LECTURE NOTES

ON

CLOUD APPLICATION DEVELOPMENT

Course code: ACS011
B.Tech VII SEM (IARE-R16)

By
Dr.D Kishore Babu
Associate Professor

Mr. P Anjaiah
Assistant Professor

Mr. C Praveen Kumar
Assistant Professor

Ms. B Vijaya Durga
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
DUNDIGAL, HYDERABAD - 500 043

UNIT - I

INTRODUCTION AND CLOUD APPLICATION DEVELOPMENT

Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards. It is distinguished by the notion that resources are virtual and limitless and that details of the physical systems on which software runs are abstracted from the user.

Defining Cloud Computing:

Cloud computing takes the technology, services, and applications that are similar to those on the Internet and turns them into a self-service utility. The use of the word "cloud" makes reference to the two essential concepts:

Abstraction: Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous.
Virtualization: Cloud computing virtualizes systems by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

Computing as a utility is a dream that dates from the beginning of the computing industry itself. A set of new technologies has come along that, along with the need for more efficient and affordable computing, has enabled an on-demand system to develop. It is these enabling technologies that are the focal point of this book.

Many people mistakenly believe that cloud computing is nothing more than the Internet given a different name. Many drawings of Internet-based systems and services depict the Internet as a cloud, and people refer to applications running on the Internet as "running in the cloud," so the confusion is understandable. The Internet has many of the characteristics of what is now being called cloud computing. The Internet offers abstraction, runs using the same set of protocols and standards, and uses the same applications and operating systems. These same characteristics are found in an intranet,

an internal version of the Internet. When an intranet becomes large enough that a diagram no longer wishes to differentiate between individual physical systems, the intranet too becomes identified as a cloud.

Cloud computing is an abstraction based on the notion of pooling physical resources and presenting them as a virtual resource. It is a new model for provisioning resources, for staging applications, and for platform-independent user access to services. Clouds can come in many different types, and the services and applications that run on clouds may or may not be delivered by a cloud service provider. These different types and levels of cloud services mean that it is important to define what type of cloud computing system you are working with.

Cloud computing has changed the nature of commercial system deployment, consider these three examples:

Google: In the last decade, Google has built a worldwide network of datacenters to service its search engine. In doing so Google has captured a substantial portion of the world's advertising revenue. That revenue has enabled Google to offer free software to users based on that infrastructure and has changed the market for user-facing software. This is the classic Software as a Service case

Azure Platform: By contrast, Microsoft is creating the Azure Platform. It enables .NET Framework applications to run over the Internet as an alternate platform for Microsoft developer software running on desktops

Amazon Web Services: One of the most successful cloud-based businesses is Amazon Web Services, which is an Infrastructure as a Service offering that lets you rent virtual computers on Amazon's own infrastructure. These new capabilities enable applications to be written and deployed with minimal expense and to be rapidly scaled and made available worldwide as business conditions permit. This is truly a revolutionary change in the way enterprise computing is created and deployed.

Characteristics of Cloud Computing:

Cloud computing builds on so many older concepts in computer technology that it can be hard for people newly introduced to the concept to grasp that it represents a paradigm

shift in computing. It's an evolutionary change that enables a revolutionary new approach to how computing services are produced and consumed.

Paradigm shift

When you choose a cloud service provider, you are renting or leasing part of an enormous infrastructure of datacenters, computers, storage, and networking capacity. Many of these datacenters are multi-million-dollar investments by the companies that run them Most of the large cloud computing service providers have multiple datacenters located all over the world. An accurate count can be difficult to obtain, the location of some 20 datacenters in Amazon Web Service's cloud are detailed. Google's cloud includes perhaps some 35 datacenters worldwide.

In the 1960s, military initiative aimed at miniaturizing electronics funded many of the semiconductor production lines that led to advanced microprocessors, dense memory arrays, and the sophisticated integrated circuit technology that makes computers, mobile devices, and so much more possible today. In the 1990s, the commercialization of the Internet gave rise to some very large companies that were forced to build very large computing infrastructures to support their businesses.

Amazon.com's infrastructure was built to support elastic demand so the system could accommodate peak traffic on a busy shopping day such as "Black Monday." Because much of the capacity was idle, Amazon.com first opened its network to partners and then as Amazon Web Services to customers.

FIGURE 1.1 The Google Dalles, Oregon, datacenter shown in Google Earth is an industrial-sized information technology utility.

As these various datacenters grew in size, businesses have developed their datacenters as “green-field” projects. Datacenters have been sited to do the following:

- Have access to low cost power
- Leverage renewable power source
- Be near abundant water
- Be sited where high-speed network backbone connections can be made
- Keep land costs modest and occupation unobtrusive
- Obtain tax breaks
- Optimize the overall system latency

These characteristics make cloud computing networks highly efficient and capture enough margin to make utility computing profitable. It has been estimated that the Internet consumes roughly 10 percent of the world’s total power, so these companies are very big energy consumers. In some cases, such as Google, these companies may also become some of the major energy producers of the 21st century. Essentially what has happened is that the Internet has funded the creation of the first information technology utilities. That’s why cloud computing is such a big deal. According to the research firm IDC, the following areas were the top five cloud applications in use in 2010:

- Collaboration applications
- Web applications/Web serving
- Cloud backup
- Business applications
- Personal productivity applications

The last five years have seen a proliferation of services and productivity applications delivered online as cloud computing applications. Examples of the impact of cloud computing abound in your everyday life, although many people do not make the connection to what was once a straight forward client/server Internet deployment. Movement of these applications to the cloud has been transparent, and in many cases the older on-premises deployment is supported by the same applications hosted in the cloud.

For example, many people have used ChannelAdvisor.com for their auction listings and sales management. That site recently expanded its service to include a CRM connector to Salesforce.com. One of the largest call center operations companies is a cloud-based service, Liveops.com Cloud computing has shifted the economics of software delivery in a manner similar to the way that music downloads have shifted the delivery of commercial music. The cost advantages of cloud computing have enabled new software vendors to create productivity applications that they can make available to people at a much smaller cost than would be possible for shrink-wrapped software. Given the general demise of the big-box computer store along with many other traditional retail models, it has become increasingly difficult for vendors to get shelf space. You can visit your local Wal-Mart to get some sense of this issue.

This new model of computer application delivery has allowed vendors like Google to offer complete office suites to individuals for free, supported by its advertiser subscription model. Even Google's business offerings have had some major successes against industry leader Microsoft Office. Last year, Los Angeles County switched to Google Docs.

Benefits of cloud computing:

On-demand self-service: A client can provision computer resources without the need for interaction with cloud service provider personnel.
Broad network access: Access to resources in the cloud is available over the network using standard methods in a manner that provides platform-independent access to clients of all types. This includes a mixture of heterogeneous operating systems, and thick and thin platforms such as laptops, mobile phones, and PDA.
Resource pooling: A cloud service provider creates resources that are pooled together in a system that supports multi-tenant usage. Physical and virtual systems are dynamically allocated or reallocated as needed. Intrinsic in this concept of pooling is the idea of abstraction that hides the location of resources such as virtual machines, processing, memory, storage, and network bandwidth and connectivity.
Rapid elasticity: Resources can be rapidly and elastically provisioned. The system can add resources by either scaling up systems (more powerful computers) or scaling out systems (more computers of the same kind), and scaling may be automatic or manual. From the standpoint of the client, cloud computing resources should look limitless and can be purchased at any time and in any quantity.

Measured service: The use of cloud system resources is measured, audited, and reported to the customer based on a metered system.

A client can be charged based on a known metric such as amount of storage used, number of transactions, network I/O (Input/output) or bandwidth, amount of processing power used, and so forth. A client is charged based on the level of services provided.

While these five core features of cloud computing are on almost anybody's list, you also should consider these additional advantages:

Lower costs: Because cloud networks operate at higher efficiencies and with greater utilization, significant cost reductions are often encountered.

Ease of utilization: Depending upon the type of service being offered, you may find that you do not require hardware or software licenses to implement your service.

Quality of Service: The Quality of Service (QoS) is something that you can obtain under contract from your vendor.

Reliability: The scale of cloud computing networks and their ability to provide load balancing and failover makes them highly reliable, often much more reliable than what you can achieve in a single organization.

Outsourced IT management: A cloud computing deployment lets someone else manage your computing infrastructure while you manage your business. In most instances, you achieve considerable reductions in IT staffing costs.

Simplified maintenance and upgrade: Because the system is centralized, you can easily apply patches and upgrades. This means your users always have access to the latest software versions.

Low Barrier to Entry: In particular, upfront capital expenditures are dramatically reduced. In cloud computing, anyone can be a giant at any time. This very long list of benefits should make it obvious why so many people are excited about the idea of cloud computing. Cloud computing is not a panacea, however. In many instances, cloud computing doesn't work well for particular applications.

Major challenges faced by cloud computing:

- Cloud computing inherits some of the challenges of parallel and distributed computing at the same time, it faces major challenges of its own.
- The specific challenges differ for the three cloud delivery models, but in all cases the difficulties are created by the very nature of utility computing, which is based on resource sharing and resource virtualization and requires a different

trust model than the ubiquitous user-centric model we have been accustomed to for a very long time.

- The most significant challenge is security; gaining the trust of a large user base is critical for the future of cloud computing. It is unrealistic to expect that a public cloud will provide a suitable environment for all applications.
- Highly sensitive applications related to the management of the critical infrastructure, healthcare applications, and others will most likely be hosted by private clouds.
- Many real-time applications will probably still be confined to private clouds. Some applications may be best served by a hybrid cloud setup; such applications could keep sensitive data on a private cloud and use a public cloud for some of the processing.
- The SaaS model faces similar challenges as other online services required to protect private information, such as financial or healthcare services. In this case a user interacts with cloud services through a well-defined interface
- Data in storage is most vulnerable to attack, so special attention should be devoted to the protection of storage servers.

The IaaS model is by far the most challenging to defend against attacks. Indeed, an IaaS user has considerably more degrees of freedom than the other two cloud delivery models. An additional source of concern is that the considerable resources of a cloud could be used to initiate attacks against the network and the computing infrastructure. Virtualization is a critical design option for this model, but it exposes the system to new sources of attack. The trusted computing base (TCB) of a virtual environment includes not only the hardware and the hypervisor but also the management operating system.

The entire state of a virtual machine (VM) can be saved to a file to allow migration and recovery, both highly desirable operations; yet this possibility challenges the strategies to bring the servers belonging to an organization to a desirable and stable state. Indeed, an infected VM can be inactive when the systems are cleaned up, and it can wake up later and infect other systems. This is another example of the deep intertwining of desirable and undesirable effects of basic cloud computing technologies. The next major challenge is related to resource management on a cloud. Any systematic rather than ad hoc resource management strategy requires the existence of controllers tasked to

implement several classes of policies: admission control, capacity allocation, load balancing, energy optimization, and last but not least, to provide QoS guarantees.

To implement these policies the controllers need accurate information about the global state of the system. Determining the state of a complex system with 106 servers or more, distributed over a large geographic area, is not feasible.

Indeed, the external load, as well as the state of individual resources, changes very rapidly. Thus, controllers must be able to function with incomplete or approximate knowledge of the system state. It seems reasonable to expect that such a complex system can only function based on self-management principles. But self-management and self-organization raise the bar for the implementation of logging and auditing procedures critical to the security and trust in a provider of cloud computing services. Under self-management it becomes next to impossible to identify the reasons that a certain action that resulted in a security breach was taken.

- The last major challenge we want to address is related to interoperability and standardization. Vendor lock-in, the fact that a user is tied to a particular cloud service provider, is a major concern for cloud users (Standardization would support interoperability and thus alleviate some of the fears that a service critical for a large organization may not be available for an extended period of time. But imposing standards at a time when a technology is still evolving is not only challenging, it can be counterproductive because it may stifle innovation.

Cloud models:

Defining Infrastructure as a Service (IaaS):

We can broadly partition cloud computing into four layers that form a cloud computing ecosystem as shown in Figure. The Application layer forms the basis for Software as a Service (SaaS),while the Platform layer forms the basis for Platform as a Service (PaaS) models that are described in the next two sections.

- Infrastructure as a Service (IaaS) creates what may be determined to be a utility computing model, something that you can tap into and draw from as you need it without significant limits on the scalability of your deployment. You pay only for what you need when you need it.

- IaaS may be seen to be an incredibly disruptive technology, one that can help turn a small business into a large business nearly overnight. This is a most exciting prospect; one that is fueling a number of IaaS startups during one of the most difficult recessions of recent memory.
- Infrastructure as a Service (IaaS) is a cloud computing service model in which hardware is virtualized in the cloud. In this particular model, the service vendor owns the equipment: servers, storage, network infrastructure, and so forth.
- The developer creates virtual hardware on which to develop applications and services. Essentially, an IaaS vendor has created a hardware utility service where the user provisions virtual resources as required.

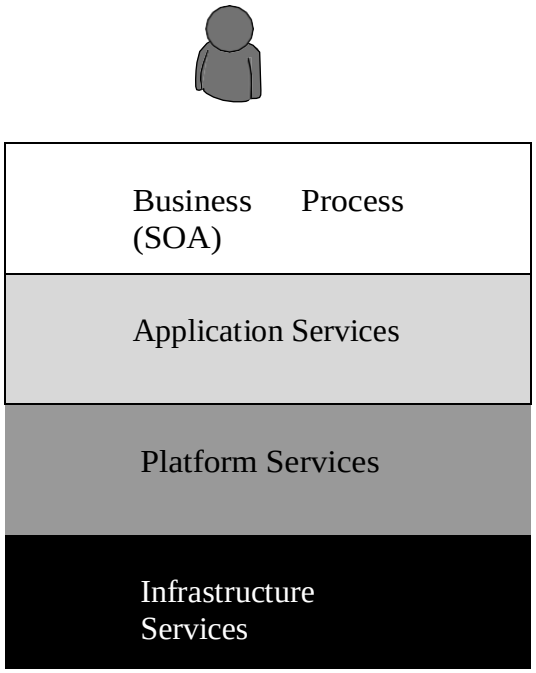

Fig 1.2 The cloud computing ecosystem

The developer interacts with the IaaS model to create virtual private servers, virtual private storage, virtual private networks, and so on, and then populates these virtual systems with the applications and services it needs to complete its solution. In IaaS, the virtualized resources are mapped to real systems. When the client interacts with an IaaS service and requests resources from the virtual systems, those requests are redirected to the real servers that do the actual work.

IaaS workloads:

The fundamental unit of virtualized client in an IaaS deployment is called a workload. A workload simulates the ability of a certain type of real or physical server to do an amount of work. The work done can be measured by the number of Transactions per Minute (TPM) or a similar metric against a certain type of system. In addition to throughput, a workload has certain other attributes such as Disk I/Os measured in Input/output Per Second IOPS, the amount of RAM consumed under load in MB, network throughput and latency, and so forth. In a hosted application environment, a client's application runs on a dedicated server inside a server rack or perhaps as a standalone server in a room full of servers.

In cloud computing, a provisioned server called an instance is reserved by a customer, and the necessary amount of computing resources needed to achieve that type of physical server is allocated to the client's needs. Figure shows how three virtual private server instances are partitioned in an IaaS stack. The three workloads require three different sizes of computers: small, medium, and large.

A client would reserve a machine equivalent required to run each of these workloads. The IaaS infrastructure runs these server instances in the data center that the service offers, drawing from a pool of virtualized machines, RAID storage, and network interface capacity.

These three layers are expressions of physical systems that are partitioned as logical units. LUNs, the cloud interconnect layer, and the virtual application software layer are logical constructs. LUNs are logical storage containers, the cloud interconnect layer is a virtual network layer that is assigned IP addresses from the IaaS network pool, and the virtual application software layer contains software that runs on the physical VM instance(s) that have been partitioned from physical assets on the IaaS' private cloud. From an architectural standpoint, the client in an IaaS infrastructure is assigned its own private network.

The Amazon Elastic Computer Cloud (EC2),behaves as if each server is its own separate network—unless you create your own Virtual Private Cloud(an EC2 add-on feature), which provides a workaround to this problem. When you scale your EC2 deployment, you are adding additional networks to your infrastructure, which makes it

easy to logically scale an EC2 deployment, but imposes additional network overhead because traffic must be routed between logical networks. Amazon Web Service's routing limits broadcast and multicast traffic because Layer-2 (Data Link) networking is not supported. Other IaaS infrastructures such as the one Cloudscaling.com offers or traditional VMware cloud-assigned networks on a per-user basis, which allows for Level 2 networking options. The most prominent Level 2 protocols that you might use are tunneling options, because they enable VLANs.

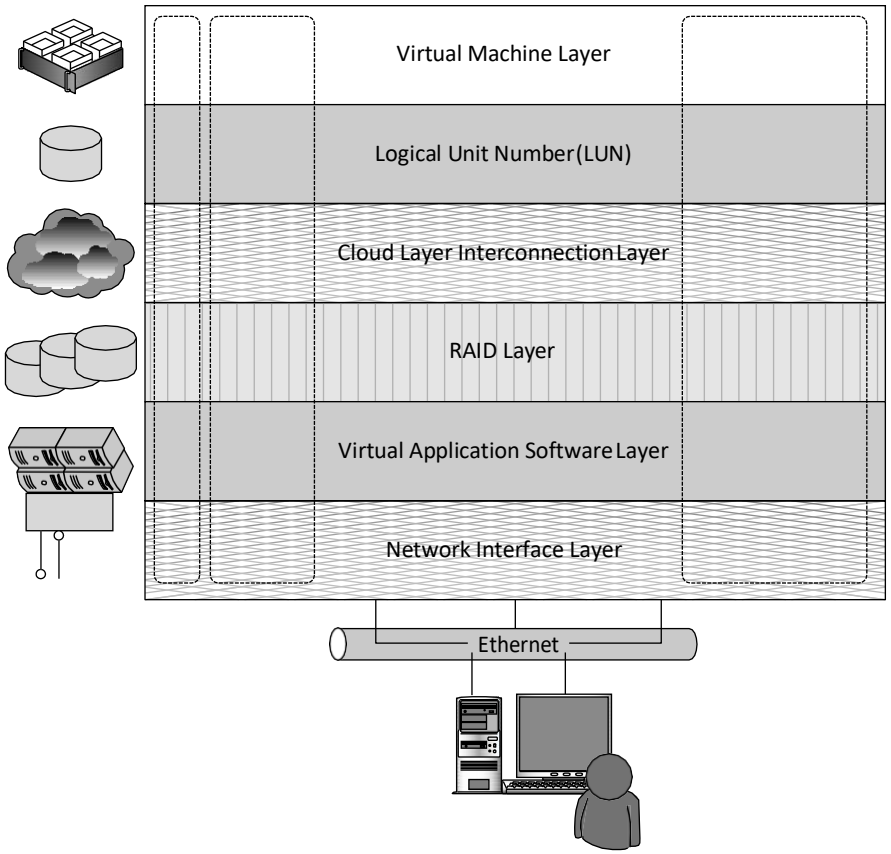

Figure 1.3 Virtual private server partition in an IaaS cloud

Consider a transactional ecommerce system, for which a typical stack contains the following components:

- Web server

- Application server
- File server
- Database
- Transaction engine

This ecommerce system has several different workloads that are operating: queries against the database, processing of business logic, and serving up clients' Web pages.

