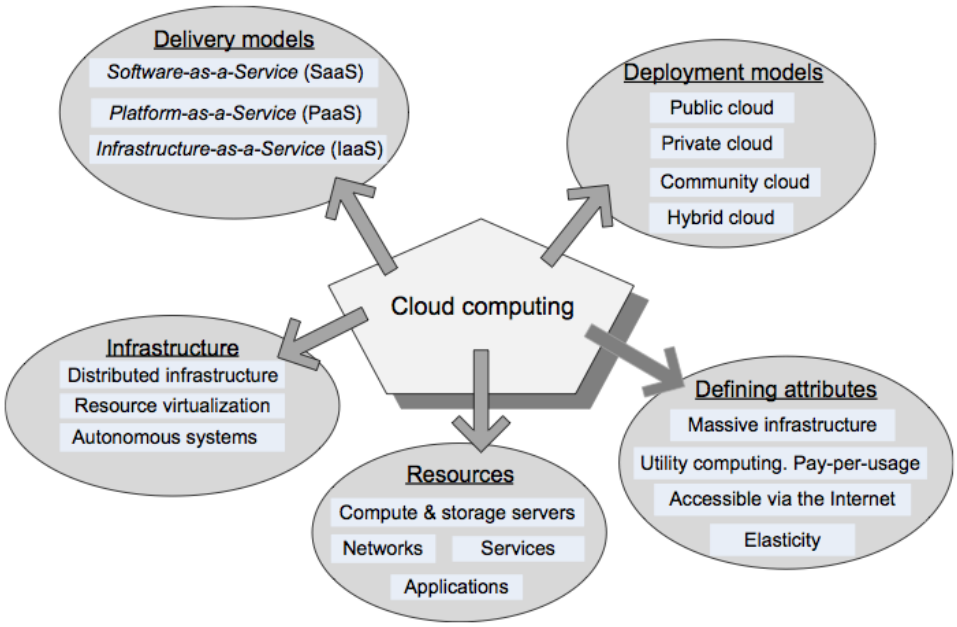
# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 3: Cloud Characteristics
# Module No: CS/CC/3
# Quadrant 1 —e-text

## 1. Introduction

Most of us use cloud computing all day long without even realizing it. For example, when one types a query into Google's search box, the words typed are swiftly shuttled over the Internet to one of Google's data centers, which bring out the desired results. Cloud computing is the reason why all theseare possible. In our last module we discussed why cloud computing is actually the realization of long-desired utility computing. We also looked at various definitions of cloud and talked about the resources needed and data centers where these resources are housed. In this module we continue our discussion on the various components that together with the necessary stitching technologies deliver the required effect expected in a cloud. In Figure 1 below we reproduce the diagram we have introduced in the earlier module. It shows various components of cloud computing. We have already seen that the place where the combination of all the resources is found is called a data center.




Curtsey: Dan C. Marinescue [1]

**Figure 1: Cloud Computing Components**

It is said that cloud computing is a natural evolution from data centers. Data centers are the core of a cloud environment. Actually, the concept of data centers is quite old. While the early, large, roomful computers of the 50s can be thought of as a precursor to today's data centers, the boom of real data centers came during the late 90s to early 2000 when, due to a sudden increase in the information technology (IT) requirements, companies needed a large infrastructure to deploy software systems quickly. Virtualized data centers came about that time. However, with companies developing their own infrastructure for various reasons, the boom gave way to proprietary data centers. But when these proprietary data centers are no longer economical, came the cloud with data centers as the basic infrastructure.

With this understanding, we proceed to the next aspect, *i.e.*, the infrastructural requirements of cloud.

**2. Learning Outcome**

In this third module of the course Cloud Computing, understanding the fundamental concepts of various parts of cloud computing will be provided. At the end of this module, students will be able to:

1. Explain the features of the infrastructural requirement of cloud.
2. Understand and appreciate the various characteristics that differentiate cloud from other paradigms.
3. Learn about the negative effects of cloud that exist today and how to mitigate these.
4. Explore the various deployment models of cloud.
5. 
6. Understand and appreciate the different entities that form cloud.
7. Gather knowledge about various components of resource requirements in a cloud computing facility with a focus about data centers.
8. Overview
9. Explain
10. Features
11. Integrate the learning with

Having seen the basic of what a cloud computing is all about and the resources needed to implement a cloud, let us take a look at the other aspects of cloud.

**3.Cloud Infrastructure**

Cloud being massive in scale, its infrastructure also is expected to be large. Indeed, as a distributed system, one would expect cloud to be essentially distributed in nature. Thus we identify the first point of cloud infrastructure being distributed across geographical regions. The data centers that house the essential components of cloud are built in different regions for various reasons and hence they are distributed. The second point of infrastructure is about virtualization. The infrastructure is distributed implies that whatever service is being provided to a user may be located in any of the multiple data centers used by the service provider. The

users should be able to acquire the services from any remote location wherever they are and this perhaps is the most important characteristic of cloud computing. This is why cloud computing is part of distributed system.

The success of cloud lies in an underlying technology called virtualization. Many of the essential characteristics of cloud come from the fact that the resources are virtualized. Typically, the data centers are virtualized to maximize resource sharing and hence resource utilization, thereby rendering the whole environment profitable for both service providers as well as service consumers. The effect of virtualization is also that we do not expose the actual location or the details of the underlying hardware to the user. A layer of indirection on top of the hardware implements virtualization and this allows flexible use of the underlying resources. Since virtualization is a very important part of cloud computing, we will talk about virtualization in detail in later modules. We will study virtualization in detail later. For the time being it is sufficient to mention that the ability of cloud to allow many users use the resources efficiently comes from the concept of virtualization.

