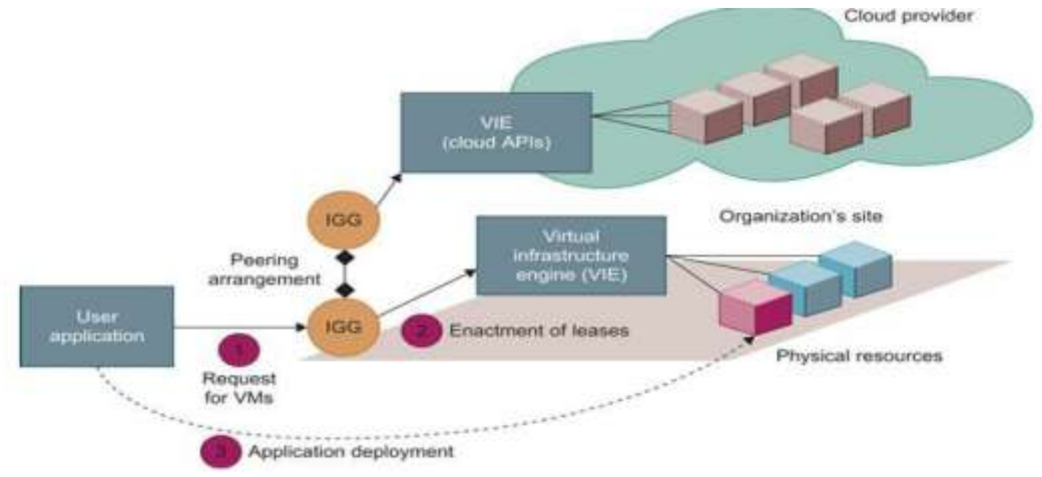
This scheme adds or removes machine instances based on a specific time event. The scheme works better for seasonal or predicted events such as Christmastime in the West and the Lunar New Year in the East. During these events, the number of users grows before the event period and then decreases during the event period. This scheme anticipates peak traffic before it happens. The method results in a minimal loss of QoS, if the event is predicted correctly.

Popularity-driven method

In this method, the Internet searches for popularity of certain applications and creates the instances by popularity demand. The scheme anticipates increased traffic with popularity. Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct. Resources may be wasted if traffic does not occur as expected.

Dynamic Resource Deployment

The cloud uses VMs as building blocks to create an execution environment across multiple resource sites. 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 enable the allocation of resources from multiple grids to establish the execution environment. In figure, a scenario is illustrated by which an Intergrid gateway (IGG) allocates resources from a local cluster to deploy applications in three steps: (1) requesting the VMs, (2) enacting the leases, and (3) deploying the VMs as requested. Under peak demand, this IGG interacts with another IGG that can allocate resources from a cloud computing provider.



grid has predefined peering arrangements with other grids, which the 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. 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.

The InterGrid allocates and provides 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 behalf of specific user applications. The core component of the IGG is a scheduler for implementing 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.

Provisioning of Storage Resources

The data storage layer is built on top of the physical or virtual servers. provider. The service can be accessed anywhere in the world. One example is e-mail systems. A typical large e-mail system might have millions of users and each user can have thousands of e-mails and consume multiple gigabytes of disk space. Another example is a web searching application. In storage technologies, hard disk drives may be augmented with solid-state drives in the future. The biggest barriers to adopting flash memory in data centers have been price, capacity, and, to some extent, a lack of sophisticated query-processing techniques.

Distributed file system is very important for storing large-scale data. However, other forms of data storage also exist. Some data does not need the namespace of a tree structure file system, and instead, databases are built with stored data files. In cloud computing, another form of data storage is (Key, Value) pairs.