The classic example of an IaaS service model is Amazon.com's Amazon Web Services (AWS). AWS has several data centers in which servers run on top of a virtualization platform (Xen) and may be partitioned into logical compute units of various sizes. Developers can then apply system images containing different operating systems and applications or create their own system images. Storage may be partitions, databases may be created, and a range of services such a messaging and notification can be called upon to make distributed application work correctly.

Pods, aggregation, and silos

Workloads support a certain number of users, at which point you exceed the load that the instance sizing allows. When you reach the limit of the largest virtual machine instance possible, you must make a copy or clone of the instance to support additional users. A group of users within a particular instance is called a pod. Pods are managed by a Cloud Control System (CCS). In AWS, the CCS is the AWS Management Console. Sizing limitations for pods need to be accounted for if you are building a large cloud-based application. Pods are aggregated into pools within an IaaS region or site called an availability zone. In very large cloud computing networks, when systems fail, they fail on a pod-by-pod basis, and often on a zone-by-zone basis. A failover system between zones gives IaaS private clouds a very high degree of availability. Figure shows how pods are aggregated and virtualized in IaaS across zones.

When a cloud computing infrastructure isolates user clouds from each other so the management system is incapable of interoperating with other private clouds, it creates an information silo, or simply a silo. Most often, the term silo is applied to PaaS offerings such as Force.com or Quick Base, but silos often are an expression of the manner in which a cloud computing infrastructure is architected. Silos are the cloud computing equivalent of compute islands: They are processing domains that are sealed

off from the outside. When you create a private virtual network within an IaaS framework, the chances are high that you are creating a silo. Silos impose restrictions on interoperability that runs counter to the open nature of build-componentized service-oriented applications. However, that is not always a bad thing. A silo can be its own ecosystem; it can be protected and secured in ways that an open system can't be. Silos just aren't as flexible as open systems and are subject to vendor lock-in. Pods, aggregation, and failover in IaaS.

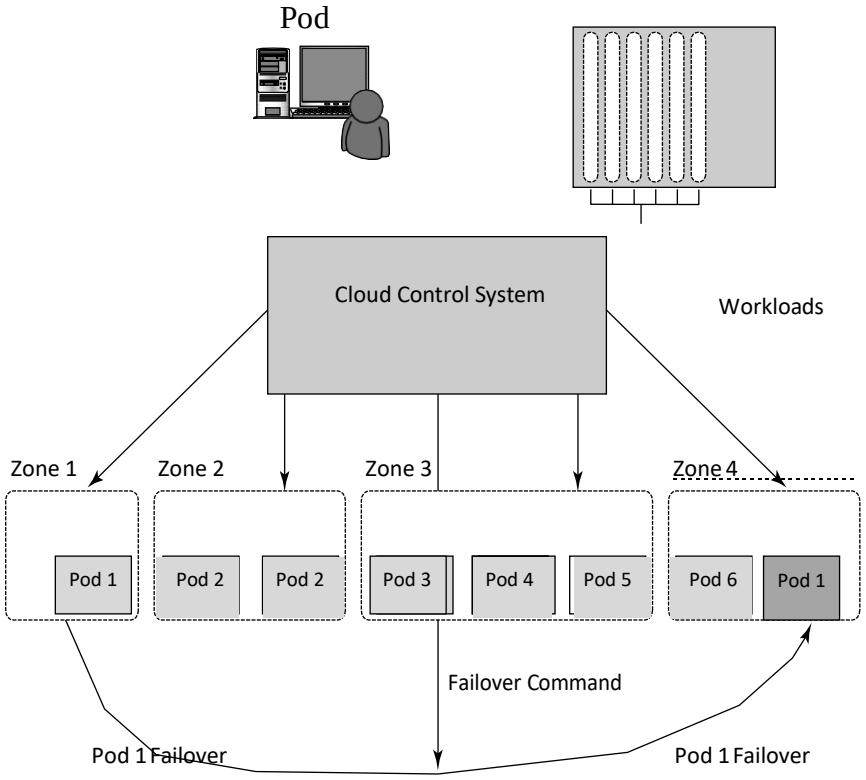

Figure 1.4 Defining Platform As A Service (Paas)

The Platform as a Service model describes a software environment in which a developer can create customized solutions within the context of the development tools that the platform provides. Platforms can be based on specific types of development languages, application frameworks, or other constructs. A PaaS offering provides the tools and development environment to deploy applications on another vendor's application. Often a PaaS tool is a fully integrated development

environment; that is, all the tools and services are part of the PaaS service. To be useful as a cloud computing offering, PaaS systems must offer a way to create user interfaces, and thus support standards such as HTLM, JavaScript, or other rich media technologies.

In a PaaS model, customers may interact with the software to enter and retrieve data, perform actions, get results, and to the degree that the vendor allows it, customize the platform involved. The customer takes no responsibility for maintaining the hardware, the software, or the development of the applications and is responsible only for his interaction with the platform. The vendor is responsible for all the operational aspects of the service, for maintenance, and for managing the product(s) lifecycle.

The one example that is most quoted as a PaaS offering is Google's App Engine platform, which is described in more detail in Chapter 8. Developers program against the App Engine using Google's published APIs. The tools for working within the development framework, as well as the structure of the file system and data stores, are defined by Google. Another example of a PaaS offering is Force.com; Salesforce.com's developer platform for its SaaS offerings, Force.com is an example of an add-on development environment.

A developer might write an application in a programming language like Python using the Google API. The vendor of the PaaS solution is in most cases the developer, who is offering a complete solution to the customer. Google itself also serves as a PaaS vendor within this system, because it offers many of its Web service applications to customers as part of this service model. You can think of Google Maps, Google Earth, Gmail, and the myriad of other PaaS offerings as conforming to the PaaS service model, although these applications themselves are offered to customers under what is more aptly described as the Software as a Service (SaaS) model that is described. The difficulty with PaaS is that it locks the developer (and the customer) into a solution that is dependent upon the platform vendor. An application written in Python against Google's API using the Google App Engine is likely to work only in that environment. There is considerable vendor lock-in associated with a PaaS solution.

Defining Software as a Service (SaaS)

The most complete cloud computing service model is one in which the computing hardware and software, as well as the solution itself, are provided by a vendor as a complete service offering. It is referred to as the Software as a Service (SaaS) model. SaaS provides the complete infrastructure, software, and solution stack as the service offering. A good way to think about SaaS is that it is the cloud-based equivalent of shrink-wrapped software.

Software as a Service (SaaS) may be succinctly described as software that is deployed on a hosted service and can be accessed globally over the Internet, most often in a browser. With the exception of the user interaction with the software, all other aspects of the service are abstracted away. Every computer user is familiar with SaaS systems, which are either replacements or substitutes for locally installed software. Examples of SaaS software for end-users are Google Gmail and Calendar, QuickBooks online, Zoho Office Suite, and others that are equally well known. SaaS applications come in all shapes and sizes, and include custom software such as billing and invoicing systems, Customer Relationship Management (CRM) applications, Help Desk applications, Human Resource (HR) solutions, as well as myriad online versions of familiar applications.

Many people believe that SaaS software is not customizable, and in many SaaS applications this is indeed the case. For user-centric applications such as an office suite, that is mostly true; those suites allow you to set only options or preferences. However, many other SaaS solutions expose Application Programming Interfaces (API) to developers to allow them to create custom composite applications. These APIs may alter the security model used, the data schema, workflow characteristics, and other fundamental features of the service's expression as experienced by the user. Examples of a SaaS platform with an exposed API are Salesforce.com and Quicken.com. So SaaS does not necessarily mean that the software is static or monolithic.

SaaS characteristics

All Software as a Service (SaaS) applications share the following characteristics:

1. The software is available over the Internet globally through a browser on demand.

2. The typical license is subscription-based or usage-based and is billed on a recurring basis. In a small number of cases a flat fee may be changed, often coupled with a maintenance fee.

3. The software and the service are monitored and maintained by the vendor, regardless of where all the different software components are running. There may be executable client-side code, but the user isn't responsible for maintaining that code or its interaction with the service.

4. Reduced distribution and maintenance costs and minimal end-user system costs generally make SaaS applications cheaper to use than their shrink-wrapped versions.

5. Such applications feature automated upgrades, updates, and patch management and much faster rollout of changes.

6. SaaS applications often have a much lower barrier to entry than their locally installed competitors, a known recurring cost, and they scale on demand (a property of cloud computing in general).

7. All users have the same version of the software so each user's software is compatible with another's.

8. SaaS supports multiple users and provides a shared data model through a single-instance; multi-tenancy model. The alternative of software virtualization of individual instances also exists, but is less common.

Open SaaS and SOA

A considerable amount of SaaS software is based on open source software. When open source soft- ware is used in a SaaS, you may hear it referred to as *Open SAAS*. The advantages of using open source software are that systems are much cheaper to deploy because you don't have to purchase the operating system or software, there is less vendor lock-in, and applications are more portable. The popularity of open source software, from Linux to APACHE, MySql, and Perl (the LAMP platform) on the Internet, and the number of people who are trained in open source software make Open SaaS an attractive proposition. The impact of Open SaaS will likely translate into better profitability for the companies that deploy open source software in the cloud, resulting in lower development costs and more robust solutions.

Cloud Types

To discuss cloud computing intelligently, you need to define the lexicon of cloud computing; many acronyms in this area probably won't survive long. Most people separate cloud computing into two distinct sets of models:

- **Deployment models:** This refers to the location and management of the cloud's infrastructure.
- **Service models:** This consists of the particular types of services that you can access on a cloud computing platform. This is a very useful demarcation that is now widely accepted.

The NIST model

The United States government is a major consumer of computer services and, therefore, one of the major users of cloud computing networks. The U.S. National Institute of Standards and Technology (NIST) has a set of working definitions that separate cloud computing into service models and deployment models. Those models and their relationship to essential characteristics of cloud computing are shown in Figure 1.1.

The NIST model originally did not require a cloud to use virtualization to pool resources, nor did it absolutely require that a cloud support multi-tenancy in the earliest definitions of cloud computing. Multi-tenancy is the sharing of resources among two or more clients. The latest version of the NIST definition does require that cloud computing networks use virtualization and support multi-tenancy.

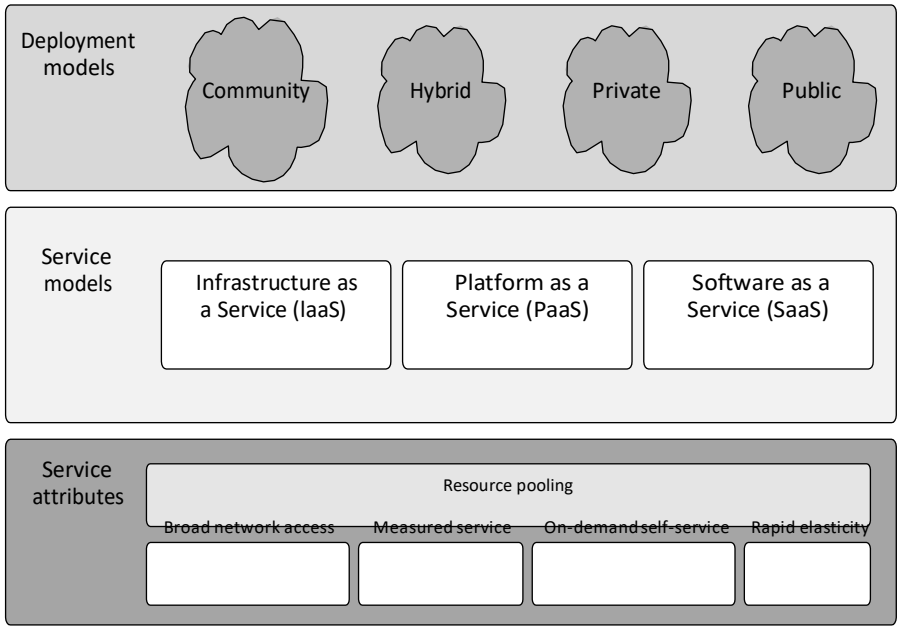

Figure 1.5 SaaS implementation based on SOA

Because cloud computing is moving toward a set of modular interacting components based on standards such as the Service Oriented Architecture you might expect that future versions of the NIST model may add those features as well. The NIST cloud model doesn't address a number of intermediary services such as transaction or service brokers, provisioning, integration, and interoperability services that form the basis for many cloud computing discussions. Given the emerging roles of service buses, brokers, and cloud APIs at various levels, undoubtedly these elements need to be added to capture the whole story.

The CLOUD CUBE Model

The Open Group maintains an association called the Jericho Forum whose main focus is how to protect cloud networks. The group has an interesting model that attempts to categorize a cloud network based on four dimensional factors. As described in its paper called "Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration the type of cloud networks you use dramatically changes the notion of where the boundary between the client's network and the cloud begins and ends. The four dimensions of the Cloud Cube Model are shown in Figure 1.2 and listed here:

- **Physical location of the data:** Internal (I) / External (E) determines your organization's boundaries.
- **Ownership:** Proprietary (P) / Open (O) is a measure of not only the technology ownership, but of interoperability, ease of data transfer, and degree of vendor application lock-in.
- **Security boundary:** Perimeterised (Per) / De-perimiterised (D-p) is a measure of whether the operation is inside or outside the security boundary or network firewall.
- **Sourcing**: In sourced or Outsourced means whether the service is provided by the customer or the service provider.

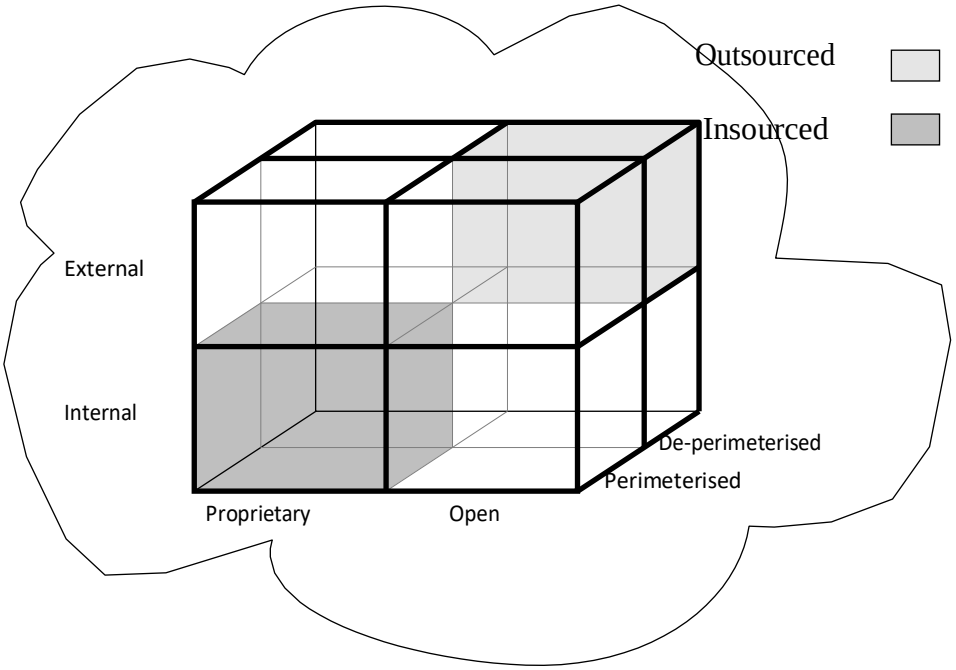

Figure 1.6 Cloud Cube Model

Taken together, the fourth dimension corresponds to two different states in the eight possible cloud forms: Per (IP, IO, EP, EO) and D-p (IP, IO, EP, EO). The sourcing dimension addresses the deliverer of the service. What the Cloud Cube Model is meant to show is that the traditional notion of a network boundary being the network's firewall no longer applies in cloud computing.

Deployment models

A deployment model defines the purpose of the cloud and the nature of how the cloud is located. The NIST definition for the four deployment models is as follows:

- **Public cloud:** The public cloud infrastructure is available for public use alternatively for a large industry group and is owned by an organization selling cloud services.
- **Private cloud:** The private cloud infrastructure is operated for the exclusive use of an organization. The cloud may be managed by that organization or a third party. Private clouds may be either on- or off-premises.

- **Hybrid cloud:** A hybrid cloud combines multiple clouds (private, community of public) where those clouds retain their unique identities, but are bound together as a unit. A hybrid cloud may offer standardized or proprietary access to data and applications, as well as application portability.
- **Community cloud:** A community cloud is one where the cloud has been organized to serve a common function or purpose. It may be for one organization or for several organizations, but they share common concerns such as their mission, policies, security, regulatory compliance needs, and so on. A community cloud may be managed by the constituent organization(s) or by a third party.

Service models

In the deployment model, different cloud types are an expression of the manner in which infra- structure is deployed. You can think of the cloud as the boundary between where a client's net- work, management, and responsibilities ends and the cloud service provider's begins. As cloud computing has developed, different vendors offer clouds that have different services associated with them. The portfolio of services offered adds another set of definitions called the service model.

There are many different service models described in the literature, all of which take the following form:

XaaS, or "<Something> as a Service"

Three service types have been universally accepted:

Infrastructure as a Service: IaaS provides virtual machines, virtual storage, virtual infrastructure, and other hardware assets as resources that clients can provision.The IaaS service provider manages all the infrastructure, while the client is responsible for all other aspects of the deployment. This can include the operating system, applications, and user interactions with the system.

Platform as a Service: PaaS provides virtual machines, operating systems, applications, services, development frameworks, transactions, and control structures.

The client can deploy its applications on the cloud infrastructure or use applications that Were programmed using languages and tools that are supported by the PaaS service provider. The service provider manages the cloud infrastructure, the operating systems, and the enabling software. The client is responsible for installing and managing the application that it is deploying.

Software as a Service: SaaS is a complete operating environment with applications,

management, and the user interface. In the SaaS model, the application is provided to the client through a thin client interface (a browser, usually), and the customer's responsibility begins and ends with entering and managing its data and user interaction. Everything from the application down to the infrastructure is the vendor's responsibility.

The three different service models taken together have come to be known as the SPI model of cloud computing. Many other service models have been mentioned: SaaS, Storage as a Service; IdaaS, Identity as a Service; CaaS, Compliance as a Service; and so forth. However, the SPI services encompass all the other possibilities. At the bottom of the stack is the hardware or infrastructure that comprises the network. As you move upward in the stack, each service model inherits the capabilities of the service model beneath it. IaaS has the least levels of integrated functionality and the lowest levels of integration, and SaaS has the most.

Examples of IaaS service providers include:

- Amazon Elastic Compute Cloud (EC2)
- Eucalyptus
- GoGrid
- FlexiScale
- Linode
- RackSpace Cloud
- Terremark

All these vendors offer direct access to hardware resources. On Amazon EC2, considered the classic IaaS example, a client would provision a computer in the form of a virtual machine image, provision storage, and then go on to install the operating system and applications onto that virtual system. Amazon has a number of operating systems and some enterprise applications that they offer on a rental basis to customers in the form of a number of canned images, but customers are free to install whatever software they want to run. Amazon's responsibilities as expressed in its Service Level Agreement, which is published on Amazon's Web site, contractually obligates Amazon to provide a level of performance commensurate with the type of resource chosen, as well as a certain level of reliability as measured by the system's uptime.