The third point of infrastructure is the autonomous system. It is obvious that the large scale and the flawless resource provisioning that is expected of cloud cannot be achieved using manual control. Thus cloud is an autonomous environment. The point is that without human intervention the system should be able to continue on its own and the mechanisms of the systems are such that while each one can independently work, collaboration is also possible. This implies that such autonomous systems should be able to work independently as well as together to achieve a better goal than a single autonomous system**.**

In the next subsection, we look at the essential characteristics of cloud.With the proliferation of cloud, various opinions on cloud were developed. However, NIST, in their document, put forward, for the first time, the essential characteristics that are required in a system to be called a cloud. Hence we start our next section with this.

## 4. Essential Characteristics of a Cloud Environment

There various ways of looking at the various attributes that define what is cloud. In this section, we will look at two such ways, the NIST way and the Gartner way.

### 4.1. The NIST Essential Attributes.

The five essential attributes of a cloud computing environment according to the NISTare as follows:

1. On Demand Self Service
2. Broad Network Access
3. Resource Pooling
4. Rapid Elasticity
5. Measured Service

Let us understand each of these.

### 4.1.1On Demand Self Service

Cloud services are available on-demand and essentially used on a "pay-as-you go" model or on the basis of subscription. Thus, user pays for the services that are consumed just like paying for electricity. However, with the advent and proliferation of different modes of deployment, usage of cloud may also be free or may be compensated via other means, like advertisements on websites.

The other important part of this characteristic is self service. By self service, we mean that a consumer can unilaterally acquire resources without any human intervention from either side. These resources include computing capabilities, server time as well as storage over the network. This is highly advantageous, as the user does not have to go through a lengthy procedure to access resources. Instead, a simple and automated request mechanism to a service provider's portal allows a customer to acquire and use the necessary amount of computing, storage, software, process, or other resources and services.

### 4.1.2. Broad Network Access

All the capabilities and resources hosted by a service provider's cloud network are available to the users. They are accessed through mechanisms developed and standardized by the service providerthrough heterogeneous client platforms including PCs, mobile phones, tablets etc. accessed over the Internet. This provides a very important characteristic and actually catapults the cloud advantage. During the mainframe era, resources were expensive and scarce. To reduce the usage and hence save expense, various criteria such as some priority or the importance of the work would be used. However, with advents of technologies and more competition, the resources are more easily available. Further, the access to the network has improved with greater bandwidth being available making the environment more scalable. In the cloud computing era, it has become imperative that the network be unrestrained by the use of broad access.

### 4.1.3. Resource Pooling

The essence of cloud computing is perhaps the fact that there is a pool of shared resources. It is not possible to improve scalability unless there is an improvement on the resource pooling.If resources like computing, networks, and even storage are not put to service in a pool, a service provider may have to operate across multiple independent resources with few or no interconnections to cater to the customers' requests, thereby losing the primary benefit provided in cloud.Therefore, resource pooling helps the providers to combine in a pool all the resources and be able to satisfy the needs of many consumers in such a way that the consumer is not able to understand or control over the location from where the resources are being served.

However, even in this situation, users are able to specify if they need some location specific service such as country, state or even data centers. Also the pooled resources help provide service using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.The concept of multi-tenancy directly stems from this characteristic. Let us understand multi-tenancy.

#### 4.1.3.1. Multitenant Environment

Let us understand the concept of multitenant environment. In such a situation,adjacent resources may be used by multiple customers at the same time in a sharable manner. This model extends to the sharing of both software as well as hardware resources. In a cloud environment, resources are dynamically assigned and reassigned according to consumer requirements. It drives cost efficiency and improves utilization. However, this sharing also leads to an inherent increase in operational expenditures, althoughthe additional expenditure gets compensated by the benefits thereof. Let us understand this using a practical example.

Let us consider an apartment building versus an independent house. An apartment building has certain resources (like the staircase, common passage ways, swimming pool, gymnasium etc.), which are not 'owned' by any one resident, but are sharedamong all the residents. As a result, the facilities are utilized to the maximum possible extent, more so than they would be in a private house. However, as a byproduct of this increased utilization, maintenance costs also rises.

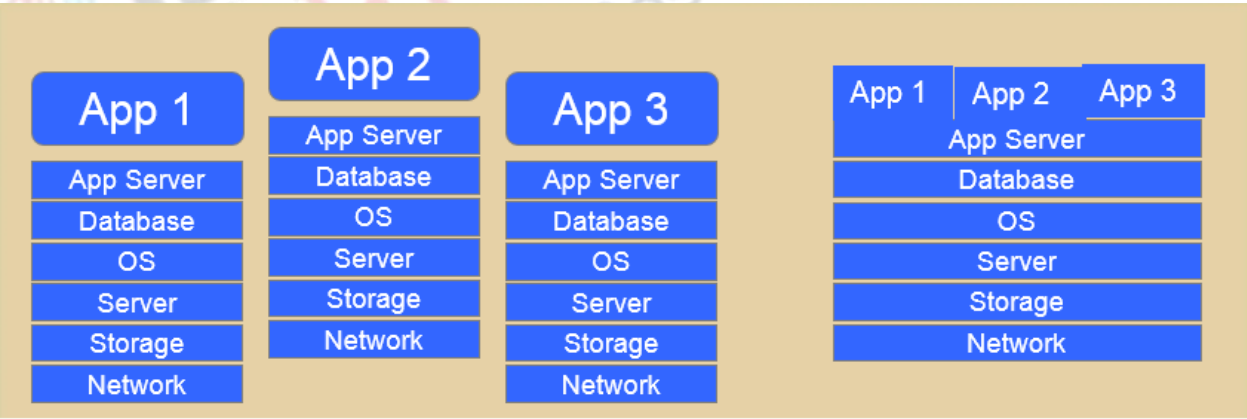



**Figure 2: Single Tenancy and Multi Tenancy**

The Figure 2 above displays the architecture of typical single and multi tenant systems. Single tenancy gives each customer a dedicated software track. Configuration, monitoring, upgrades, security updates, patches, tuning and disaster recovery are all focused on servicing one user at a time.