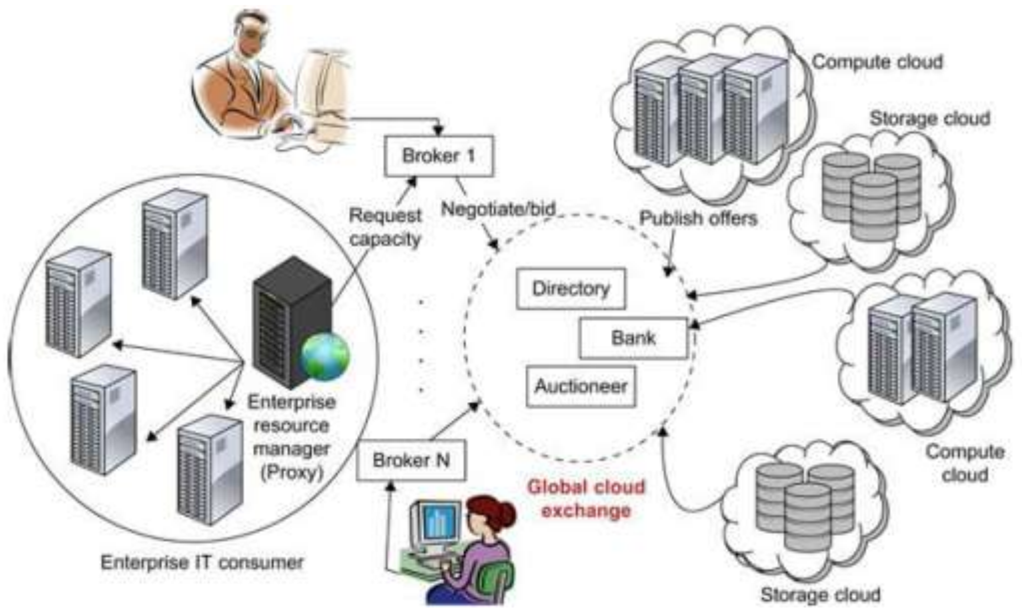
Many cloud computing companies have developed large-scale data storage systems to keep huge amount of data collected every day. For example, Google’s GFS stores web data and some other data, such as geographic data for Google Earth. A similar system from the open source community is the Hadoop Distributed File System (HDFS) for Apache. Hadoop is the open source implementation of Google’ s cloud computing infrastructure. Similar systems include Microsoft’ s Cosmos file system for the cloud.

The main purpose is to store the data in structural or semi-structural ways so that application developers can use it easily and build their applications rapidly. Web pages are an example of semistructural data in HTML format. For example, applications might want to process the information contained in a web page. Traditional databases will meet the performance bottleneck while the system is expanded to a larger scale. However, some real applications do not need such strong consistency. The scale of such databases can be quite large. Typical cloud databases include BigTable from Google, SimpleDB from Amazon, and the SQL service from Microsoft Azure.

Global Exchange of Cloud Resources

In order to support a large number of application service consumers from around the world, cloud infrastructure providers have established data centers in multiple geographical locations. For example, Amazon has data centers in the United States (e.g., one on the East Coast and another on the West Coast) and Europe. Amazon does not provide seamless/automatic mechanisms for scaling its hosted services across data centers.

It is difficult for cloud customers to determine in advance the best location for hosting their services as they may not know the origin of consumers of their services. SaaS providers may not be able to meet the QoS expectations of their service consumers originating from multiple geographical locations. Figure shows the high-level components of the Melbourne group’s proposed InterCloud architecture



In order to meet the QoS expectations of customer, it is required make use of services of multiple cloud infrastructure service providers who can provide better support for their specific consumer needs. This necessitates federation of cloud infrastructure service providers for seamless provisioning of services across different cloud providers. Cloudbus Project at the University of Melbourne has proposed InterCloud architecture supporting brokering and exchange of cloud resources for scaling applications across multiple clouds.

System consists of client brokering and coordinator services that support utility-driven federation of clouds: application scheduling, resource allocation, and migration of workloads. The architecture cohesively couples the administratively and topologically distributed storage and compute capabilities of clouds as part of a single resource leasing abstraction. The system will ease the cross-domain capability integration for on-demand, flexible, energy-efficient, and reliable access to the infrastructure based on virtualization.

The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers. It aggregates the infrastructure demands from application brokers and evaluates them against the available supply currently published by the cloud coordinators. It supports trading of cloud services based on competitive economic models such as commodity markets and auctions. CEx allows participants to locate providers and consumers with fitting offers. An SLA specifies the details of the service to be provided in terms of metrics agreed upon by all parties, and incentives and penalties for meeting and violating the expectations, respectively. The availability of a banking system within the market ensures that financial transactions pertaining to SLAs between participants are carried out in a secure and dependable environment.