A PaaS service adds integration features, middleware, and other orchestration and choreography services to the IaaS model. Examples of PaaS services are:

- Force.com
- GoGrid CloudCenter
- Google AppEngine
- Windows Azure Platform

When a cloud computing vendor offers software running in the cloud with use of the application on a pay-as-you-go model, it is referred to as SaaS. With SaaS, the customer uses the application as needed and is not responsible for the installation of the application, its maintenance, or its upkeep. A good example of an SaaS offering is an online accounting package, with the online versions of Quicken and Quickbooks a prime example. Figure 1.6 shows a home page for QuickBooks Online plus on the Intuit.com Web site

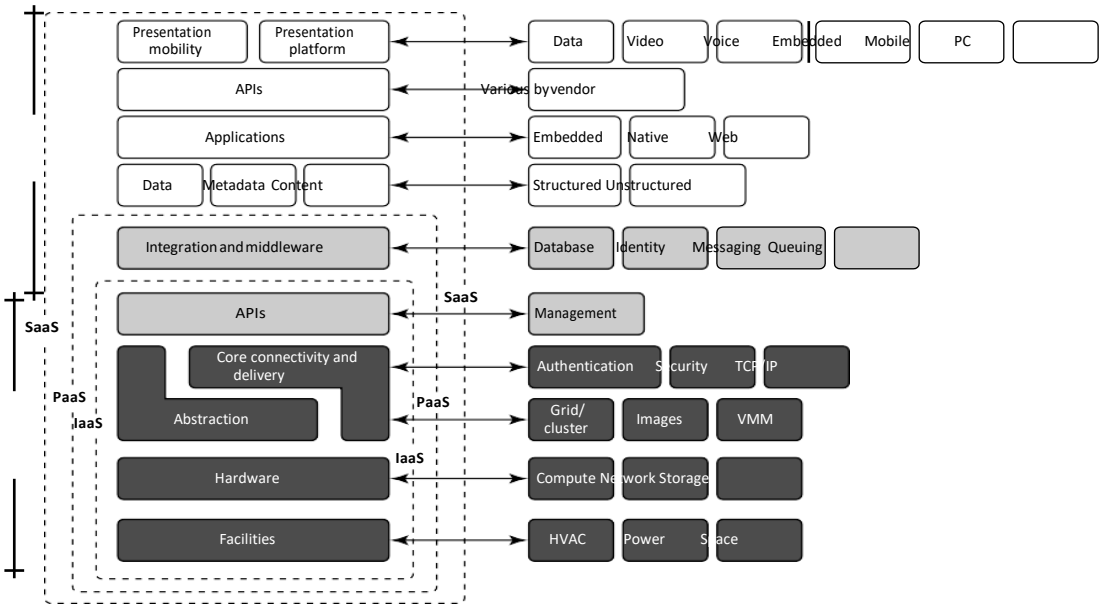

Figure 1.7 The Cloud Reference Model

A client using an SaaS service might—as is the case for QuickBooks online—log into the service from his browser, create an account, and enter data into the system. Intuit.com has a service agreement that not only covers the performance of the hardware and software, but extends to protecting the data that they store for clients, and other fundamental characteristics.

Other good examples of SaaS cloud service providers are:

- GoogleApps

- Oracle On Demand
- SalesForce.com
- SQL Azure

These service model classifications start to get confusing rather quickly when you have a cloud service provider that starts out offering services in one area and then develops services that are classified as another type. For example, SalesForce.com started out as a Customer Relationship Management SaaS platform that allowed clients to add their own applications. Over time SalesForce.com opened an API called the Force API that allowed developers to create applications based on the SalesForce.com technologies. Force.com is thus their PaaS service.

Grid Computing:

Grid computing is a combination of resources from multiple administrative domains to reach a common target, and this group of computers can distributed on several location and each a group of grids can be connected to each other. The need of access to additional resources and the collaborating between organizations leads to the need for grid computing. Grid environments are extremely well suited to run jobs that can be split into smaller chunks and run concurrently on many nodes.

- The grid is a loosely coupled systems and from its characteristics that it's a distributed Job Management and scheduling, the computers in the grid are not required to be in the same physical location and can be operated independently, so each computer on the grid is concerned a distinct computer, the computers in the grid are not tied to only on operating system and can run different OSs and different hardware, when it comes to a large project, the grid divides it to multiple computers to easily use their resources.
- The grid is multiple owned, it could be owned by several companies. Interconnection network is mostly internet with high latency and low bandwidth. The security in the grid is public and private based on authentication and mapping user to an account. And it has limited support privacy, its capacity is not stable, it varies, but it's high.
- The self healing in the cluster is Limited, it's often restarts the failed tasks and applications. its service negotiations are based on service level agreements, and the user management is decentralized. The grid computing is usually used in predictive modeling and simulations, engineering design and automation, energy resources exploration, medical, military, basic Research and visualization.

Architecture:

Fabric layer to provide the resources which shared access is mediated by grid

computing. Connectivity layer and it means the core communication and authentication protocols required for grid specific network functions. Resource layer and it defines the protocols, APIs and SDK for secure negotiations, imitations, monitoring control, accounting and payment of sharing operations on individual resources. Collective layer which it contains protocols and services that capture interactions among a collection of resources Finally the Application layer, and it's user applications that operate within VO environment.

Advantages: One of the advantages of grid computing that you don't need to buy large servers for applications that can be split up and farmed out to smaller commodity type servers; secondly it's more efficient in use of resources. Also the grid environments are much more modular and don't have much points of failure. About policies in the grid it can be managed by the grid software, beside that upgrading can be done without scheduling downtime, and jobs can be executed in parallel speeding performance

Disadvantages: It needs fast interconnect between computers resources, and some applications may need to be pushed to take full advantage of the new model, and licensing across many servers may make it forbidden for some applications, and the grid environments include many smaller servers across various administrative domains. also political challenges associated with sharing resources especially across different admin domains.

Utility Computing:

Utility Computing refers to a type of computing technologies and business models which provide services and computing resources to the customers, such as storage, applications and computing power. This model has the advantage of a low cost with no initial price to reach the computer resources. This repackaging of computing services is the foundation of the shift to on demand computing, software as a service and cloud computing models which late developed the idea of computing, applications and network as a service. Utility computing is kind of virtualization, that means the whole web storage space and computing power which it's available to users is much larger than the single time-sharing computer. Multiple backend web servers used to make this kind of web service possible. Utility computing is similar to cloud computing and it often requires a cloud-like infrastructure.

Architecture of utility computing:

It depends on the devices of the users and the providing company for the utility service and similar to the architecture of the cloud.

Advantages of utility computing: Here are some benefits of utility computing such as that the client doesn't have to buy all the hardware, software and licenses needed to do business. Instead, the client relies on another party to provide these services. It also gives companies the option to subscribe to a single service and use the same suite of software throughout the entire client organization. And it offers compatibility of all the computers in large companies.

Disadvantages of utility computing:

There's some issues are considered as disadvantages such as reliability which means the service could be stopped from the utility computing company for any reason such as a financial trouble or equipment problems. Also utility computing systems can also be attractive targets for hackers, and much of the responsibility of keeping the system safe falls to the provider

Cloud computing at Amazon: Amazon introduced a computing platform that has changed the face of computing in the last decade.

- First, it installed a powerful computing infrastructure to sustain its core business, e-commerce, selling a variety of goods ranging from books and CDs to gourmet foods and home appliances.
- Then Amazon discovered that this infrastructure could be further extended to provide affordable and easy-to-use resources for enterprise computing as well as computing for the masses. In mid-2000 Amazon introduced Amazon Web Services (AWS), based on the IaaS delivery model.
- In this model the cloud service provider offers an infrastructure consisting of compute and storage servers interconnected by high-speed networks that support a set of services to access these resources.
- An application developer is responsible for installing applications on a platform of his or her choice and managing the resources provided by Amazon

Elastic Compute Cloud (EC2)1 is a Web service with a simple interface for launching instances of an application under several operating systems, such as several Linux distributions, Microsoft Windows Server 2003 and 2008, Open Solaris, FreeBSD, and NetBSD.

- An instance is created either from a predefined Amazon Machine Image (AMI) digitally signed and stored in S3 or from a user-defined image. The image includes the operating system, the run-time environment, the libraries, and the application desired by the user.
- AMI images create an exact copy of the original image but without configuration-dependent information such as the hostname or the MAC address.

A user can:

(i) Launch an instance from an existing AMI and terminate an instance;
(ii) Start and stop an instance;
(iii) Create a new image;
(iv) Add tags to identify an image; and
(v) Reboot an instance.

EC2 is based on the Xen virtualization strategy. In EC2 each virtual machine or instance functions as a virtual private server. An instance specifies the maximum amount of resources available to an application, the interface for that instance, and the cost per hour. A user can interact with EC2 using a set of SOAP messages and can list available AMI images, boot an instance from an image, terminate an image, display the running instances of a user, display console output, and so on. The user has root access to each instance in the elastic and secure computing environment of EC2. The instances can be placed in multiple locations in different regions and availability zones. EC2 allows the import of virtual machine images from the user environment to an instance through a facility called VM import.

EC2 associates an elastic IP address with an account; this mechanism allows a user to mask the failure of an instance and remap a public IP address to any instance of the account without the need to interact with the software support team. Simple Storage System (S3) is a storage service designed to store large objects. It supports a minimal set of functions: write, read, and delete. S3 allows an application to handle an unlimited number of objects ranging in size from one byte to five terabytes. An object is stored in a bucket and retrieved via a unique developer-assigned key. A bucket can be stored in a region selected by the user. S3 maintains the name, modification time, an access control list, and up to four kilobytes of user-defined metadata for each object.

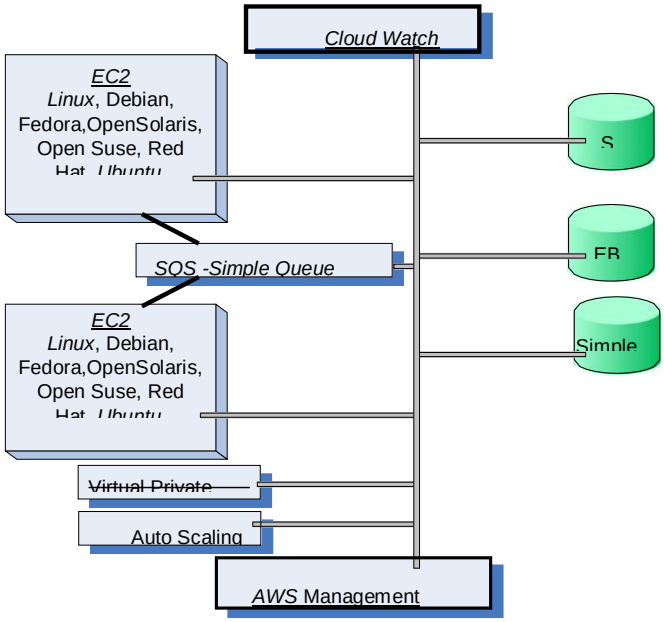

Figure 1.8 Services offered by AWS

Services offered by AWS are accessible from the AWS Management Console. Applications running under a variety of operating systems can be launched using EC2. Multiple EC2 instances can communicate using SQS. Several storage services are available: S3, Simple DB, and EBS. The Cloud Watch supports performance monitoring; the Auto Scaling supports elastic resource management. The Virtual PRIVATE Cloud allows direct migration of parallel applications

- S3 computes the MD52 of every object written and returns it in a field called ETag.
- A user is expected to compute the MD5 of an object stored or written and compare this with the ETag; if the two values do not match, then the object was corrupted during transmission or storage.
- The Amazon S3 SLA guarantees reliability. S3 uses standards-based REST and SOAP interfaces, the default download protocol is HTTP, but BitTorrent3 protocol interface is also provided to lower costs for high-scale distribution.
- Elastic Block Store (EBS) provides persistent block-level storage volumes for use with Amazon EC2 instances.

A volume appears to an application as a raw, unformatted, and reliable physical disk; the size of the storage volumes ranges from one gigabyte to one terabyte. The volumes are grouped together in availability zones and are automatically replicated in each zone. An EC2 instance may mount multiple volumes, but a volume cannot be shared among multiple instances. The EBS supports the creation of snapshots of the volumes attached to an instance and then uses them to restart an instance. The storage strategy provided by EBS is suitable for database applications, file systems, and applications using raw data devices. Simple DB is a non relational data store that allows developers to store and query data items via Web services requests. It supports store-and-query functions traditionally provided only by relational databases.

Simple DB creates multiple geographically distributed copies of each data item and supports high-performance Web applications; at the same time, it automatically manages infrastructure provisioning, hardware and software maintenance, replication and indexing of data items, and performance tuning. Simple Queue Service (SQS) is a hosted message queue.

- SQS is a system for supporting automated workflows; it allows multiple Amazon EC2 instances to coordinate their activities by sending and receiving SQS messages. Any computer connected to the Internet can add or read messages without any installed software or special firewall configurations.
- Applications using SQS can run independently and asynchronously and do not

need to be developed with the same technologies.

- A received message is "locked" during processing; if processing fails, the lock expires and the message is available again. The time-out for locking can be changed dynamically via the Change Message Visibility operation.
- Developers can access SQS through standards-based SOAP and Query interfaces. Queues can be shared with other AWS accounts and anonymously; queue sharing can also be restricted by IP address and time-of-day.

Cloud Watch is a monitoring infrastructure used by application developers, users, and system administrators to collect and track metrics important for optimizing the performance of applications and for increasing the efficiency of resource utilization. Without installing any software, a user can monitor approximately a dozen preselected metrics and then view graphs and statistics for these metrics. When launching an Amazon Machine Image (AMI), a user can start the Cloud Watch and specify the type of monitoring. Basic Monitoring is free of charge and collects data at five-minute intervals for up to 10 metrics; Detailed Monitoring is subject to a charge and collects data at one-minute intervals.

This service can also be used to monitor the latency of access to EBS volumes, the available storage space for RDS DB instances, the number of messages in SQS, and other parameters of interest for applications. Virtual Private Cloud (VPC) provides a bridge between the existing IT infrastructure of an organization and the AWS cloud. The existing infrastructure is connected via a virtual private network (VPN) to a set of isolated AWS compute resources. VPC allows existing management capabilities such as security services, firewalls, and intrusion detection systems to operate seamlessly within the cloud. Auto Scaling exploits cloud elasticity and provides automatic scaling of EC2 instances.

The service supports grouping of instances, monitoring of the instances in a group, and defining triggers and pairs of Cloud Watch alarms and policies, which allow the size of the group to be scaled up or down. Typically, a maximum, a minimum, and a regular size for the group are specified. An Auto Scaling group consists of a set of instances described in a static fashion by launch configurations. When the group scales up, new instances are started using the parameters for the run Instances EC2 call provided by the launch configuration. When the group scales down, the instances with older launch configurations are terminated first. The monitoring function of the Auto Scaling service carries out health checks to enforce the specified policies; for example, a user may specify a health check for elastic load balancing and then Auto Scaling will terminate an instance exhibiting a low performance and start a new one.

Two new services, the Elastic Beanstalk and the Cloud Formation. Some of the

management functions provided by the service are:

- deployment of a new application version (or rollback to a previous version)
- Access to the results reported by CloudWatch monitoring service
- email notifications when application status changes or application servers are added or removed
- access to server login files without needing to login to the application servers.

CloudFormation allows the creation of a stack describing the infrastructure for an application. The user creates a template, a text file formatted as in Javascript Object Notation (JSON), describing the resources, the configuration values, and the interconnection among these resources. The template can be parameterized to allow customization at run time, (e.g., to specify the types of instances, database port numbers, or RDS size). A template for the creation of an EC2 instance follows:

```
{
"Description" : "Create instance running Ubuntu Server 12.04 LTS
64 bit AMI
"Parameters" : {
"KeyPair" : {
"Description" : "Key Pair to allow SSH access to the instance",
"Type" : "String"
}
},
"Resources" : {
"Ec2Instance" : {
"Type" : "AWS::EC2::Instance",
"Properties" : {
"KeyName" : {"Ref" : "KeyPair"},
"ImageId" : "aki-004ec330"
}
}
},
"Outputs" : {
"InstanceId" : {
"Description" : "The InstanceId of the newly created
instance",
"Value" : { "Ref" : "Ec2InstDCM"}
}
},
"AWSTemplateFormatVersion" : "2012-03-09"
}
```

The Amazon Web Services Licensing Agreement (AWSLA) allows the cloud service provider to terminate service to any customer at any time for any reason and contains a covenant not to sue Amazon or its affiliates for any damages that might arise out of the use of AWS. The AWSLA prohibits the use of "other information obtained through AWS for the purpose of direct marketing, spamming, contacting sellers or customers." It prohibits AWS from being used to store any content

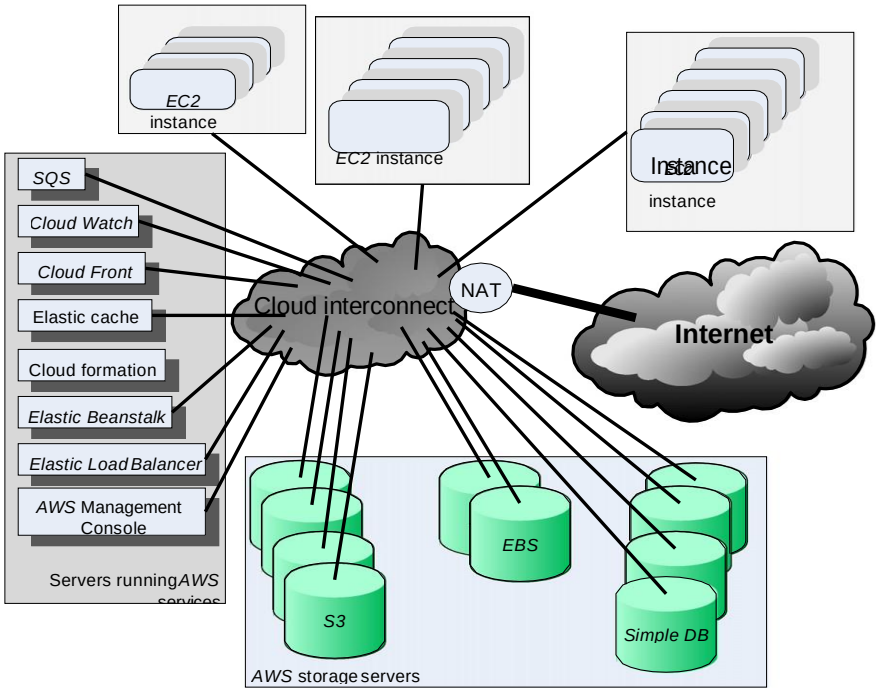

Figure 1.9 The AWS

The configuration of an availability zone supporting *AWS* services. A cloud interconnect supports high-speed communication among compute and storage servers in the zone. It also supports communication with servers in other availablity zones and with cloud users via a Network Address Translation (NAT). NAT maps external IP addresses to internal ones. Multitenancy increases server utilization and lowers costs.

Cloud computing: The Google perspective Google's effort is concentrated in the area of Software-as-a-Service (SaaS). The data for these services is stored in data centers on the cloud. The Gmail service hosts emails on Google servers and, provides a Web interface to access them and tools for migrating from Lotus Notes and Microsoft Exchange. Google Docs is Web-based software for building text documents,

spreadsheets, and presentations. It supports features such as tables, bullet points, basic fonts, and text size; it allows multiple users to edit and update the same document and view the history of document changes; and it provides a spell checker. The service allows users to import and export files in several formats, including Microsoft Office,

Few users are aware of this service.