On the other hand, in a multi-tenant environment, all applications run in a single logical environment. Obviously, it is faster, more secure, more available, automatically upgraded and maintained.

### 2.1.4. Rapid Elasticity

This characteristic ensures that capabilities of the data centers of the provider be elastically provisioned and released, perhaps automatically, to scale rapidly outward and inward as per the demand. This means that user should feel that any resource could be appropriated in any quantity at any time.The requirement of resources to be elastic stems from a need to provide resources to users at improved speeds at the same timereducing associated costs. A cloud-based architecture should be able to provisionany amount of resources at a fast rate so as to appear unlimited and instantaneous to the consumer. This leads to the illusion of infinite computing resources in cloud.

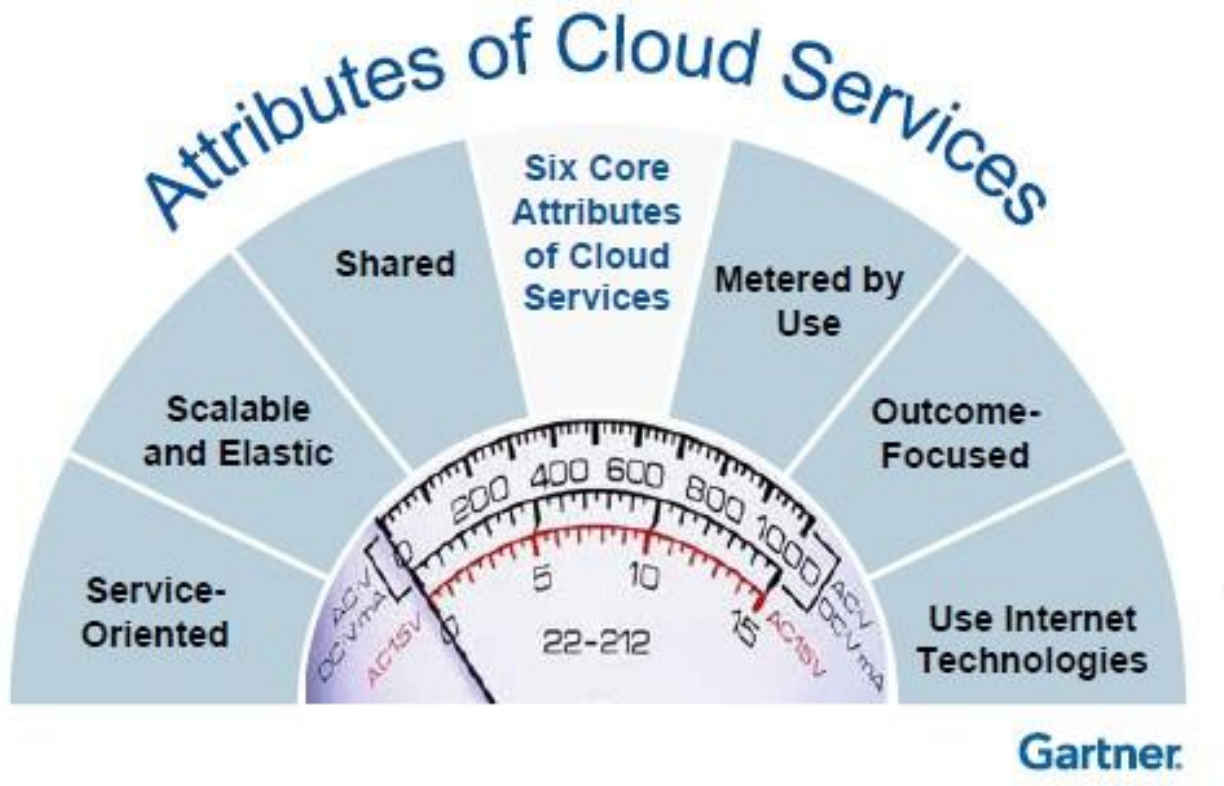
### 4.1.5. Measured Service

While resource pooling is a key aspect of cloud computing from the implementation perspective, measured service is a key aspect of cloud from the financial perspective. Measured service implies that usage of the pooled resources is monitored and reported, providing a mechanism to capture the amount of consumption and hence their associated costs. In a way, the usage of resources is controlled by leveraging a metering capability on them. This is helpful from the perspective of both the service providers as well as the consumers. This monitoring, controlling and reporting of resource usage provides a transparency for both the provider as well as the consumer. This model reflects the pay-per-use system that is employed typically in electricity and other day-to-day utility services.

### 4.2. Gartner'sCore Attributes

While the above are the characteristics as per NIST, Gartner, an American IT research and advisorycompany providing technology related insights, envision cloud services to have six core attributes as shown in Figure 3 below. These are:

- Service-Orientated
- Scalable and Elastic
- Shared
- Metered by Use
- Outcome-Focused
- Use Internet Technologies




Curtsey: Gartner[2]

**Figure 3: Gartner's Core Attributes of Cloud Services**

#### 4.2.1. Service Oriented

In a cloud environment, the implementation details are hidden from the customers using well-defined service interfaces. This also enables a completely automated response by the provider to the consumer of the service. Service orientation is also implemented in the sense that the consumer gets what is needed from the provider. Services are not created as per the technological development, rather these technologies are tailored to meet thedemands of the consumers making service to be supreme in the world of cloud.

#### 4.2.2. Scalable And Elastic

As mentioned before, cloud resources and services must be created in such a way that these can easily be scaled up or down to meet the requirements of the customers. While scalability is a feature of the underlying infrastructure and software platforms,elasticity is a trait of a shared resource pool.With scalability comes the option of upward scaling where new services may be added on demand and downward scaling whereby resources/services may be removed as needed. The cloud computing environment thus enables scaling in both directions in an automated fashion.

#### 4.2.3. Shared

Cloud resources are shared by multiple users of the service, in order to achieve maximum efficiency.The details of the sharing are usually unknown to the consumers, so each feels as if she is getting exclusive access to the entire set of resources. This model of having a common pool of virtualized and shared resources enables maximal utilization of all available resources, allowing resources to serve different purposes for different users simultaneously.

#### 4.2.4. Metered By Use

Usage metrics are applied to track the usage of services by customers. The service provider is expected to possess a usage accounting model to measure and keep track of the amount of services used by consumers each time resources are consumed. Based on this information, various pricing plans and models can be created and offered to the user. Any unit can be used to measure the usage, for example it can be the number of hours of execution, or it can be the amount of data that is transferred or any other metric. In certain cases, it can even be free, for example Google apps or Google App Engine.

#### 4.2.5. Outcome Focused

As the term suggests, the usage of resources in a cloud environment must be oriented towards result or the outcome of the usage. This means that the resources are aimed at providing maximal utility to the users, tailored to suit their needs.

#### 4.2.6. Usage Of Internet Technologies

Cloud services are delivered using Internet protocols, formats and identifiers such as URLs, HTTP, IP and representational state transfer web-oriented architecture. Thus Internet connectivity and usage is a prerequisite of cloud use.

### 5. Negative Side Of Cloud

While cloud eliminates the need for up-front financial commitment, is based on a pay-as-you-go approach and has attracted new users for existing applications as well as new applicationsthrough the technological breakthroughs that have made cloud computing feasible, there are still major obstacles for this new technology.A few of the most obvious obstacles are discussed here.

#### 5.1 Availability of Service.

While consumers are supposed to completely depend on cloud for all their needs, what

happens when the service provider cannot deliver? Can a large company such as General Motors move its complete IT needs to the cloud and have assurances that its activity will not be negatively affected by cloud overload, power outage or any other problem? A partial answer to this question is provided by service-level agreements (SLAs). These are agreements between two parties, a service consumer and the corresponding service provider that should be implemented and respected by both the parties. While for the time being it is sufficient to know that SLAs may be an answer to provide some guarantee to a user, we will discuss the various implementation issues of SLAs at length later. However, even through SLAs, it is impossible for a service provider to guarantee 100% availability of its services at all time. A temporary fix of thisnegative economical implication is *overprovisioning* by the service provider, that is, the service provider should acquire enough resources to satisfy the largest projected demand and must have enough back up facility. However, neither of the solutions, SLA and overprovisioning, is devoid of problems.

### 5.2. Vendor Lock-In

Once a customer stores her data and runs the applications in a cloud solution provided by a specific service provider, it is hard to move to another service provider, mainly due to the lack of standardization among various service providers. Typically the format or the method of data storage, computation execution, results presentation are proprietary and not be same or even similar amongst various service providers making the migration from one provider to the next a difficult, even an impossible, task. The standardization efforts at National Institute of Standards and Technology (NIST) attempt to address this problem.

### 5.3. Data Confidentiality And Auditability

The data of the client are out of the control of the client and under complete control of the service provider -- either the data is stored in the cloud, which is a server away from the client's premise or data has to be supplied during computation being executed in a cloud server. This is indeed a serious problem that has led to the development of a private cloud that we are going to discuss next. However, this is a serious problem in cloud being accepted completely by the industry.

### 5.4. Data Transfer Bottlenecks

Cloud communication is Internet based and many applications are data-intensive. Hence a basic need is to transfer large amount of data for different purpose. However, transferring data over the Internet may be very slow. For example, transferring 1 TB of data on a 1 Mbps network takes approximately 10 days making large data transfer a costly affair. A solution therefore is found in incorporating the strategy of storing the data as close as possible to the site where it is needed. Very high-speed networks will alleviate this problem permanently in the future.

Having seen the possible negative aspects of cloud, let us turn our attention to the different deployment models offered in cloud.

## 6. Types Of Cloud

It is often said that the term *cloud* is overloaded, since it covers infrastructures of different sizes, with different management and different user populations. Let us turn our attention to the various types of clouds that make this diversity possible. The deployment of cloud facilities are

generally divided into four categories:

1. Public Cloud
2. Private Cloud
3. Hybrid Cloud
4. Community Cloud

### 6.1. Public Cloud

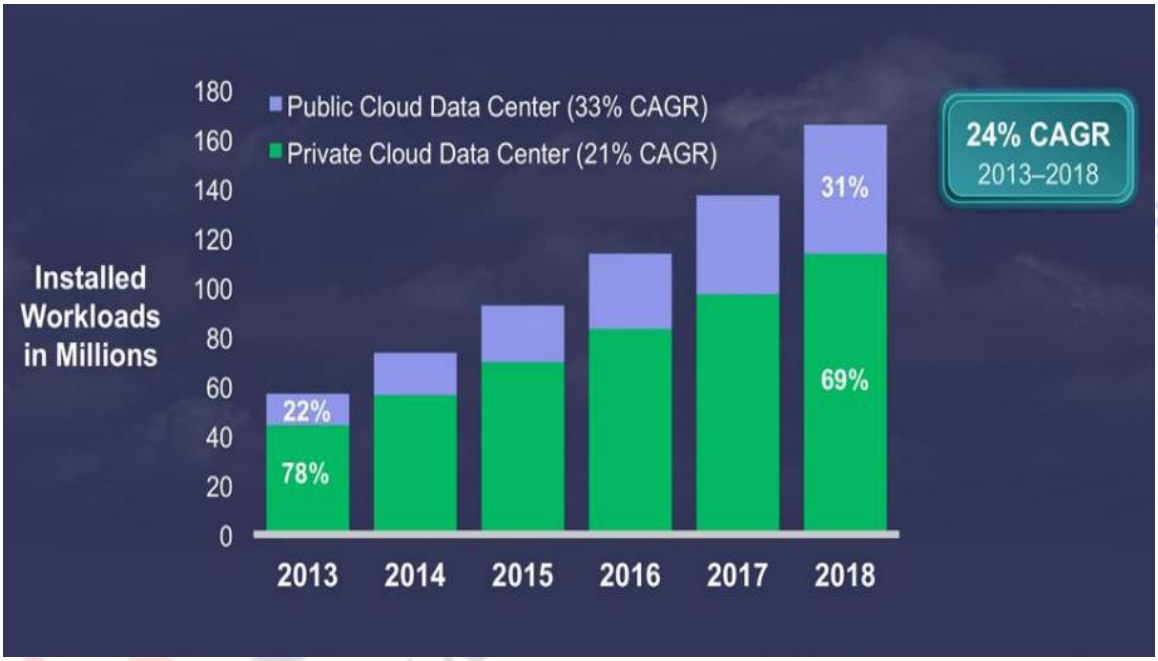
A public cloud is the most common and well known type of cloud. Any organization may own a public cloud. Alternatively, a collaboration of business houses and/or academmic organizations may jointly own such a cloud. Operating rules and managing the cloud would belong to these owners, and they will set guidelines for the general public or any other group belonging to large industry group. The cloud services are expanded over a network open for public use, like the Internet. Public cloud services may be free or offered on some payment model (such as pay-per-use). The public cloud model has a number of advantages, such as cost effectiveness and better reliability attributed to the sheer number of servers and datacentres involved. Examples of public clouds include Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google AppEngine and Microsoft's Windows Azure Services Platform.

### 6.2. Private Cloud

A private cloud is the one that is offered to a small set of specific users. Just like public cloud, private cloud also may be owned by multiple organizations and managed by them. But unlike public cloud, the rules of a private cloud are set by the organzation to whom the cloud belongs. Typically, the services of a private cloud are made available to workers of the owner organizaton or the industry group. A private cloud has restricted access to its resources. While it may be owned, managed, and operated by the organization, a third party, or some combination of them may also be used to create, deploy and manage a private cloud. In a private cloud, the services are accessed through a secure network connection, similar to intranets. That is, access to these pooled computing resources is controlled and they are made available only to specific user and not to anyone who is ready to pay for it. Although a private cloud is capable of offering the same features and benefits of public cloud systems, many problems idetified for publiccloud can be avoided in the private cloud. For example, in a private cloud since the provider is the same organization as the user, the data security is not threatened. Private clouds are precursors of public clouds. In fact, many of the current popular public clouds such as Amazon and Google were originally used as private clouds by the owner companies for other lines of business. If an organisation has enough users and capacity, a private cloud can behave much like a scaled down version of a public cloud.

While private and public clouds are the two sides of the same coin, the concept of private and public cloud differs from each other in the sense that they are meant for a specific set of users in private while they are meant for general public for a public cloud. There is more workload in private cloud than in public cloud as shown in Figure 4 below by CISCO. CISCO also made a prediction of how the private cloud would grow with respect to public private cloud as shown in the figure. Public cloud, as indicated by the workloads growth, is growing faster than the private cloud and there will be a notable shift to public cloud services. As the business sensitivity to costs associated with dedicated IT resources grows along with demand for agility, it is observed that public cloud is enjoying a greater adoption by the businesses. This is perhaps attributed to

the recent attention that public cloud security enjoys and the attempts made to strengthen it. The market adoption of the public cloud is increasing as it commands a higher level of security, thanks to all the efforts. However, the mission-critical work is still not being transferred to a public cloud, the reason for which is quite obvious. This is apparent from the figure. The share of the cloud workloads was 78% for the private cloud and only 22% for the public cloud in the year 2013. However, these numbers change drastically in the projection in 2018, where the share of the cloud workloads is predicted to be 69% for the private cloud and only 31% for the public cloud. Therefore the compound annual growth rate (CAGR)of the public cloud workloads are going to grow at 33-percent CAGR from 2013 to 2018 while private cloud workloads will grow at a slower pace of 21-percent CAGR from 2013 to 2018.




Curtsey: CISCO Index Report[3]

**Figure 4: Pubic versus Private Cloud Growth**

### 6.3. Hybrid Cloud

As the name suggests, a hybrid cloud model is a fusion of public and private clouds as shown in Figure 5 below.