- Google Base is accessed in response to keyword queries posed on Google.com, provided that there is relevant data in the database. To fully integrate Google Base, the results should be ranked across properties.
- In addition, the service needs to propose appropriate refinements with candidate values in select menus; this is done by computing histograms on attributes and their values during query time.
- Specialized structure-aware search engines for several interest areas, including travel, weather, and local services, have already been implemented. However, the data available on the Web covers a wealth of human knowledge; it is not feasible to define all the possible domains and it is nearly impossible to decide where one domain ends and another begins.
- Google has also redefined the laptop with the introduction of the Chrome book, a purely Web-centric device running Chrome OS. Cloud-based applications, extreme portability, built-in 3G connectivity, almost instant-on, and all-day battery life are the main attractions of this device with a keyboard.

Google adheres to a bottom-up, engineer-driven, liberal licensing and user application development philosophy. Google as well as the other cloud service providers must manage vast amounts of data. In a world where users would most likely desire to use multiple cloud services from independent providers, the question of whether the traditional data base management services (DBMSs) are sufficient to ensure interoperability comes to mind. A DBMS efficiently supports data manipulations and query processing but operates in a single administrative domain and uses well-defined schema. The interoperability of data management services requires semantic integration of services based on different schemas. An answer to the limitations of traditional DBMS is the so-called data spaces introduced in data spaces do not aim at data integration but rather at data coexistence

Microsoft Windows Azure and online services Azure and Online Services are, respectively, PaaS and SaaS cloud platforms from Microsoft. Windows Azure is an operating system, SQL Azure is a cloud-based version of the SQL Server, and Azure AppFabric (formerly .NET Services) is a collection of services for cloud applications. Windows Azure has three core components Compute, which provides a computation

environment; Storage for scalable storage; and Fabric Controller, which deploys, manages, and monitors applications; it interconnects nodes consisting of servers, high-speed connections, and switches

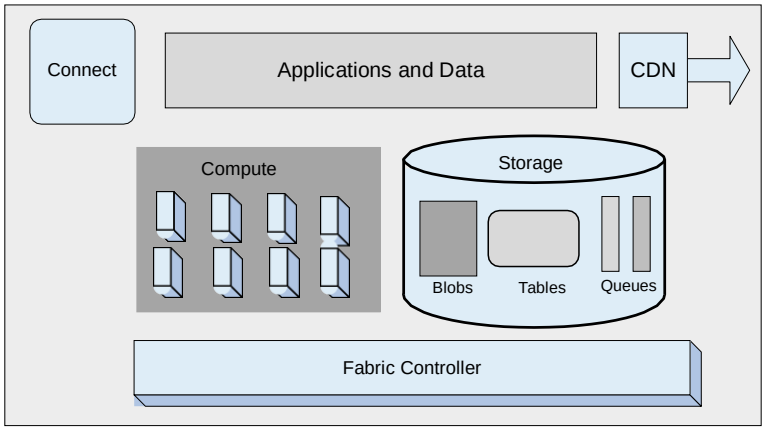

Figure 1.10 Microsoft Windows Azure and online services

The components of Windows Azure: Compute, which runs cloud applications; Storage, which uses blobs, tables, and queues to store data; Fabric Controller, which deploys, manages, and monitors applications; CDN, which maintains cache copies of data; and Connect, which allows IP connections between the user systems and applications running on Windows Azure.

Cloud based services and applications:
Health care

In order to understand Cloud Computing in the healthcare industry, we must understand the basics of cloud computing in general. Fig 1, gives overview of Cloud computing There are different definition available of cloud computing, one of them is as "Cloud computing is a new way of delivering computing resources and services. There are many researcher and experts believe that it can improve health care services, benefit of health care research, and change the face of health information technology." However, as with new innovation type, cloud computing should be more rigorously evaluated before its widespread adoption.

Cloud-based services are steadily becoming very large adopted by healthcare organizations. The accelerating migration to cloud computing represents a change for the way the healthcare industry sources its information technology, Different architecture is available for Cloud Computing; we can broadly divide the cloud

architecture into two following parts:

1. Front End

2. Back End

Energy systems

Electrical power has become an indispensable part of modern day life. Hebra and NIST styled today's electric power system as a multifaceted system of power generation, transmission, and distribution. With the global economy more reliant on sustainable development of energy, a series of problems, such as energy shortage, electricity blackout and global warming are gaining attention. ABB pointed out that there are tenacious economic as well as environmental urgings for the refurbishment of the conventional power systems, and its replacement with a Smart Electrical Power Grid or simply Smart Grid.

A smart grid is an electricity network that uses digital and other advanced technologies to monitor and manage the transport of electricity from all generation sources to meet the varying electricity demands of end-users. Energy demand from the users changes dynamically in different time-periods. The existing power grids need optimal balancing of electricity demand and supply between the customers and the utility providers. To address these requirements in smart grid, the energy management systems (EMS) such as building energy management (BEMS), demand side management (DSM) and home energy management (HEM) are integrated. A smart grid allows various renewable energy sources to have an efficient management of supply and demand.

Communication network plays an important role for reliable energy management as smart grid is the combination of electrical and communication network. Cloud computing techniques proposed in the existing literature for energy management in smart grid are discussed briefly. A model of power dispatching in smart grid with cloud is described. In short, our objective in this paper is to offer the following: A comprehensive overview of smart grid and cloud computing in terms of energy management.

- A highlight on cost effective cloud based power dispatching in smart grid
- The rest of the paper is organized as follows.

Transportation:

IBM introduces urban traffic management system in the year of 1956. Today, transportation research and development is no longer a field dominated by civil, mechanical, operations research, and other traditional engineering and management disciplines. Rather, computer sciences, control, communication, the Internet, and methods developed in artificial intelligence (AI), computational intelligence, web sciences, and many other emerging information sciences and engineering areas have formed the core of new ITS technology and become integral and important parts of modern transportation engineering. Cloud computing control the traffic allocation process provides optimal solution with five stages specification. In the first phase, computers were huge and costly, so mainframes were usually shared by many terminals.

INTELLIGENT TRAFFIC CLOUDS

Urban-traffic management systems using intelligent traffic clouds have overcome the issues we've described. With the support of cloud computing technologies, it will go far beyond than any other multi agent traffic management systems, addressing issues such as infinite system scalability, an appropriate agent management scheme, reducing the upfront investment and risk for users, and minimizing the total cost of ownership.

Manufacturing

The Business Benefits of Cloud in manufacturing another important question addressed in the survey was how companies were justifying their investment. By far the biggest factor for manufacturers, which is consistent with overall results, is to save on hardware spend. Although hardware cost reduction is the dominant factor, the data shows the percentages for what was top ranked, and the differences in the secondary factors provide an interesting perspective on the manufacturing industry. For example, the second-most-frequent choice as the top business benefit was reducing software license costs, which switches positions with IT staff productivity in the overall sample

Education

Students are demanding more technology services from their schools. It's important not only to keep pace with their evolving needs, but also to prepare them for the demands of the workplace tomorrow. At the same time, education institutions are under increasing pressure to deliver more for less, and they need to find ways to offer rich, affordable services and tools. Those educators who can deliver these sophisticated communications environments, including the desktop applications that employers use today, will be helping their students find better jobs and greater opportunities in the future.

Cloud options range from everyday services, such as e-mail, calendaring, and collaboration tools that members of your education community can use to collaborate online, to infrastructure services that free IT operations from mundane tasks and help you build on the investments you already have in place. System administrators can bring new services and computing capacity online quickly while managing costs as operational expenses.

Government

Over the past 10 years Internet and Web-based services have grown rapidly and has been used by many companies. However, the cost of data storage and the power consumption by the hardware is increased. At the same time major companies started extensive studies to reduce costs, better utilizing of existing resources and also to support their own business. In these studies, they found a new solution to answer their challenges, to use and to get maximum benefit from the resources and it was nothing but cloud computing. This new technology is what which can answer thousands of their hardware and software needs.

- Cloud Computing was introduced in other styles, such as Grid computing and service-oriented architecture which goal of these styles is processing large quantities of data using clusters of computers. Such high-volume computational problems can be solved easily and in an appropriate manner with the expansion and evolution of cloud computing.
- The other benefits of cloud computing in e-government should not be ignored of course. Which cost Reduction, integration and reusability of services can be noted,
- Due to the cloud computing novelty, in order to identify cloud computing benefits and weaknesses, it is necessary that this technology get completely identified for the development and use of it in e-government architecture, and its different domains should be considered much as possible to be used in the e-government and should've tried to overcome its shortcomings.

Cloud computing have various definitions which some have been brought here. The definition of the national institute of standard and technology of America is as follows "Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." On other common & acceptable definition is of Mater et al. "A very exact scalable instrument, capable of technology-enabled service, which is available easily on the internet when needed." Following the

definition of cloud computing, we should comprehend their important features, developed models, the way of using services and also the way of protecting it, in order to know well and accept it . Here are the five key features of cloud computing.

- Service demand on self. Using this feature when needed the customer can easily and automatically access to computing facilities like server, net, storage and soon from any provider.

- Ubiquitous network access. It implies that the facilities are accessible on the net and they can be used following standard methods. The methods which support weak and strong clients like laptop and mobile phones.

- Location-independent resource pooling. This features pools different customers needed resources in the same place dynamically by the providers. These resources can include the storage, memory, the bandwidth of net and virtual machines.

- Rapid elasticity. Using this feature, the facilities can be provided rapidly and with high elasticity and can be expanded or release fast. In other words the services can always be updated and improved and accessible for the users.

- Measured service. This feature enables monitoring, control and reporting of the resources, and can apparently control and report the amount and quantity of resource using for both customer and the provider of the infrastructure. In other words all these features cover the coherence and appearance of the clouds.

Mobile communication:

Cloud computing for mobile world or, rather, Mobile Cloud Computing (MCC) is a well accepted concept that aims at using cloud computing techniques for storage and processing of data on mobile devices, thereby reducing their limitations. This underlines the importance of cloud computing for mobile. The end mobile device user will eventually be the benefactor of the Mobile Cloud Computing. Company users can share resources and applications without a high level of capital expenditure on hardware and software resources. Nature of cloud applications also is advantageous for users since they do not need to have very technical hardware to run applications as these computing operations are run within the cloud. This reduces the price of mobile computing to the end users. They could see a huge number of new features enhancing their phones due to Mobile Cloud Computing This section started with some introductory remarks to Cloud computing and mobile cloud computing. The rest of the paper is organized as follows. Section II describes the concept of cloud computing, mobile Internet, mobile cloud

computing and its characters.

Application development:

Cloud computing is rapidly emerging as a new paradigm for delivering IT services as utility-oriented services on subscription basis. The rapid development of applications and their deployment in Cloud computing environments in efficient manner is a complex task.

Application development, deployment and runtime management have always been reliant on development platforms such as Microsoft's .NET, WebSphere, or JBoss, which have been deployed on-premise traditionally. In the Cloud-computing context, applications are generally deployed by Cloud providers to provide highly scalable and elastic services to as many end users as possible. The need for support as many users to access and utilize the same application services, with elastic resources allocation have led to enhancement in development platform technologies and architectures to handle performance, security, resource allocation, application monitoring, billing, and fault tolerance.

UNIT -II

CLOUD COMPUTING ARCHITECTURE

Cloud computing architecture refers to the components and subcomponents required for cloud computing. These components typically consist of a front end platform (fat client, thin client, mobile device), back end platforms (servers, storage), a cloud based delivery, and a network (Internet, Intranet, Inter) Combined, these components make up cloud computing architecture.

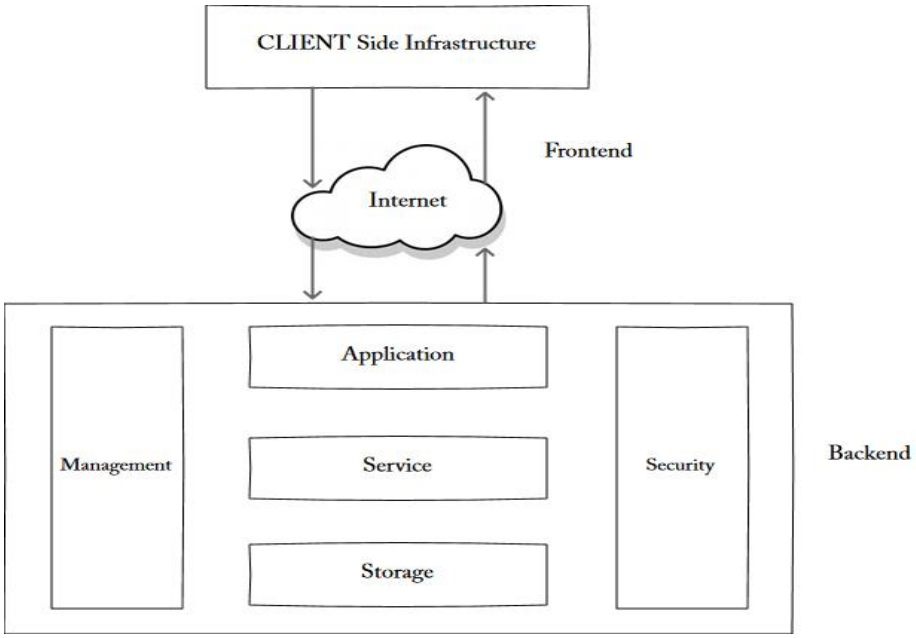

Figure 2.1 Cloud computing architecture

Businesses used cloud infrastructures to work with these applications. Unlike subscription-based models of pricing, payment structure of the cloud enables the user to subscribe to vendor services & cloud infrastructures are paid on a 'pay-per-use' basis. The cloud technology architecture also consists of front-end platforms (as read in the early chapters) called the cloud client which comprises servers, thin & fat client, tablets & mobile devices. The interaction is done through middleware or via web-browser or virtual sessions. The cloud architecture is a combination of both services oriented architecture & event-driven architecture. SO cloud architecture encompasses all elements of the cloud environment.

Cloud Computing Reference Architecture: An Overview

The Conceptual Reference Model

Figure presents an overview of the NIST cloud computing reference architecture, which identifies the major actors, their activities and functions in cloud computing. The diagram depicts a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

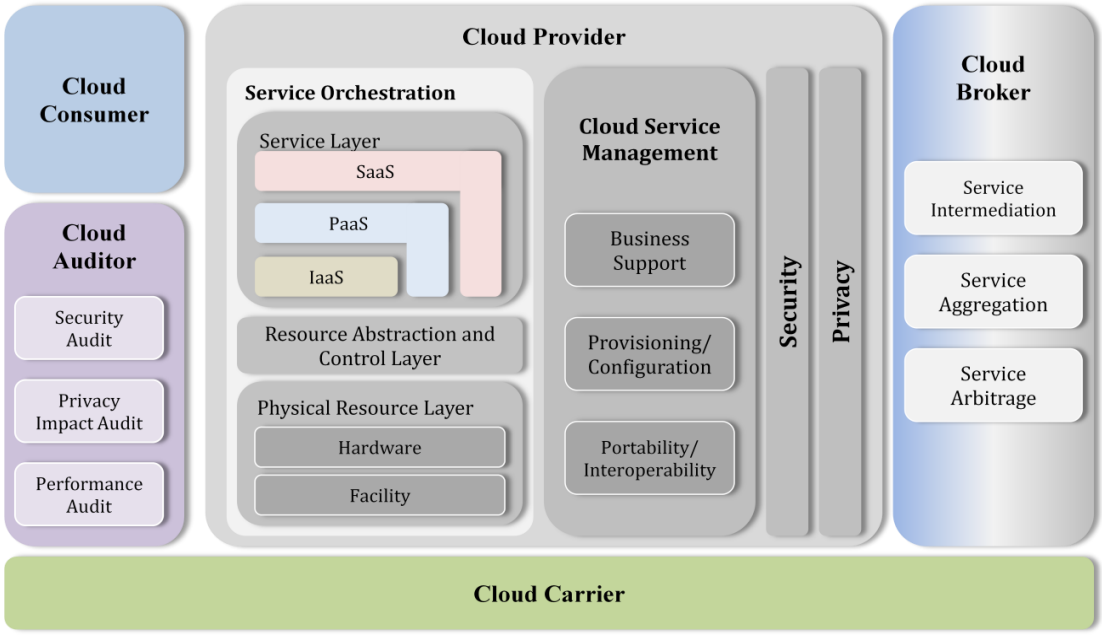

Figure 2.2 The Conceptual Reference Model

As shown the NIST cloud computing reference architecture defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker. Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing. Table 1 briefly lists the actors defined in the NIST cloud computing reference architecture.

- The general activities of the actors are discussed in the remainder of this section. A cloud consumer may request cloud services from a cloud provider directly or via a cloud broker.

- A cloud auditor conducts independent audits and may contact the others to

collect necessary information. The details will be discussed in the following sections and presented in increasing level of details in successive diagrams.

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, *Cloud Providers*.
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

Table 1: Actors in Cloud Computing

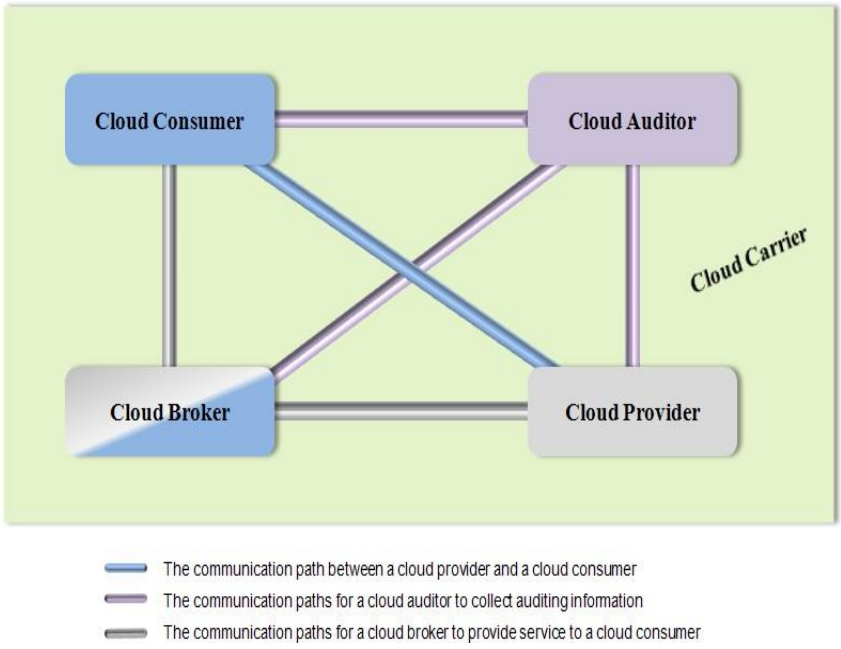

Figure 2.3 Interactions between the Actors in Cloud Computing

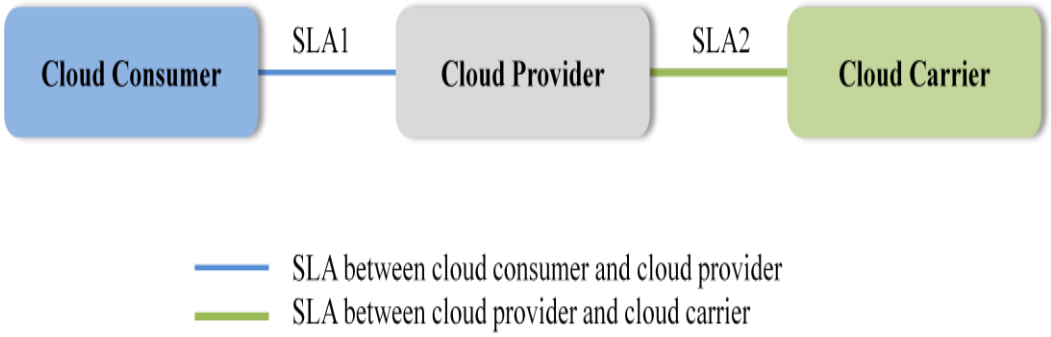

Figure 2.4 Cloud provider

Example Usage Scenario 1: A cloud consumer may request service from a cloud broker instead of contacting a cloud provider directly. The cloud broker may create a new service by combining multiple services or by enhancing an existing service. In this example, the actual cloud providers are invisible to the cloud consumer and the cloud consumer interacts directly with the cloud broker.