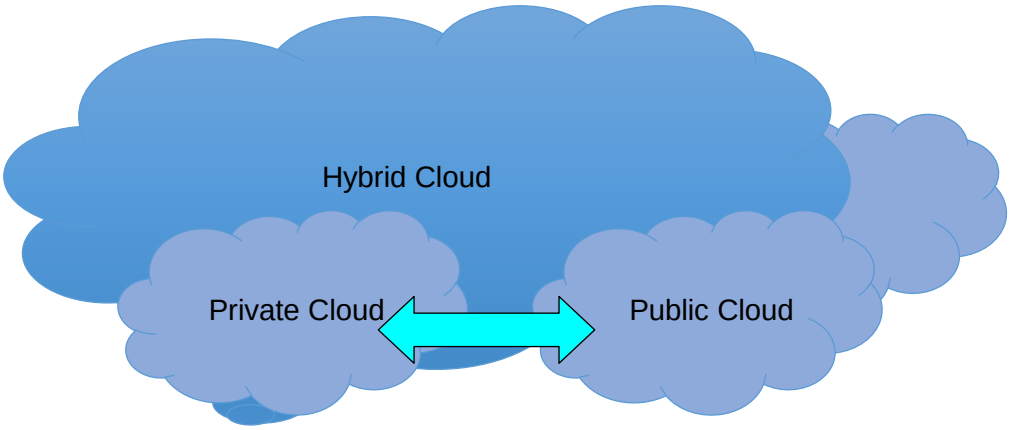
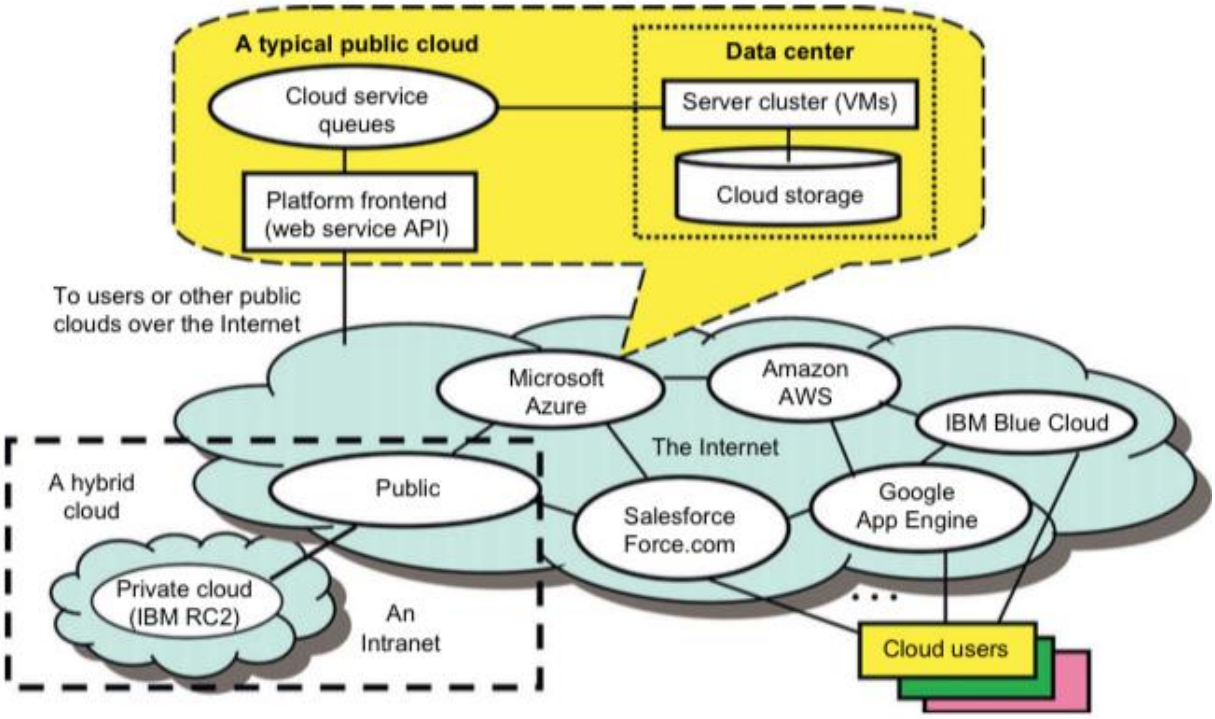



**Figure 5: Hybrid Cloud**

Some enterprises may adopt the hybrid approach to cloud. Hybrid cloud environment is one where an enterprise maintains the data and computation partially in-house while some part is out-sourced to an external provider. Cloud bursting is a typical example of hybrid cloud. Here daily computing requirements are handled by a private cloud, but when there is a sudden additional demand, a public cloud is accessed.Thus, hybrid cloud provides greater flexibility to businesses along with more data deployment options.Here private and public clouds are used for computing needs with variable costs as is applicable.

Figure 3.6 below shows a combined private, public and hybrid cloud. As many clouds are generated by commercial providers or by enterprises in a distributed manner, they will have to be interconnected over the Internet to achieve scalable and efficient computing services. Commercial cloud providers such as Amazon, Google, and Microsoft created their platforms to be distributed geographically. This distribution is partially attributed to fault tolerance, response latency reduction, and even legal reasons. Intranet-based private clouds are linked to public clouds to get additional resources in a hybrid cloud.