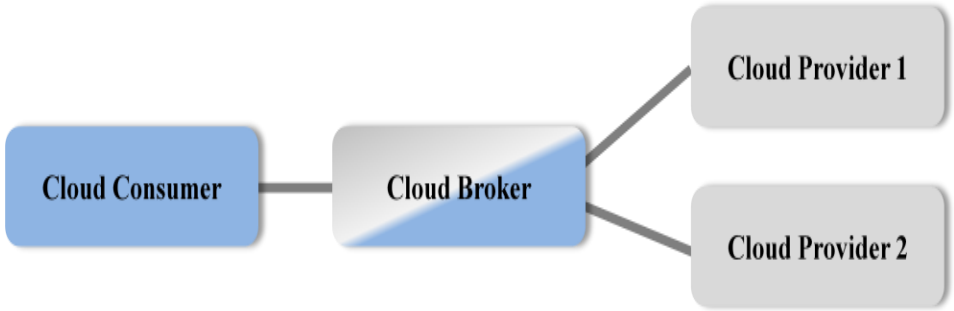

Figure 2.5 Usage Scenarios for Cloud Brokers

- **Example Usage Scenario 2**: Cloud carriers provide the connectivity and transport of cloud services from cloud providers to cloud consumers. As illustrated in Figure 4, a cloud provider participates in and arranges for two unique service level agreements (SLAs), one with a cloud carrier (e.g. SLA2) and one with a cloud consumer (e.g. SLA1). A cloud provider arranges service level agreements (SLAs) with a cloud carrier and may request dedicated and encrypted connections to ensure the cloud services are consumed at a consistent level according to the contractual obligations with the cloud consumers. In this case, the provider may specify its requirements on capability, flexibility and functionality in SLA2 in order to provide essential requirements in SLA1.

- **Example Usage Scenario 3**: For a cloud service, a cloud auditor conducts independent assessments of the operation and security of the cloud service implementation. The audit may involve interactions with both the Cloud Consumer and the Cloud Provider.

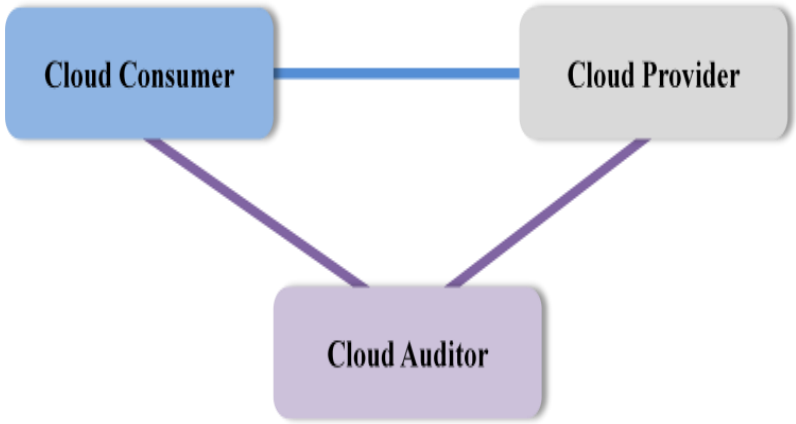

Figure 2. 6 Usage Scenario for Cloud Auditors

The Three-Tier Model of Cloud Computing

Cloud services are usually divided into three basic levels, or tiers, that are differentiated by the level of abstraction they present to consumers. The lowest tier is infrastructure-as-a-service (IaaS). With IaaS, users run software on machines owned and configured by a third party.

- IaaS makes up the hardware and software that supports the whole system, from the server to storage and networks. The next tier is platform-as-a-service (PaaS).

- PaaS is basically a platform that supports the complete life-cycle of building and delivering applications and services over the Web.

- PaaS provides services to deploy, test, host, and maintain applications within the same integrated development environment. With PaaS, each platform component whether middleware, messaging, integration, or communication – is provided as a service.

- The most visible layer to end users is software-as-a-service (SaaS), which rests on top of the PaaS layer. With SaaS, users can access pre-baked services simply by

navigating to them via a Web browser, without bothering with the hardware and software details involved in the implementation.

Architectural styles of cloud applications

Cloud transitions can be difficult to begin. Transitions can be difficult to design and plan, as much of the diligence now falls on the customer side. This change is a double-edged sword; it cuts both ways. It enables the customer to have significantly more control over designs, technical choices, economics, and risk. It also places the significantly more of the cloud computing architecture burden on the customer, who may not have the level of solution design experience that many service providers do.

Baseline cloud computing architectures are foundational building blocks to cornerstone design ideas. These common design arrangements can be used to jump-start solution efforts. Baseline architectures are useful when leveraging standard cloud computing patterns. Patterns represent cloud service requirements, while baseline architectures provide useful models for handling common architectural components and their associated requirements.

Each of the following sections will build on the section previous. The baseline compute component takes into account a web layer, application layer, and database layer, each having some level of storage. Storage attributes will change based on design requirements. Nearly all modern designs will have web, app, and database layers in their designs.

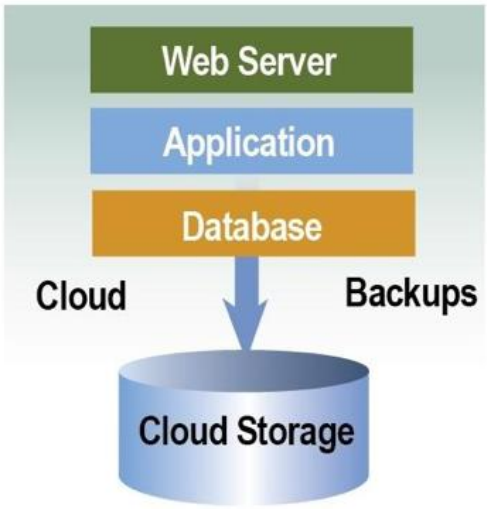

Figure.2.7 Three-tier-model1

This type of layering is called tiring. Most designs will have three or four tiers. Tiers are typically the number of individual isolated layers between the environment entry point and the destination data. As an example, three-tier architecture has a web layer, app layer, and database layer. Single-server architecture will have all three layers residing on the same virtual or physical server.

In this article, we will cover the following topics:

- Baseline architecture types
- OSI model and layer description
- Complex architecture types
- Architecting for hybrid clouds

Baseline architecture types

The various types of baseline architectures are as follows.

Single server

Single server templates represent the use of one server, virtual or physical, that contains a web server, an application, and a database. Single server architectures are not very common, as they have inherent security risks as one compromise can compromise all. These architectures are commonly deployed for development work, allowing developers to quickly build functionality without having to deal with connectivity and communication issues between different servers, potentially in different locations.

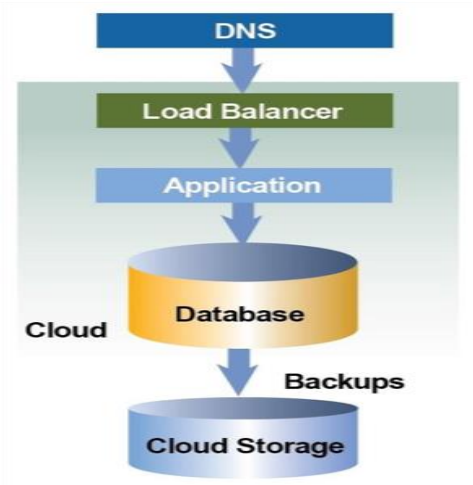

Figure 2.8 Single-site non-redundant-cloud architecture2

Single-site architectures take the single server architecture and split all of the layers into their own compute instances, creating the three-tier architecture mentioned. With all compute resources located in the same location, single-site architecture is created. There are two versions of single-site architectures: non- redundant and redundant.

Non-redundant three-tier architectures

Non-redundant three-tier architectures (at right) are used to save on costs and resources but must accept a higher risk. A single failure in any component, a single point of failure, can stop traffic flowing correctly into or out of the environment. This approach is commonly used for development or testing environments only. The following figure shows each layer, or tier, as a separate server, virtual or physical. Using this type of design for production environments is not recommended.

Redundant three-tier architectures

Redundant three-tier architectures add another set of the same components for redundancy. Additional design components do increase complexity, but are required if designing for failover and recovery protection. Designing redundant infrastructures requires a well thought out plan for the components within each layer (horizontal scaling), as well as a plan for how the traffic will flow from one layer to another (vertical scaling).

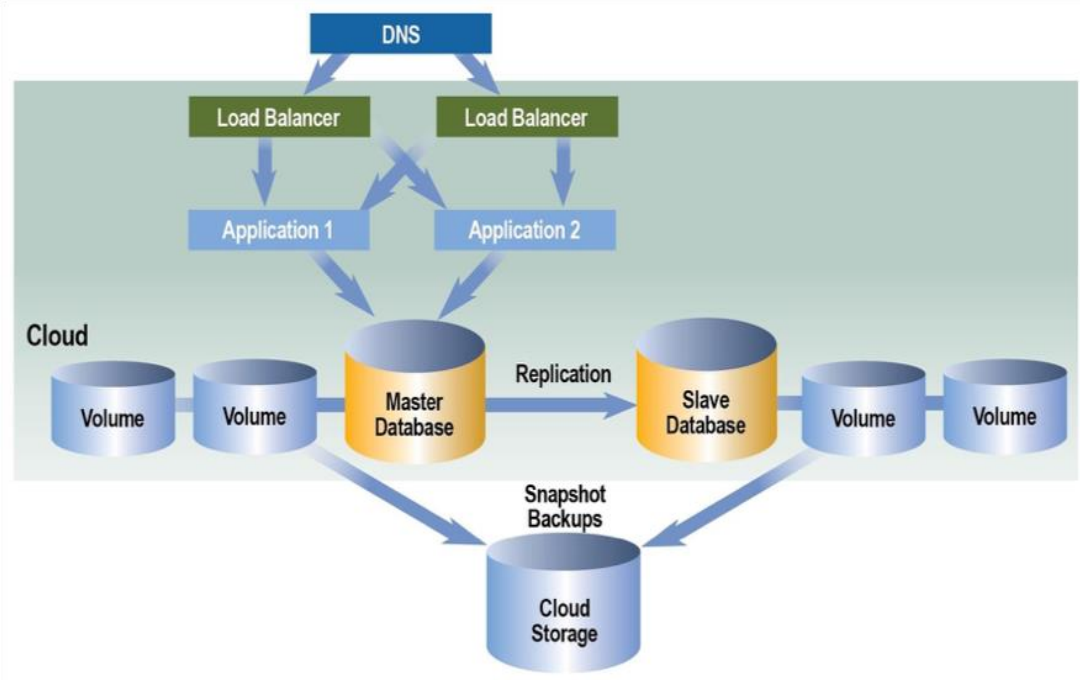

Figure 2.9 Redundant-cloud-architecture

Single points of failure

In redundant architectures, duplicate components eliminate the single point of failure present when only one device or component is present in the layer. With one component in a layer, there is only one way in and one way out. A second device adds multiple ingress and egress points to the design, eliminating the single point of failure associated with single-component layer designs.

Redundancy versus resiliency

Redundancy and resiliency are often confused. They are related, but not interchangeable. Redundancy is something that is done to prevent failure, implying that it happens before an issue happens. Resiliency, from the word resolve, relates to how to find solutions after a problem has occurred. Redundancy is before the issue. Resiliency is after. For example, redundant databases with replication can be utilized. Multiple components and copies of data create a redundant design. If the primary side of the database pair fails, the secondary side will promote to primary and begin to pick up the load while the failed side self-repairs.

Data-intensive computing

Data-intensive computing is a class of parallel computing applications which use a data parallel approach to process large volumes of data typically terabytes or petabytes in size and typically referred to as big data. Computing applications which devote most of their execution time to computational requirements are deemed compute-intensive, whereas computing applications which require large volumes of data and devote most of their processing time to I/O and manipulation of data are deemed data-intensive

Several common characteristics of data-intensive computing systems distinguish them from other forms of computing:

1. The principle of collection of the data and programs or algorithms is used to perform the computation. To achieve high performance in data-intensive computing, it is important to minimize the movement of data. This characteristic allows processing algorithms to execute on the nodes where the data resides reducing system overhead and increasing performance. Newer technologies such as InfiniBand allow data to be stored in a separate repository and provide performance comparable to collocated data.
2. The programming model utilized. Data-intensive computing systems utilize a machine-independent approach in which applications are expressed in terms of high-level operations on data, and the runtime system transparently controls the scheduling, execution, load balancing, communications, and movement of programs and data across the distributed computing cluster. The programming

abstraction and language tools allow the processing to be expressed in terms of data flows and transformations incorporating new dataflow programming languages and shared libraries of common data manipulation algorithms such as sorting.

3. A focus on reliability and availability. Large-scale systems with hundreds or thousands of processing nodes are inherently more susceptible to hardware failures, communications errors, and software bugs. Data-intensive computing systems are designed to be fault resilient. This typically includes redundant copies of all data files on disk, storage of intermediate processing results on disk, automatic detection of node or processing failures, and selective re-computation of results.
4. The inherent scalability of the underlying hardware and software architecture. Data-intensive computing systems can typically be scaled in a linear fashion to accommodate virtually any amount of data, or to meet time-critical performance requirements by simply adding additional processing nodes.

Compute-intensive

Compute-intensive is a term that applies to any computer application that demands a lot of computation, such as meteorology programs and other scientific applications. A similar but distinct term, computer-intensive, refers to applications that require a lot of computers, such as grid computing. The two types of applications are not necessarily mutually exclusive: some applications are both compute- and computer-intensive.

Compute-intensive applications

There are a number of characteristics that can make these applications unsuitable for traditional Java EE programming models:

- The need for asynchronous submission and start of work
- The need for work to run for extended periods of time
- The need for individual units of work to be visible to and manageable by operators and administrators

The compute-intensive programming model provides an environment that addresses these needs. The compute-intensive programming model is centered on two basic concepts:

- The use of jobs to submit and manage work asynchronously
- A minor extension to the asynchronous beans programming model to support work that runs for an extended period

UNIT-III

CLOUD RESOURCE VIRTUALIZATION

Cloud resource virtualization:

Resource Virtualization: Resource virtualization is used to create a layer of abstraction between actual physical hardware providing resources and the logical or semantic activities which consume those resources

Basics of virtualization

In computing, virtualization means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments.
Three Types of Server Virtualization: There are three ways to create virtual servers: full virtualization, para-virtualization and OS-level virtualization. They have little in common. Physical server is called host.

Types of Virtualization Techniques

Virtualization in Cloud Computing is making a virtual platform of server operating system and storage devices. ... Cloud Computing can also be known as services and application delivered to help the virtualized environment. This environment can be either public or private. Virtualization is technology that separates functions from hardware, while cloud surely on that split. Assuming intranet access, internet access, or both is already established, virtualization is what creates clouds. Software called a hypervisor sits on top of physical hardware and abstracts the machine's resources.

- Hardware/Server Virtualization.
- Network Virtualization.
- Storage Virtualization.
- Memory Virtualization.
- Software Virtualization.
- Data Virtualization.
- Desktop virtualization.

Full virtualization uses a special kind of software called a hypervisor. The hypervisor interacts directly with the physical server's CPU and disk space. Unlike the full virtualization technique, the guest servers in a Para-virtualization system are aware of one another. The very core of cloud computing is virtualization, which is used to separate a single physical machine into multiple virtual machines in a cost-effective way.

Hypervisor

A Type 1 hypervisor (also called a bare metal hypervisor) is installed directly on physical host server hardware just like an operating system.

Microsoft Hyper-V, codenamed Viridian and formerly known as Windows Server Virtualization, is a native hypervisor; it can create virtual machines on x86-64 systems running Windows. A server computer running Hyper-V can be configured to expose individual virtual machines to one or more networks.

Disadvantages: Virtual machines are less efficient than real machines because they access the hardware indirectly. Running software on top of the host operating system means that it will have to request access to the hardware from the host. A virtual machine can be infected with the weaknesses of the host machine.

Virtualization can increase IT agility, flexibility and scalability while creating significant cost savings. Greater workload mobility, increased performance and availability of resources, automated operations – they're all benefits of virtualization that make IT simpler to manage and less costly to own and operate.

Advantages of Server Virtualization is Saves money on IT costs. When you partition one physical server into several virtual machines, you can deploy, operate and manage multiple operating system instances at once on that single physical server. Fewer physical servers mean less money spent on those servers.

Full virtualization uses a special kind of software called a hypervisor. The hypervisor interacts directly with the physical server's CPU and disk space. Unlike the full virtualization technique, the guest servers in a Para-virtualization system are aware of one another.

It is true that in Para virtualization guest OSes run in isolation. Para virtualized AMIs have been well known to support only Linux. A guest OS ensures that it is in a virtualized state

Full virtualization

Full virtualization has proven highly successful for: sharing a computer system among multiple users; isolating users from each other (and from the control program); emulating new hardware to achieve improved reliability, security and productivity.

Paravirtualization is an enhancement of virtualization technology in which a guest OS is recompiled prior to installation inside a virtual machine. Paravirtualization allows for an interface to the virtual machine that can differ somewhat from that of the underlying hardware.

A Virtual Machine Monitor (VMM)

- It is a software program that enables the creation, management and governance of virtual machines (VM) and manages the operation of a virtualized environment on top of a physical host machine.

- Three Types of Server Virtualization: There are three ways to create virtual servers: full virtualization, para-virtualization and OS-level virtualization. They have little in common. Physical server is called host.

- Virtualization is technology that separates functions from hardware, while clouds rely on that split. Assuming intranet access, internet access, or both is already established, virtualization is what creates clouds. Software called a hypervisor sits on top of physical hardware and abstracts the machine's resources.

- Virtualization is the fundamental technology that powers cloud computing .It enables businesses to reduce IT costs while increasing the efficiency, utilization and flexibility of their existing computer hardware. Virtualization differs from cloud computing because virtualization is software that manipulates hardware

- Virtual machines are based on computer architectures and provide functionality of a physical computer. System virtual machines (also termed full virtualization VMs) provide a substitute for a real machine. They provide functionality needed to execute entire operating systems.

Virtualization limits costs by reducing the need for physical hardware systems. Virtual machines more efficiently use hardware, which lowers the quantities of hardware and associated maintenance costs, and reduces power and cooling demand. The reserved memory setting specifies a maximum amount of RAM that VMware Workstation is allowed to use. But this memory is not allocated in advance. Even if multiple virtual machines are running at the same time, VMware Workstation may be using only a fraction of the RAM you specify here.

Hardware, server, or platform virtualization is the technology of running a virtual operating system inside of another operating system. Basically, you now have two computers going. (If you don't already know, operating systems are what "run" your computer. Without them, you couldn't do much.

Virtualization limits costs by reducing the need for physical hardware systems. Virtual machines more efficiently use hardware, which lowers the quantities of hardware and associated maintenance costs, and reduces power and cooling demand.

Disadvantages:

- Virtual machines are less efficient than real machines because they access the hardware indirectly.
- Running software on top of the host operating system means that it will have to request access to the hardware from the host. That will slow the usability

Taxonomy of Virtual Machine

Virtual machine (VM) is an abstraction of computing resources presented to services to allow them to operate simultaneously on the same physical hardware infrastructure. VMs can be classified into two main categories: process and system virtual machines. A VM may be defined as a software implementation of a computing and operational environment on which system software like an operating system (OS) or application software like user programs can be installed and also run. The principal focus of virtualization is to maximization the hardware utilization and to minimize hardware cost by regrouping several machines virtualized into one host machine, minimize power utilization and make things easier for security and system management

Virtualization and VirtualMachine

A virtual machine is basically defined by “an efficient, isolated duplicate of a real machine” [by Popek and Goldberg]. Virtual machines have no direct interaction with any real hardware. A VM may be defined as a software implementation of a computing and operational environment on which system software like an operating

system (OS) or application software like user programs can be installed and also run. VMs are created within the Virtualization layer; the operating system, running on the servers or data centers, can be referred as Host Operating System and the operating systems in each virtual machine partition are called as Guest Operating System. The operating systems that run on the virtualization platform can be referred as Virtual Machine Monitor (VMM). The operating systems communicate with the hardware via VMM which can also be called as Virtual machine controller program shown in the figure The VMM virtualizes the hardware for each virtual machine.

Typically, guest operating systems and programs are not aware that they are running on a virtual platform. For example, the guest OS might appear to have a physical hard disk attached to it, but actual I/O requests are translated by the virtualization layer so they actually occur against a file that is accessible by the host OS. To make you understand, we present a schematic diagram concerned with virtualized system and also non-virtualized system.

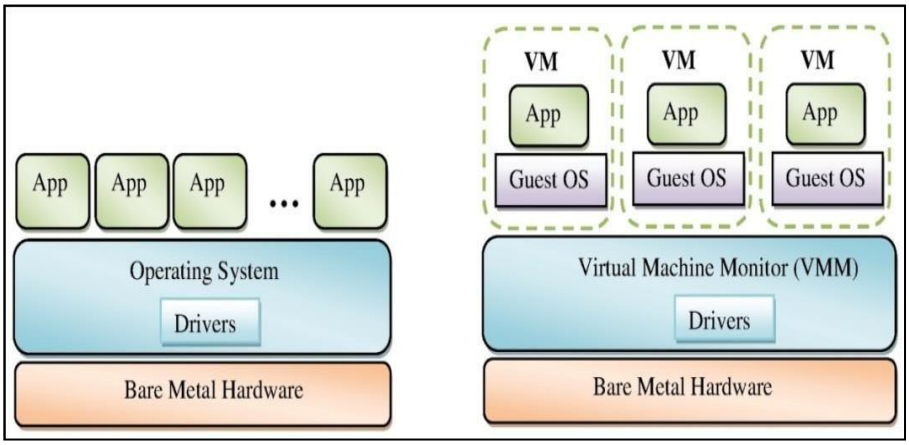

Figure 3.1 Virtualization and VirtualMachine

Virtualization enables different operating systems to run in the same computer concurrently at the same time. The Virtual Machine Manager (VMM) starts at the time of system-booting. All virtual machines run simultaneously.

The major characteristic of virtualization is the capability to make the abstraction of technical complexity from the cloud users, so that it can easily increase independence of cloud services. Secondly, physical and computing resources can be efficiently utilized after appropriate configuration, considering that on the same machine multiple applications are able to run. Thirdly, fault tolerance and quick recovery are also

permitted. Virtual environment can be easily backed up and migrated with no interruption in service.

System Virtual Machine:

A system virtual machine enables one computer to behave like two or more computers by sharing the host hardware's resources. Multiple virtual machines each of which running their own OS (This operating system is called guest operating system) are frequently utilized in server consolidation, where different cloud services are used to execute on separate machines. The use of virtual machines to support different guest OSes is becoming popular in embedded systems; Form the user point of view, they are using the hardware infrastructure through Virtual Machine and while the user request comes to the Cloud Service Provider (CSP), according to the need of the user, a particular VM is created and the resources are provisioned. When the task is completed, Virtual machine is also destroyed or we can say they shut down.

Process Virtual Machine:

A process virtual machine sometimes referred as an application virtual machine,

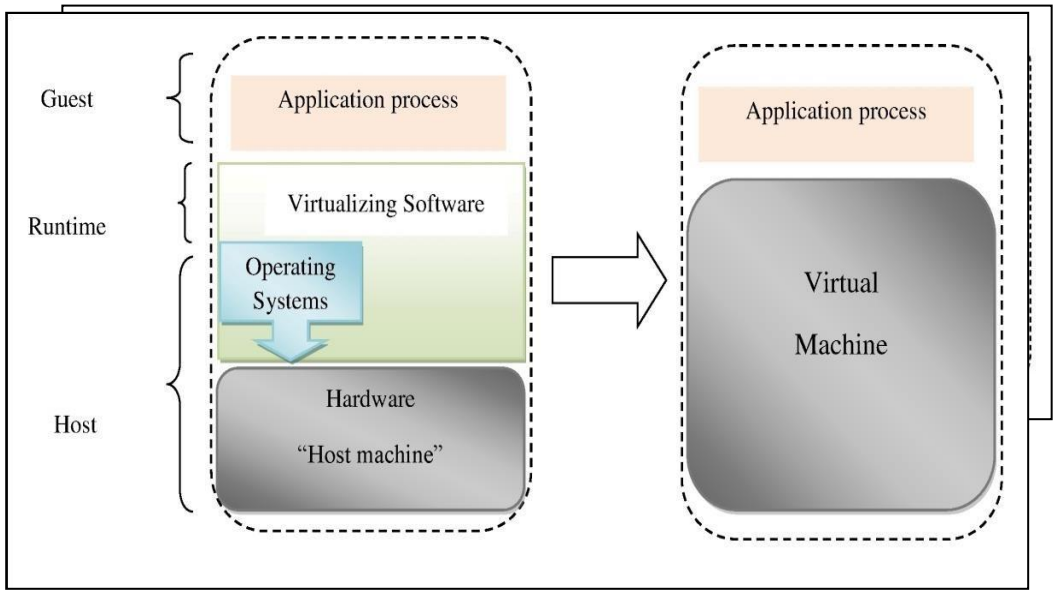

Figure 3.2 Process Virtual Machine

This runs as a normal application process inside operating systems and also supports a single process. Process Virtual machine is created when that particular process is

started and also destroyed when it exits. The Process Virtual Machine aims to provide a platform- independent programming environment which abstracts away the details of the underlying infrastructure.

A process VM can able to provide a high-level abstraction — that of a high-level programming language.

- Process VMs are being implemented using an interpreter; performance comparable to compiled programming languages is gained by the use of just-in-time compilation
- A schematic diagram of process virtual machine presented in figure the bare metal hardware of host machine, a host OS is installed and after that a vitalizing software is there to virtualize the hardware resources in run-time environment.
- According to the need of the user, a particular VM is created and the resources are provisioned. When the task is completed, Virtual machine is also destroyed or shut down.

Categorization of Guest OS Virtualization Techniques

The main focus of virtualization are to increase the hardware utilization to a maximum, decrease hardware costs by regrouping multiple machines virtualized into one physical machine, minimize power consumption and make things easier for security and system management.
We categorized the guest operating system virtualization techniques into three ways such as:

- Full virtualization
- Paravirtualization
- Hardware-assisted virtualization

Full virtualization

Full virtualization requires a virtualizable architecture; the hardware is fully exposed to the guest OS which runs unchanged and this ensure that this direct execution mode is efficient.

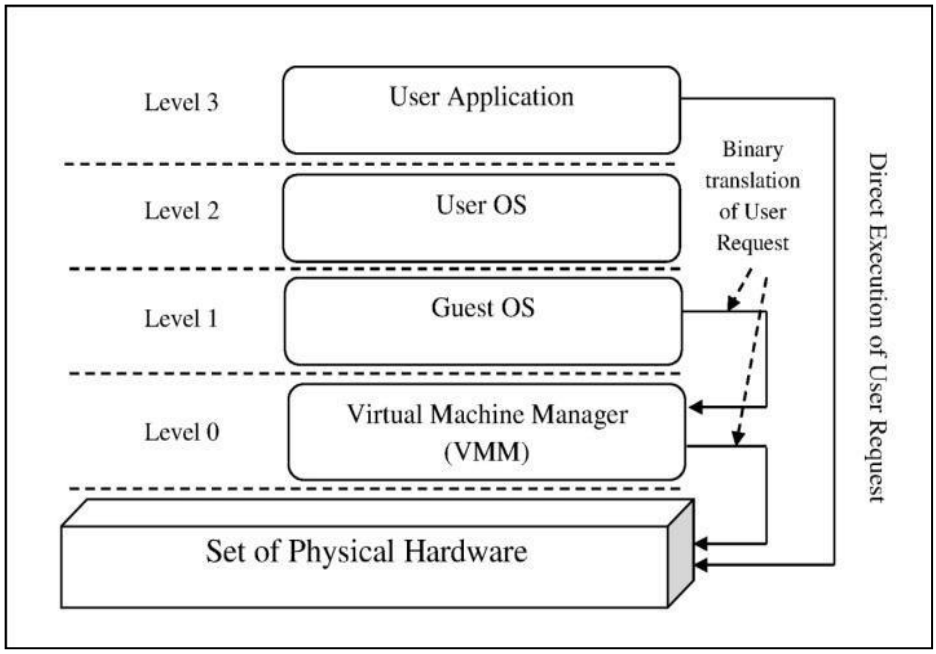

Figure 3.3 Full virtualization

In Full Virtualized architecture:

- Guest Operating systems are provided with all the services and that is given be physical computing systems that include virtualized memory, virtual devices and also virtual BIOS.

- In full virtualization operating system doesn't know that it is being virtualized. And that is the cause modification is not being required.

- In the figure VMM and guest OS will run on level-0 and level-1 respectively.

- To run the guest operating system without any modification, a particular technique is used which is Binary Translation.

Paravirtualization:

Paravirtualization is a process that is used to virtualize a guest OS. It is mainly helpful because it provides better performance than hardware-assisted or full virtualization

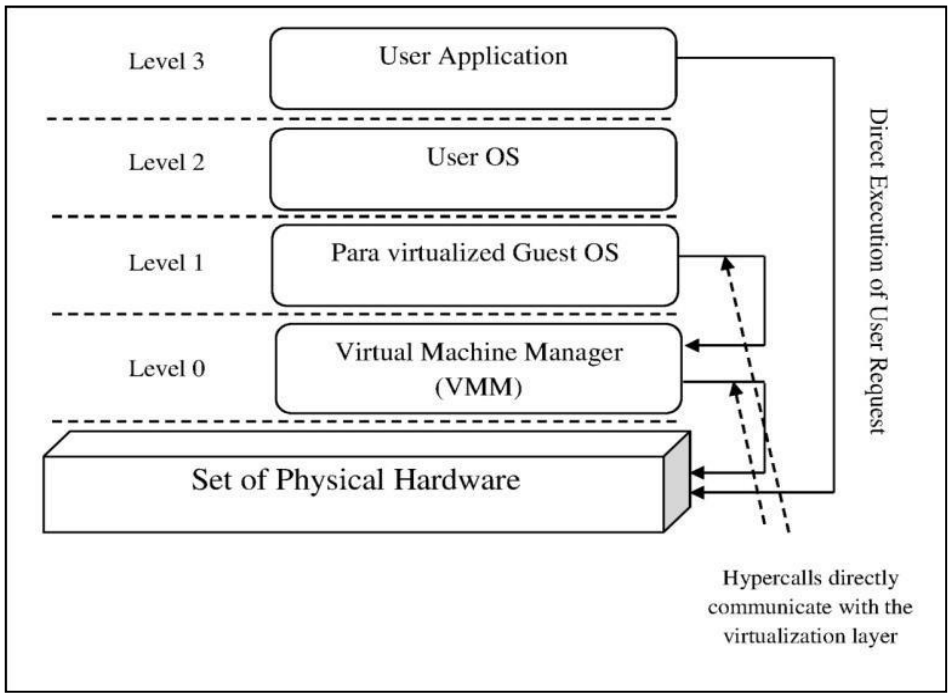

Figure 3.4 Paravirtualization

The reasons for adopting paravirtualization are as follows: firstly, some features of the hardware can't be virtualized. Secondly, it is used to present to the users a simpler interface. Paravirtualized Guest operating system is capable of communicating directly with the hypervisor and for that reason, as shown as the figure to communicate with the hypervisor or VMM, guest OS is needed the translation non-virtualizable instruction with hyper calls. For example, Xen and Denali are based on the technique of paravirtualization

Hardware-assisted virtualization

The shortcomings of paravirtualization which has been raised due to modification of hardware that allows the guest operating system to make communication with the hypervisor or VMM without any modifications, may be overcome by Hardware-assist virtualization

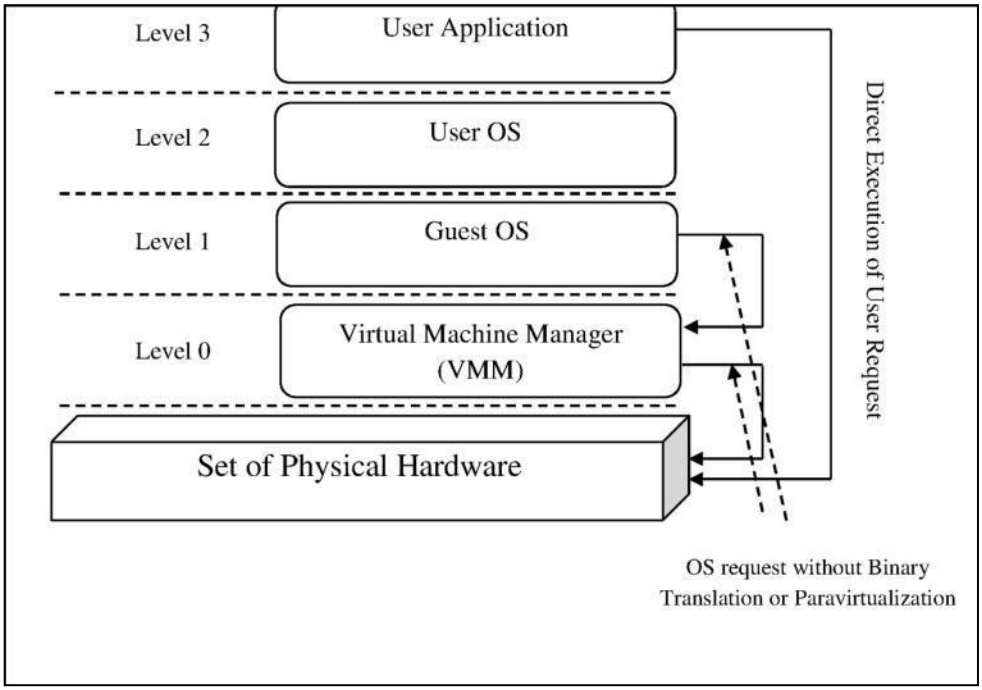

Figure 3.5 Hardware-assisted virtualization

Hardware-assisted guest operating system (level-1) is capable of communicating directly with the hypervisor (level-0) and for that reason, as shown as the figure, to communicate with the hypervisor or VMM without any paravirtualization or binary translation.

Characteristics of Virtualization

The cloud computing virtualization environment has the following characteristics discussed below:

Consolidation: Virtualization eliminates the need of a dedicated single system to one application and multiple OS can run in the same server. Both old and advanced version of OS may capable of deploying in the same platform without purchasing additional hardware and new required applications may be run simultaneously on their respective OS.

Easier development flexibility: Application developers may able to run and test their applications and programs in heterogeneous OS environments on the same virtualized machine. It facilitates the virtual machines to host heterogeneous OS. Isolation of different applications in their respective virtual partition also helps the developers. Migration and cloning: virtual machine can be moved from one site to another to

balance the workload. As the result of migration, users can access updated hardware as well as make recovery from hardware failure. Cloned virtual machines are easy to deploy in the local sites as well as remote sites.

Stability and security: In a virtualized atmosphere, host operating systems are hosting different types of multiple guest operating systems containing multiple applications. Each virtual machine is isolated from each other and they are not at all interfering into the other's work which in turn helps the security and stability aspect.

Paravirtualization: Paravirtualization is one of the important aspects of virtualization. In a virtual machine, guest OS can run on the host OS with or without modifications. If any changes or modifications are made to the operating system to be familiar with the Virtual Machine Manager (VMM), this process is said to be "paravirtualized".

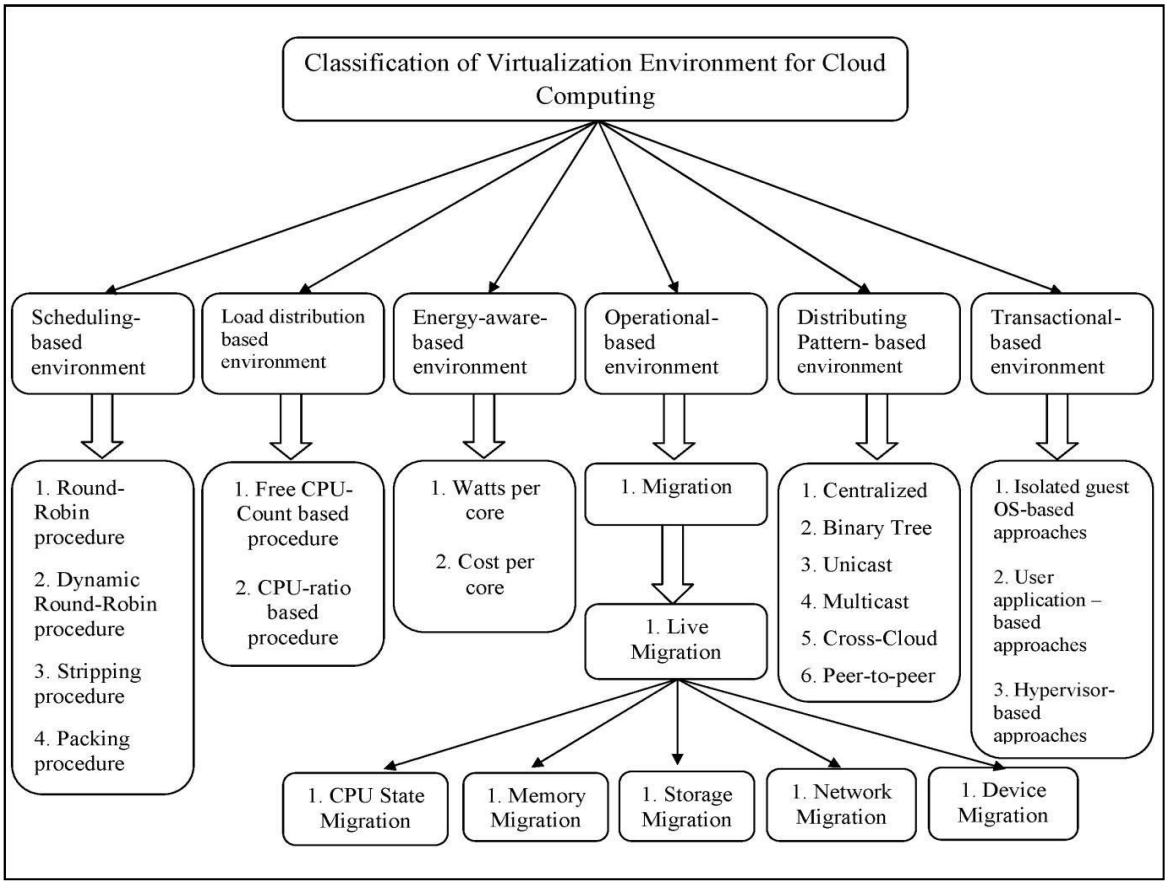

Figure 3.6 Schematic Diagram of Classification of Virtualization Environment

UNIT –IV

CLOUD RESOURCE MANAGEMENT AND SCHEDULING

Resource management is a core function of any man-made system. It affects the three basic criteria for the evaluation of a system: performance, functionality, and cost. An inefficient resource management has a direct negative effect on performance and cost and an indirect effect on the functionality of a system. Indeed, some functions provided by the system may become too expensive or may be avoided due to poor performance. A cloud is a complex system with a very large number of shared resources subject to unpredictable requests and affected by external events it cannot control. Cloud resource management requires complex policies and decisions for multi- objective optimization. Cloud resource management is extremely challenging because of the complexity of the system, which makes it impossible to have accurate global state information, and because of the unpredictable interactions with the environment.