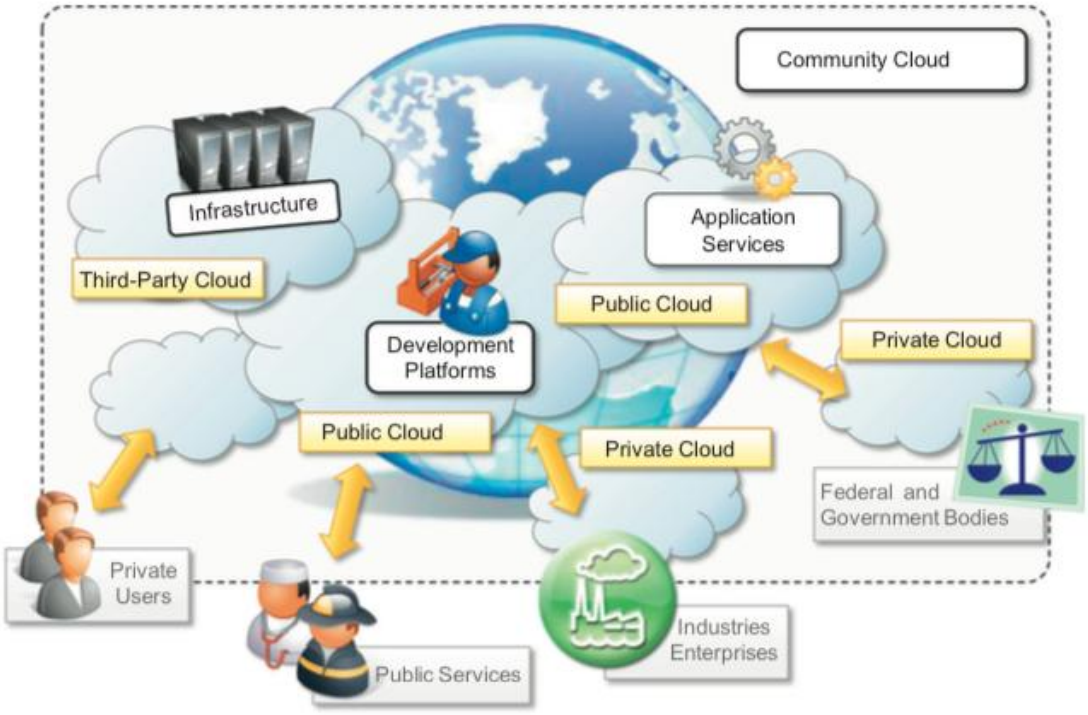



Curtsey: Kai Hwang [4]

**Figure 6: Public, Private and Hybrid Cloud**

### 6.4. Community Cloud

Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector as shown in Figure 7 below.




Curtsey: RajkumarBuyya et al.[5]

**Figure 7: Community Cloud**

NIST characterizes community clouds as follows:

> "The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises."

In a community cloud, the cloud infrastructure is provisioned for exclusive use by a specific community of consumers. Thesemaybe from certain organizations, government bodies, industries, or even simple users, that have some shared computing concerns. Such common concern may be related to specific performance requirements, such as hosting applications or may be regulatory compliance, or may even be a common goal of achieving specific targets etc. A community cloud aims at acheiving all benefits of a public cloud, such as resource pooling and multi tenacity, with the added levels of security, privacy and a policy compliance typically enjoyed by a private cloud. The costs of a community cloud tend to be lower than that of a public cloud as the services are spread over fewer users.Community clouds are different from public clouds, which serve a multitude of users with different needs. Community clouds are also different from private clouds, where the services are generally delivered within the institution that owns the cloud.

**7. Summary**

We started this module with a recapitulation of our earlier module, where we started discussing the various aspects expected in cloud. Then we proceded with the next aspect, that is the infrastructural requirements of cloud. Next we learnt about the various characteristics of cloud computing from two perspectives, viz., the NIST and the Gartner. The essential characteristics proposed by the National Institute of Standards and Technologies (NIST) and the core attributes proposed by Gartner, while essentially being similar, has certain differences and this section highlights these differences. At this juncture, it is important to note the negative side of this excellent phenomena called cloud and our next section discussed these. We concluded the section by elaborating the various deployment models used in cloud envirinment.

**Referrences**


1. Dan C. Marinescu, “Cloud Computing, Theory and Practice”, 1st Edition, Morgan Kaufmann, 2013.
2. http://www.gartner.com
3. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf
4. http://www.ciscopress.com/articles/article.asp?p=1925617&seqNum=3
5. Kai Hwang, Jack Dongarra, Geoffrey C., “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things”, Morgan Kaufmann, 2011.
6. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, “Mastering Cloud Computing Foundations and Applications Programming”, Elsevier Morgan Kaufmann, 2013.
7. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
8. http://en.wikipedia.org/
9.

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 38: Cloud Security
# Module No: CS/CC/38
# Quadrant 1 — e-text

## 1. Introduction

Cloud has two major entities, the service providers and the service consumers. Unfortunately, this space is also one where many users would find their archrivals. Since cloud provides an ambience where all users are expected to share resources, a greater environment of mistrust is generated. One, and perhaps the only, method of ensuring harmony and collaboration is to ensure perfect security of data, as is done using security. In this section, we will explore all the security issues that arise in various parts of the cloud. We will also look at the various standards that are available in a cloud environment.

## 2. Learning Outcome

At the end of this module, students will be able to:

1. Understand what is meant by security in a cloud environment.
2. Recall various perspectives of security in cloud.
3. Inspect the critical aspects of cloud security.
4. Investigate the requirements of security in cloud.
5. Understand the issues to see why we need to focus more on the security aspects in cloud environment.
6. Discuss about the various standards used in cloud.
7. Understand what is meant by security as a service.

## 3. Security and Cloud

According to Wikipedia, "information security, sometimes shortened to InfoSec, is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. It is a general term that can be used regardless of the form the data may take (e.g. electronic, physical)" [7]. Therefore, security is the prevention of or protection against the access to all the resources, viz., data, hardware, software, network etc. by unauthorized recipients; intentional and unauthorized destruction or alteration of information and protection against all the vulnerabilities that a system may exhibit.

### *3.1. Cloud Security*

Since cloud is all about resources and resource sharing, security is expected to be a large part of learning cloud. However, there are certain characteristics that make cloud security much more challenging than a normal computer security or information security.