The strategies for resource management associated with the three cloud delivery models, IaaS, PaaS, and SaaS, differ from one another. In all cases the cloud service providers are faced with large, fluctuating loads that challenge the claim of cloud elasticity. In some cases, when a spike can be predicted, the resources can be provisioned in advance, e.g., for Web services subject to seasonal spikes. For an unplanned spike, the situation is slightly more complicated. Auto Scaling can be used for unplanned spike loads, provided that

- There is a pool of resources that can be released or allocated on demand and
- There is a monitoring system that allows a control loop to decide in real time to reallocate resources.

Auto Scaling is supported by PaaS services such as Google App Engine. It has been argued for some time that in a cloud, where changes are frequent and unpredictable, centralized control is unlikely to provide continuous service and performance guarantees. Indeed, centralized control cannot provide adequate solutions to the host of cloud management policies that have to be enforced. Autonomic policies are of great interest due to the scale of the system, the large number of service requests, the large user population, and the unpredictability of the load. The ratio of the mean to the peak resource needs can be very large.

Policies and mechanisms for resource management

A policy typically refers to the principal guiding decisions, whereas mechanisms represent the means to implement policies. Separation of policies from mechanisms is a guiding principle in computer science. Cloud resource management policies can be loosely grouped into five classes:

- Admission control.
- Capacity allocation.
- Load balancing.
- Energy optimization.
- Quality-of-service (QoS) guarantees.

The explicit goal of an admission control policy is to prevent the system from accepting workloads in violation of high-level system policies; for example, a system may not accept an additional workload that would prevent it from completing work already in progress or contracted. Limiting the workload requires some knowledge of the global state of the system. In a dynamic system such knowledge, when available, is at best obsolete. Capacity allocation means to allocate resources for individual instances; an instance is an activation of a service. Locating resources subject to multiple global optimization constraints requires a search of a very large search space when the state of individual systems changes rapidly. Load balancing and energy optimization can be done locally, but global load balancing and energy optimization policies encounter the same difficulties as the one we have already discussed.

The common meaning of the term load balancing is that of evenly distributing the load to a set of servers. For example, consider the case of four identical servers, A, B,C, and D, whose relative loads are 80%, 60%, 40%, and 20%, respectively, of their capacity. As a result of perfect load balancing, all servers would end with the same load − 50% of each server's capacity. In cloud computing a critical goal is minimizing the cost of providing the service and, in particular, minimizing the energy consumption. This leads to a different meaning of the term load balancing; instead of having the load evenly distributed among all servers, we want to concentrate it and use the smallest number of servers while switching the others to standby mode, a state in which a server uses less energy. In our example, the load from D will migrate to A and the load from C will migrate to B; thus, A and B will be loaded at full capacity, whereas C and D will be switched to standby mode.

The performance decreases at a lower rate than does the energy when the clock rate decreases.		
CPU Speed (GHz)	**Normalized Energy *(%)***	**Normalized Performance *(%)***
0.6	0.44	0.61
0.8	0.48	0.70
1.0	0.52	0.79
1.2	0.58	0.81
1.4	0.62	0.88
1.6	0.70	0.90
1.8	0.82	0.95
2.0	0.90	0.99
2.2	1.00	1.00

Table 4.1 The normalized performance and energy consumption function of the processor speed

Speed Step and AMD's Power Now lower the voltage and the frequency to decrease power consumption. Motivated initially by the need to save power for mobile devices; these techniques have migrated to virtually all processors, including the ones used for high-performance servers.

Virtually all optimal – or near-optimal – mechanisms to address the five classes of policies do not scale up and typically target a single aspect of resource management, e.g., admission control, but ignore energy conservation. Many require complex computations that cannot be done effectively in the time available to respond. The performance models are very complex, analytical solutions are intractable, and the monitoring systems used to gather state information for these models can be too intrusive and unable to provide accurate data. Many techniques are concentrated on system performance in terms of throughput and time in system, but they rarely include energy tradeoffs or QoS guarantees. Some techniques are based on unrealistic assumptions; for example, capacity allocation is viewed as an optimization problem, but under the assumption that servers are protected from overload. Allocation techniques in computer clouds must be based on a disciplined approach rather than adhoc methods.

The four basic mechanisms for the implementation of resource management policies are:

- Control theory. Control theory uses the feedback to guarantee system stability and predict transient behavior but can be used only to predict local rather than global behavior. Kalman filters have been used for unrealistically simplified models.

- Machine learning. A major advantage of machine learning techniques is that they do not need a performance

- Utility-based. Utility-based approaches require a performance model and a mechanism to Correlate user-level performance with cost, as discussed in

- Market-oriented/economic mechanisms. Such mechanisms do not require a model of the system, e.g., combinatorial auctions for bundles of resources .A distinction should be made between interactive and non interactive workloads.

Resource bundling: Combinatorial auctions for cloud resources

Resources in a cloud are allocated in bundles, allowing users get maximum benefit from a specific combination of resources. Indeed, along with CPU cycles, an application needs specific amounts of main memory, disk space, network bandwidth, and so on. Resource bundling complicates traditional resource allocation models and has generated interest in economic models and, in particular, auction algorithms. In the context of cloud computing, an auction is the allocation of resources to the highest bidder.

Combinatorial Auctions: Auctions, in which participants can bid on combinations of items, or packages, are called combinatorial auctions. Such auctions provide a relatively simple, scalable, and tractable solution to cloud resource allocation.

- Two recent combinatorial auction algorithms are the simultaneous clock auction and the clock proxy auction.

- The algorithm discussed in this chapter and introduced in is called the ascending clock auction (ASCA). In all these algorithms the current price for each resource is represented by a “clock” seen by all participants at the auction.

- We consider a strategy in which prices and allocation are set as a result of an auction. In this auction, users provide bids for desirable bundles and the price they are willing to pay.

- We assume a population of U users, u = {1, 2,..., U}, and R resources, r = {1, 2,..., R}. The bid of user u is Bu = {Qu,πu} with Qi = (q1 u , q2 u , q3 u ,...) an R-component vector;

- Each element of this vector, qi u, represents a bundle of resources user u would accept and, in return, pay the total price πu.

- Each vector component qi u is a positive quantity and encodes the quantity of a resource desired or, if negative, the quantity of the resource offered.

- A user expresses her desires as indifference set I = (q1 u XOR q2 u XOR q3 u XOR ...). The final auction prices for individual resources are given by the vector p = (p1, p2,..., pR) and the amounts of resources allocated to user u are xu = (x1 u , x2 u ,..., x R u).

- Thus, the expression [(xu)T p] represents the total price paid by user u for the bundle of resources if the bid is successful at time T .The scalar [minq∈Qu (qT p)] is the final price established through the bidding process. The bidding process aims to optimize an objective function f (x, p).

  This function could be tailored to measure the net value of all resources traded, or it can measure the total surplus – the difference between the maximum amounts users are willing to pay minus the amount they pay. Other optimization functions could be considered for a specific system, e.g., the minimization of energy consumption or of security risks. Pricing and Allocation Algorithms, A pricing and allocation algorithm partitions the set of users into two disjoint sets, winners and losers, denoted as W and L, respectively. The algorithm should:

  1. be computationally tractable. Traditional combinatorial auction algorithms such as Vickey-ClarkeGroves (VLG) fail these criteria, because they are not computationally tractable.

  2. Scale well. Given the scale of the system and the number of requests for service, scalability is a necessary condition.

3. Be objective. Partitioning in winners and losers should only be based on the price πu of a user's bid. If the price exceeds the threshold, the user is a winner; otherwise the user is a loser.

4. be fair. Make sure that the prices are uniform. All winners within a given resource pool pay the same price.

5. Indicate clearly at the end of the auction the unit prices for each resource pool.

6. Indicate clearly all participants the relationship between the supply and the demand in the system. The function to be maximized is max x,p f (x, p). The constraints in Table correspond to our intuition:

(a) the first one states that a user either gets one of the bundles it has opted for or nothing; no partial allocation is acceptable.

(b) The second constraint expresses the fact that the system awards only available resources; only offered resources can be allocated.

(c) The third constraint is that the bid of the winners exceeds the final price.

(d) The fourth constraint states that the winners get the least expensive bundles in their indifference set.

(e) The fifth constraint states that losers bid below the final price.

(f) The last constraint states that all prices are positive numbers.

The ASCA Combinatorial Auction Algorithm: Informally, in the ASCA algorithm the participants at the auction specify the resource and the quantities of that resource offered or desired at the price listed for that time slot. Then the excess vector is computed. If all its components are negative, the auction stops; negative components mean that the demand does not exceed the offer. If the demand is larger than the offer, *z(t)* ‘‘ 0, the auctioneer increases the price for items with a positive excess demand and solicits bids at the new price. Note that the algorithm satisfies conditions 1 through 6; from Table 6.3 all users discover the price at the same time

An auctioning algorithm is very appealing because it supports resource bundling and does not require a model of the system. At the same time, a practical implementation of such algorithms is challenging. First, requests for service arrive at random times, whereas in an auction all participants must react to a bid at the same time. Periodic auctions must then be organized, but this adds to the delay of the response. Second, there is an incompatibility between cloud elasticity, which guarantees that the demand for resources of an existing application will be satisfied immediately, and the idea of

periodic auctions.

Fair queuing

Computing and communication on a cloud are intimately related. Therefore, it should be no surprise that the first algorithm we discuss can be used for scheduling packet transmission as well as threads. Interconnection networks allow cloud servers to communicate with one another and with users. These networks consist of communication links of limited bandwidth and switches/routers/gateways of limited capacity. When the load exceeds its capacity, a switch starts dropping packets because it has limited input buffers for the switching fabric and for the outgoing links, as well as limited CPU cycles. A switch must handle multiple flows and pairs of source-destination endpoints of the traffic.

Start-time fair queuing

A hierarchical CPU scheduler for multimedia operating systems was proposed. The basic idea of the start-time fair queuing (SFQ) algorithm is to organize the consumers of the CPU bandwidth in a tree structure; the root node is the processor and the leaves of this tree are the threads of each application. A scheduler acts at each level of the hierarchy. The fraction of the processor bandwidth, B, allocated to the intermediate node i is Bi B = wi n j=1 wj (6.31) with wj, 1 j n, the weight of the n children of node i; see the example in Figure 6.9. When a virtual machine is not active, its bandwidth is reallocated to the other VMs active at the time.

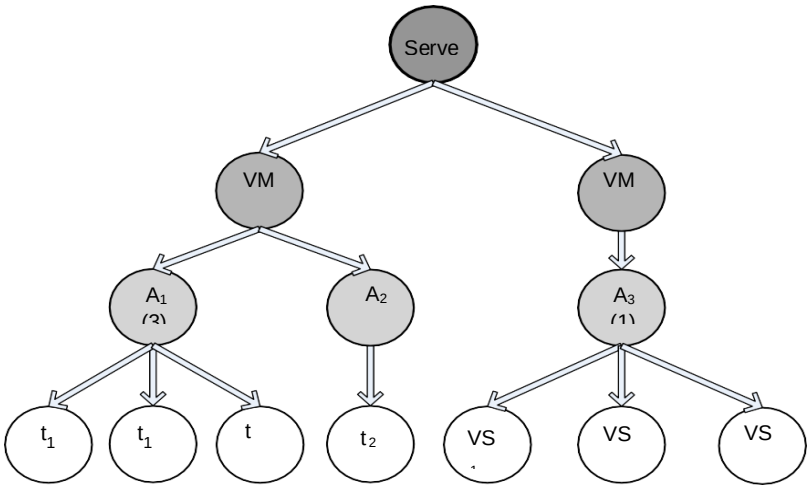

Figure 4.1 Start-time fair queuing

When one of the applications of a virtual machine is not active, its allocation is transferred to the other

Borrowed virtual time

The objective of the borrowed virtual time (BVT) algorithm is to support low-latency dispatching of real-time applications as well as a weighted sharing of the CPU among several classes of applications. Like SFQ,

- The BVT algorithm supports scheduling of a mix of applications, some with hard, some with soft real-time constraints, and applications demanding only a best effort.
- Thread i has an effective virtual time, Ei , an actual virtual time, Ai , and a virtual time warp, Wi .The scheduler thread maintains its own scheduler virtual time (SVT), defined as the minimum actual virtual time Aj of any thread.
- The threads are dispatched in the order of their effective virtual time, Ei, a policy called the earliest virtual time (EVT).
- The virtual time warp allows a thread to acquire an earlier effective virtual time – in other words, to borrow virtual time from its future CPU allocation.
- The virtual warp time is enabled when the variable warp back is set. In this case a latency-sensitive thread gains dispatching preference as Ei ← Ai if warp Back = OFF

Ai − Wi if warpBack = ON.

The algorithm measures the time in minimum charging units (mcu) and uses a time quantum called context switch allowance (C), which measures the real time a thread is allowed to run when competing with other threads, measured in multiples of mcu. Typical values for the two quantities are mcu = 100 μsec and C = 100 msec. A thread is charged an integer number of mcu. Context switches are triggered by traditional events, the running thread is blocked waiting for an event to occur, the time quantum expires, and an interrupt occurs. Context switching also occurs when a thread becomes runnable after sleeping. When the thread τi becomes runnable after sleeping, its actual virtual time is updated as follows:

Ai ← max[Ai, SVT]. (6.60)

This policy prevents a thread sleeping for a long time to claim control of the CPU for

a longer period of time than it deserves. If there are no interrupts, threads are allowed to run for the same amount of virtual time. Individual threads have weights; a thread with a larger weight consumes its virtual time more slowly. In practice, each thread τi maintains a constant ki and uses its weight wi to compute the amount used to advance its actual virtual time upon completion of a run:

Ai ← Ai + .

Given two threads a and b, = ka/wa= kb/wb.

The EVT policy requires that every time the actual virtual time is updated, a context switch from the current running thread τi to a thread τ j occurs if

Aj<=Ai – C/wi.

Cloud scheduling subject to deadlines

SLA specifies the time when the results of computations done on the cloud should be available. This motivates us to examine cloud scheduling subject to deadlines, a topic drawing on a vast body of literature devoted to real-time applications.

Task Characterization and Deadlines

Real-time applications involve periodic or a periodic tasks with deadlines. A task is characterized by a tuple (Ai,σi, Di), where Ai is the arrival time, σi > 0 is the data size of the task, and Di is the relative deadline. Instances of a periodic task, q i , with period q are identical, q i ≡ q , and arrive at times A0, A1,... Ai,..., with Ai+1 − Ai = q. The deadlines satisfy the constraint Di Ai+1 and generally the data size is the same, σi = σ. The individual instances of a periodic tasks, i , are different. Their arrival times Ai are generally uncorrelated, and the amount of data σi is different for different instances. The absolute deadline for the periodic task i is (Ai + Di). We distinguish hard deadlines from soft deadlines.

In the first case, if the task is not completed by the deadline, other tasks that depend on it may be affected and there are penalties; a hard deadline is strict and expressed precisely as milliseconds or possibly seconds. Soft deadlines play more of a guideline role and, in general, there are no penalties. Soft deadlines can be missed by fractions of the units used to express them, e.g., minutes if the deadline is expressed in hours or hours if the deadlines are expressed in days.

The scheduling of tasks on a cloud is generally subject to soft deadlines, though occasionally applications with hard deadlines may be encountered. System Model. In our discussion we consider only periodic tasks with arbitrarily divisible workloads.

The application runs on a partition of a cloud, a virtual cloud with a head node called S0 and n worker nodes S1, S2... Sn. The system is homogeneous, all workers are identical, and the communication time from the head node to any worker node is the same. The head node distributes the workload to worker nodes, and this distribution is done sequentially. In this context there are two important problems: 1. The order of execution of the tasks i . 2. The workload partitioning and the task mapping to worker nodes. The most common scheduling policies used to determine the order of execution of the tasks are: • First in, first out (FIFO). The tasks are scheduled for execution in the order of their arrival. • Earliest deadline first (EDF). The task with the earliest deadline is scheduled first. • Maximum workload derivative first (MWF).

The workload derivative DCi(nmin) of a task i when nmin nodes are assigned to the application, is defined as DCi(nmin) = Wi nmin i + 1 − Wi nmin I, with Wi(n) the workload allocated to task i when n nodes of the cloud are available; if E(σi, n) is the execution time of the task, then Wi(n) = n × E(σi, n). The MWF policy requires that: 1.The tasks are scheduled in the order of their derivatives, the one with the highest derivative DCi first. 2. The number n of nodes assigned to the application is kept to a minimum, nmin i . We discuss two workload partitioning and task mappings to worker nodes, optimal and the equal partitioning. Optimal Partitioning Rule (OPR).

Scheduling *MapReduce* applications subject to deadlines

Now we turn our attention to applications of the analysis in Section 6.12 and discuss scheduling of *MapReduce* applications on the cloud subject to deadlines. Several options for scheduling Apache *Hadoop*, an open-source implementation of the *MapReduce* algorithm, are:

- The default FIFO schedule.
- The Fair Scheduler.
- The Capacity Scheduler.

The parameters used for scheduling with deadlines.

Name	Description
Q	The query $Q = (A, \sigma, D)$
A	Arrival time of query Q
D	Deadline of query Q
γ_i^m ™ i ™ u_m	A map task, 1
γ_r j	A reduce task, 1 ™ j ™ v
J	The job to perform the query $Q = (A, \sigma, D)$, $J = \gamma^1, \gamma^2, \ldots, \gamma^u$,
$\gamma^1, \gamma^2, \ldots, \tau$	Cost for transferring a data unit
ρ_m	Cost of processing a unit data in map task
ρ_r	Cost of processing a unit data in reduce task
n_m	Number of map slots
n_r	Number of reduce slots
n_m^{min}	Minimum number of slots for the map task
n	Total number of slots, $n = n_m + n_r$
t_m^0	Start time of the map task
t_r^{max}	Maximum value for the start time of the reduce task
α	Map distribution vector; the EPR strategy is used and, $\alpha_i = 1/u$
φ	Filter ratio, the fraction of the input produced as output by the map process

Figure 4.2 The parameters used for scheduling with deadlines

Resource management and dynamic application scaling

The demand for computing resources, such as CPU cycles, primary and secondary storage, and network bandwidth, depends heavily on the volume of data processed by an application.

- The demand for resources can be a function of the time of day, can monotonically increase or decrease in time, or can experience predictable or unpredictable peaks. For example, a new Web service will experience a low request rate when the service is first introduced and the load will exponentially increase if the service is successful.
- A service for income tax processing will experience a peak around the tax filing deadline, whereas access to a service provided by Federal Emergency Management Agency (FEMA) will increase dramatically after a natural disaster. The elasticity of a public cloud, the fact that it can supply to an application precisely the amount of resources it needs and that users pay only

for the resources they consume are serious incentives to migrate to a public cloud.

- The question we address is: How scaling can actually be implemented in a cloud when a very large number of applications exhibit this often unpredictable behavior. To make matters worse, in addition to an unpredictable external load the cloud resource management has to deal with resource reallocation due to server failures. We distinguish two scaling modes: vertical and horizontal.
- Vertical scaling keeps the number of VMs of an application constant, but increases the amount of resources allocated to each one of them. This can be done either by migrating the VMs to more powerful servers or by keeping the VMs on the same servers but increasing their share of the CPU time. The first alternative involves additional overhead; the VM is stopped, a snapshot of it is taken, the file is transported to a more powerful server, and, finally, the VM is restated at the new site.
- Horizontal scaling is the most common mode of scaling on a cloud; it is done by increasing the number of VMs as the load increases and reducing the number of VMs when the load decreases. Often, this leads to an increase in communication bandwidth consumed by the application. Load balancing among the running VMs is critical to this mode of operation. For a very large application, multiple load balancers may need to cooperate with one another. In some instances the load balancing is done by a front-end server that distributes incoming requests of a transaction-oriented system to back-end servers.

An application should be designed to support scaling, the workload partitioning is static, it is decided a priori, and cannot be changed; thus, the only alternative is vertical scaling. In the case of an arbitrarily divisible application the workload can be partitioned dynamically; as the load increases, the system can allocate additional VMs to process the additional workload. Most cloud applications belong to this class, which justifies our statement that horizontal scaling is the most common scaling mode.

Mapping a computation means to assign suitable physical servers to the application. A very important first step in application processing is to identify the type of application and map it accordingly. For example, a communication-intensive application should be mapped to a powerful server to minimize the network traffic. This may increase the cost per unit of CPU usage, but it will decrease the computing time and probably reduce the overall cost for the user.

UNIT-V

CLOUD SECURITY

Security has been a concern since the early days of computing, when a computer was isolated in a room and a threat could be posed only by malicious insiders. The Pandora's box1 of threats opened wide once computers were able to communicate with one another. In an interconnected world, various embodiments of malware can migrate easily from one system to another, cross national borders, and infect systems all over the globe. The security of computing and communication systems takes on a new urgency as society becomes increasingly dependent on the information infrastructure. Nowadays, even the critical infrastructure of a nation can be attacked by exploiting flaws in computer security.

Recently, the term cyber warfare has entered the dictionary with the meaning "actions by a nation-state to penetrate another nation's computers or networks for the purposes of causing damage or disruption" .A computer cloud is a target-rich environment for malicious individuals and criminal organizations. It is thus no surprise that security is a major concern for existing users and for potential new users of cloud computing services. We identified some of the security threats perceived by cloud users, some of these risks are shared with other systems supporting network-centric computing and network-centric content, e.g., service-oriented architectures (SOAs), grids, and Web-based services.

Cloud computing is an entirely new approach to computing based on a new technology. It is therefore reasonable to expect that new methods to deal with some of the security threats will be developed, whereas other perceived threats will prove to be exaggerated. Indeed, "early on in the life cycle of a technology, there are many concerns about how this technology will be used. They represent a barrier to the acceptance over the time; however, the concerns fade, especially if the value proposition is strong enough". The idea that moving to a cloud liberates an organization from many technical concerns related to computer security and eliminates internal threats is accepted by some members of the IT community.

One of the consequences of the breathtaking pace of development of information science and technology is that standards, regulations, and laws governing the activities of organizations are supporting the new computing services, and in particular utility computing, have yet to be devised or adopted. As a result, many issues related to privacy, security, and trust in cloud computing are far from settled. The pool of resources of a cloud service provider can be dispersed over several countries or even

several continents. Since information can freely across national borders there is a need for international regulations to be adopted by the countries where data centers of cloud computing providers are located

Cloud security risks

Some believe that it is very easy, possibly too easy, to start using cloud services without a proper understanding of the security risks and without the commitment to follow the ethics rules for cloud computing. A first question is: What are the security risks faced by cloud users? There is also the possibility that a cloud could be used to launch large-scale attacks against other components of the cyber infrastructure. The next question is: How can the nefarious use of cloud resources are prevented? There are multiple ways to look at the security risks for cloud computing

- Traditional threats are those experienced for some time by any system connected to the Internet, but with some cloud-specific twists. The impact of traditional threats is amplified due to the vast amount of cloud resources and the large user population that can be affected.
- The fuzzy bounds of responsibility between the providers of cloud services and users and the difficulties in accurately identifying the cause of a problem add to cloud users' concerns. The traditional threats begin at the user site.
- The user must protect the infrastructure used to connect to the cloud and to interact with the application running on the cloud. This task is more difficult because some components of this infrastructure are outside the firewall protecting the user.

The next threat is related to the authentication and authorization process. The procedures in place for one individual do not extend to an enterprise. In this case the cloud access of the members of an organization must be nuanced; individuals should be assigned distinct levels of privilege based on their roles in the organization. It is also nontrivial to merge or adapt the internal policies and security metrics of an organization with the ones of the cloud. Moving from the user to the cloud, we see that the traditional types of attack have already affected cloud service providers. The favorite means of attack are distributed denial-of-service (DDoS) attacks, which prevent legitimate users accessing cloud services;

1 phishing
2 SQL injections
3 Cross-site scripting.

Cloud servers host multiple VMs, and multiple applications may run under each VM. Multiternency in conjunction with VMM vulnerabilities could open new attack channels for malicious users. Identifying the path followed by an attacker is much more difficult in a cloud environment. Traditional investigation methods based on digital forensics cannot be extended to a cloud, where the resources are shared among a large user population and the traces of events related to a security incident are wiped out due to the high rate of write operations on any storage media. Availability of cloud services is another major concern. System failures, power outages, and other catastrophic events could shut down cloud services for extended periods of time. When such an event occurs, data lock-in, could prevent a large organization whose business model depends on that data from functioning properly.

Clouds could also be affected by phase transition phenomena and other effects specific to complex systems. Another critical aspect of availability is that users cannot be assured that an application hosted on the cloud will return correct results. Third-party control generates a spectrum of concerns caused by the lack of transparency and limited user control. For example, a cloud provider may subcontract some resources from a third party whose level of trust is questionable. There are examples when subcontractors failed to maintain the customer data. There are also examples when the third party was not a subcontractor but a hardware supplier and the loss of data was caused by poor-quality storage devices. Storing proprietary data on a cloud is risky because cloud provider espionage poses real dangers. The terms of contractual obligations usually place all responsibilities for data security with the user.

The Amazon Web Services customer agreement, for example, does not help boost user confidence as it states: “We will not be liable to you for any direct, indirect, incidental damages nor be responsible for any compensation, reimbursement, arising in connection with:

(A) Your inability to use the services
(B) The cost of procurement of substitute goods or services or
(C) Any unauthorized access to, alteration of, or deletion, destruction, damage.

It is very difficult for a cloud user to prove that data has been deleted by the service provider. The lack of transparency makes audit ability a very difficult proposition for cloud computing.

Privacy and privacy impact assessment

The term privacy refers to the right of an individual, a group of individuals, or an organization to keep information of a personal or proprietary nature from being disclosed to others. Many nations view privacy as a basic human right. The Universal

Declaration of Human Rights, Everyone has the right to the protection of the law against such interference or attacks for example; taxation laws require individuals to share information about personal income or earnings. Individual privacy may conflict with other basic human rights, e.g., freedom of speech. Privacy laws differ from country to country; laws in one country may require public disclosure of information considered private in other countries and cultures.

The digital age has confronted legislators with significant challenges related to privacy as new threats have emerged. For example, personal information voluntarily shared, but stolen from sites granted access to it or misused, can lead to identity theft. Some countries have been more aggressive than others in addressing the new privacy concerns. This right addresses the following problem: Today it is very hard to escape your past when every photo, status update, and tweet lives forever on some Web site. Our discussion targets primarily public clouds where privacy has an entirely new dimension because the data, often in an unencrypted form, resides on servers owned by a CSP.

Operating system security

An operating system (OS) allows multiple applications to share the hardware resources of a physical system, subject to a set of policies. A critical function of an OS is to protect applications against a wide range of malicious attacks such as unauthorized access to privileged information, tempering with executable code, and spoofing. Such attacks can now target even single-user systems such as personal computers, tablets, or smart phones. Data brought into the system may contain malicious code; this could occur via a Java applet, or data imported by a browser from a malicious Web site.

- The mandatory security of an OS is considered to be “any security policy where the definition of the policy logic and the assignment of security attributes is tightly controlled by a system security policy administrator”.
- Access control, authentication usage, and cryptographic usage policies are all elements of mandatory OS security.
- The first policy specifies how the OS controls the access to different system objects, the second defines the authentication mechanisms the OS uses to authenticate a principal, and the last specifies the cryptographic mechanisms used to protect the data.
- A necessary but not sufficient condition for security is that the subsystems tasked with performing security-related functions are temper-proof and cannot be bypassed.

- The OS should confine an application to a unique security domain. Applications with special privileges that perform security-related functions are called trusted applications. Such applications should only be allowed the lowest level of privileges required to perform their functions. For example, type enforcement is a mandatory security mechanism that can be used to restrict a trusted application to the lowest level of privileges.
- Enforcing mandatory security through mechanisms left to the discretion of users could lead to a breach of security due not only to malicious intent but also carelessness or lack of understanding. Discretionary mechanisms place the burden of security on individual users.
- Moreover, an application may change a carefully defined discretionary policy without the consent of the user, whereas a mandatory policy can only be changed by a system administrator.

Commercial operating systems do not support multilayered security; such systems only distinguish between a completely privileged security domain and a completely unprivileged one.

- Some operating systems, such as Windows NT, allow a program to inherit all the privileges of the program invoking it, regardless of the level of trust in that program. The existence of trusted paths, mechanisms supporting user interactions with trusted software, is critical to system security. If such mechanisms do not exist, malicious software can impersonate trusted software.
- Some systems provide trust paths for a few functions such as login authentication and password changing and allow servers to authenticate their clients. Decompose a complex mechanism into several components with well-defined roles.
- For example, the access control mechanism for the application space could consist of enforcer and decider components. To access a protected object, the enforcer will gather the required information about the agent attempting the access and will pass this information to the decider, together with the information about the object and the elements of the policy decision. Finally, it will carry out the actions requested by the decider.

A trusted-path mechanism is required to prevent malicious software invoked by an authorized application to tamper with the attributes of the object and/or with the policy rules. A trusted path is also required to prevent an impostor from impersonating the decider agent. A similar solution is proposed for cryptography usage, which should be decomposed into an analysis of the invocation mechanisms and an analysis of the cryptographic mechanism. Another question is how an OS can protect itself and the applications running under it from malicious mobile code attempting to gain access to

the data and the other resources and compromise system confidentiality and/or integrity. Java Security Manager uses the type-safety attributes of Java to prevent unauthorized actions of an application running in a "sandbox." Yet, the Java Virtual Machine (JVM) accepts byte code in violation of language semantics; moreover, it cannot protect itself from tampering by other applications.

Virtual machine security

The hybrid and the hosted VM models in respectively, expose the entire system to the vulnerability of the host operating system; thus, we will not analyze these models. Our discussion of virtual machine security is restricted to the traditional system VM model, where the VMM controls access to the hardware. Virtual security services are typically provided by the VMM. Another alternative is to have a dedicated security services. The analysis of Xen and vBlades in Sections VM technology provides a stricter isolation of virtual machines from one another than the isolation of processes in a traditional operating system. Indeed, a VMM controls the execution of privileged operations and can thus enforce memory isolation as well as disk and network access. The VMMs are considerably less complex and better structured than traditional operating systems; thus, they are in a better position to respond to security attacks.

A major challenge is that a VMM sees only raw data regarding the state of a guest operating system, whereas security services typically operate at a higher logical level, e.g., at the level of a file rather than a disk block. A guest OS runs on simulated hardware and the VMM has access to the state of all virtual machines operating on the same hardware. The state of a guest virtual machine can be saved, restored, cloned, and encrypted by the VMM. Not only can replication ensure reliability, it can also support security, whereas cloning could be used to recognize a malicious application by testing it on a cloned system and observing whether it behaves normally.

We can also clone a running system and examine the effect of potentially dangerous applications. Another interesting possibility is to have the guest VM's files moved to a dedicated VM and thus, protect it from attacks; this is possible because inter-VM communication is faster than communication between two physical machines. Sophisticated attackers are able to fingerprint virtual machines and avoid VM honey pots designed to study the methods of attack. They can also attempt to access VM-logging files and thus recover sensitive data; such files have to be very carefully protected to prevent unauthorized access to cryptographic keys and other sensitive data.
The security group involved with the NIST project has identified the following VMM- and VM-based threats:

• VMM-based threats:

- Starvation of resources and denial of service for some VMs. Probable causes:
    - Badly configured resource limits for some VMs;
    - A rogue VM with the capability to bypass resource limits set in the VMM.
- VM side-channel attacks. Malicious attacks on one or more VMs by a rogue VM under the same VMM. Probable causes:
    - Lack of proper isolation of inter-VM traffic due to misconfiguration of the virtual network residing in the VMM
    - Limitation of packet inspection devices to handle high-speed traffic, e.g., video traffic
    - Presence of VM instances built from insecure VM images, e.g., a VM image having a guest OS without the latest patches.
- 3. Buffer overflows attacks. • VM-based threats:
    - Deployment of rogue or insecure VM. Unauthorized users may create insecure instances from images or may perform unauthorized administrative actions on existing VMs. Probable cause: improper configuration of access controls on VM administrative tasks such as instance creation, launching, suspension, reactivation, and so on.
    - Presence of insecure and tampered VM images in the VM image repository. Probable causes:
    - Lack of access control to the VM image repository;
    - Lack of mechanisms to verify the integrity of the images.

Security of virtualization

The relationship between virtualization and security is a complex one and has two distinct aspects: virtualization for security and the security of virtualization. we praised the virtues of virtualization. We also discussed two of the problems associated with virtual environments:

(a) The negative effect on performance due to the additional overhead; and

(b) The need for more powerful systems to run multiple virtual machines.

One of the most important virtues of virtualization is that the complete state of an operating system running under a virtual machine is captured by the VM. This state can be saved in a file and then the file can be copied and shared. There are several useful implications regarding this fact.

1. Ability to support the IaaS delivery model. In this model a user selects an image matching the local environment used by the application and then uploads and runs the application on the cloud using this image.

2. Increased reliability. An operating system with all the applications running under it can be replicated and switched to a hot standby in case of a system failure.

3. Straightforward mechanisms to implement resource management policies:

- To balance the load of a system, an OS and the applications running under it can be moved to another server when the load on the current server exceeds a high-water mark.
- To reduce power consumption, the load of lightly loaded servers can be moved to other servers and then these servers can be turned off or set on standby mode.

4. Improved intrusion prevention and detection. In a virtual environment a clone can look for known patterns in system activity and detect intrusion. The operator can switch to a hot standby when suspicious events are detected.

5. Secure logging and intrusion protection. Intrusion detection can be disabled and logging can be modified by an intruder when implemented at the OS level. When these services are implemented at the VMM/hypervisor layer, the services cannot be disabled or modified. In addition, the VMM may be able to log only events of interest for a post-attack analysis.

6. More efficient and flexible software testing. Instead of a very large number of dedicated systems running under different operating systems, different versions of each operating system, and different patches for each version, virtualization allows the multitude of OS instances to share a small number of physical systems. A first type of undesirable effects of virtualization leads to the diminished ability of an organization to manage its systems and track their status:

- The number of physical systems in the inventory of an organization is limited by cost, space, energy consumption, and human support. Creating a VM reduces ultimately to copying a file; therefore the explosion in the

number of VMs is a fact of life. The only limitation for the number of VMs is the amount of storage space available.

• In addition to quantity, there is also a qualitative aspect to the explosion in the number of VMs. Traditionally, organizations install and maintain the same version of system software. In a virtual environment such homogeneity cannot be enforced; thus, the number of different operating systems, their versions, and the patch status of each version will be very diverse, and this heterogeneity will tax the support team.

• Probably one of the most critical problems posed by virtualization is related to the software life cycle. The traditional assumption is that the software life cycle is a straight line, so patch management

New Virtualization System-Specific Attacks

VM jumping/guest hopping

- Attackers take advantage of hypervisor escape vulnerabilities to "jump" from one VM to another

- **VM attacks**

  –Attacks during deployment and duplication
  –Deletion of virtual images
  –Attacks on control of virtual machines
  –Code/file injection into virtualization file structure

- **VM migration**

  –VM migration is transfer of guest OS from one physical server to another with little or no downtime
  – Implemented by several virtualization products
  –Provides high availability and dynamic load balancing

- **VM migration attack**

  – If migration protocol is unencrypted, susceptible to man-in-the-middle attack
  – In default configuration, Xen Motion is susceptible (no encryption)
  –VMware's VMotion system supports encryption
  –Proof-of-concept developed by John Oberheide at the Univ. of Michigan

- **Management server attacks**

  –Exploit management console vulnerabilities that divulge password information
  –Exploit management console vulnerabilities to gain access to management server
  –Exploit vulnerabilities that allow local management server users to gain elevated privileges

- **Administrative VM attacks – exploit vulnerabilities to:**

  –Cause a denial of service by halting the system
  –Cause a denial of service by crashing the administrative VM
  –Obtain passwords that are stored in clear text
  –Exploit buffer overflows in exposed services to execute arbitrary code
  –Exploit vulnerable services to gain elevated privileges
  –Bypass authentication

- **Guest VM attacks – exploit vulnerabilities to:**

  –Gain elevated privileges
  –Crash the virtual machine
  –Truncate arbitrary files on the system
  –Execute arbitrary code with elevated privileges

- **Hypervisor attacks – exploit vulnerabilities to:**

  –Cause the hypervisor to crash
  –Escape from one guest VM to another

- **Hyper jacking**

  –Consists of installing a rogue hypervisor
    • Force kernel to be paged out by allocating large amounts of memory
    • Take action to cause driver to be executed
    • Shell code downloads the rest of the malware
    • Host OS is migrated to run in a virtual machine
  –Has been demonstrated for taking control of Host OS
  –Hyper jacking of hypervisors may be possible, but not yet demonstrated