The cloud service models along with the technologies used to enable cloud services contribute greatly to increase the security concerns of a cloud computing environment. By these, cloud presents different types and kinds of risks to an organization than traditional IT solutions.

Further, in cloud, both the providers of the services as well as the consumers of these services are to be responsible to ensure security. Since parts of the environment are hidden from the consumers and are accessible only to the providers, a greater responsibility lies on the providers. However, even the consumers have to carry a great burden in terms of ensuring security.

Let us take examples of two big companies, viz., Salesforce.com and Amazon. When a consumer is using Amazon's AWS EC2 infrastructure as a service offering, the provider is responsible to have secured physical data centers, secured environment and secured hypervisor. However, all other parts of the system that the consumer is going to use, viz., the operating system, applications, and data used in the virtual machines are the responsibility of the consumer. On the other hand, if we take the example of a consumer using Salesforce.com's resource management (CRM) SaaS, the provider here is responsible for the security of the whole stack, starting from the physical and environmental security, the infrastructure security, the applications security as well as the data security. In this case, the consumer's responsibility is minimal.

Therefore, in IaaS, the responsibility is divided among the consumer and the provider in that the provider is responsible for securing the underlying infrastructure and abstraction layers and the consumer has to ensure the security of the rest of the stack. In SaaS environments, on the other hand, almost all the responsibility lies on the providers and consumers enjoys as much security as they have paid for, while the providers have to ensure the security implementations. The security controls and their scope are negotiated into the contracts for service; service levels, privacy, and compliance are all issues to be dealt with legally in contracts. PaaS offers a balance somewhere in between, where securing the platform falls onto the provider, but both securing the applications developed against the platform and developing them securely, belong to the consumer.

Another important aspect of security is auditing. If a service or a cloud environment is auditable, many of the existing problems would be resolved. Hence we must understand what is auditing and how it can be made possible.

All these show how cloud security concerns need special attention. Let us now look at the different perspectives of cloud security.

**4. Cloud Security Perspectives**

Given the complexity of the topic, cloud security can be looked from a varied range of perspectives. In this module, we will look at the following perspectives:

i) Reasons for security problems in cloud
ii) Critical aspects of cloud security

iii) Security requirements
iv) Security issues
v) Standards

### *4.1. Reasons for Security Problem in Cloud*

There are a few concepts that are responsible for making the cloud more vulnerable to threats and attacks that are not present in the normal IT system. These are

- Virtualization
- Multi tenancy
- Loss of control on Data
- Lack of trust
- Increased attack surface
- Difficulty in auditing and forensics

#### *4.1.1. Virtualization*

Virtualization is perhaps the most important magic in cloud that makes cloud efficient in terms of shared resources. Cloud providers must make the most in virtualizing and sharing the resources to maximize the resource utilization on one hand and to create well-defined isolation across user VMs on the other. However, the control is with the hypervisor and this now becomes the point of weakness in a virtualized world. The hypervisor layer becomes an easy target compromising which an attacker can access the instances. BLUEPILL, SubVirt and DKSM are attack examples on virtual layer. Through attacks an attacker is able to modify the hypervisor and be able to access the host resources. Similarly, memory access to the host can be obtained by violating the read/write accesses.

#### *4.1.2. Multi-Tenancy*

Multi-tenancy is the capability of running multiple instances on top of the same platform. Here, multiple users share the platform concurrently, while being isolated from each other through VM instances. Therefore, all the tenants (VM users) accessing a VM instance are available on the shared platform. Hence the name multi-tenancy. As a typical real-life example, we talk of a multi-storied apartment building where the same ground space is shared by multiple tenants living on various floors of the building.

In all service levels viz., IaaS, PaaS and SaaS, multi-tenancy forces different applications share the same underlying platforms at different layers. And in all the cases, the customer's data may be stored in the same physical location. It is comparatively easy for an attacker to gain control of the data of another application in the form of *co-location*, *co-residence*, or *co-tenancy* attacks through multi-tenancy feature of cloud. Further, it is easy to achieve a Denial of Service attack due to the underlying shared platform.

As in the earlier virtualization discussion, in multi-tenant architecture as well, virtualization is used to offer shared on-demand services. Hence vulnerabilities in a hypervisor may increase possibilities of such attacks as well. Possible solutions to this problem lie in implementing appropriate isolation of VM and also SLAs. Further having stronger authentication and access control mechanisms also is needed.

*4.1.3. Loss of Control on Data*

Cloud promotes outsourcing. The industries is thriving in this outsource business model. According to this model, the responsibilities of acquiring resources and using these resources be it storage or computing requirements, are delegated to contracted third-party service providers, with the customer concentrating on finer parts of the work. This way, a customer saves her capital expenditure on the physical resources and the operational expenditure of maintaining the equipment, running the latest software or any other operational responsibilities as well. Hence both CapEx (capital expenditure) and OpEx (operational expenditure) of customers are taken care of. In fact, this is one of the most important strategies that have rendered cloud popularity. However, the glitch in this scheme is that data, perhaps the most important asset of business, must also be outsourced for this scheme to work. Hence, users, that is the business houses have to part with their data, that will now be maintained by the third party in an off-site location, outside the premises of the organization to whom the data belongs. Therefore, cloud, while providing innumerous facilities, renders customers to lose control on their data. Hence there must be trust between the service providers and consumers for the model to work.

*4.1.4. Lack of Trust*

In cloud, the customer infrastructure, data and compute all located off-premise and are managed by a third-party entity. This is an aspect in cloud that is special. It is important for the consumer to be able to trust a provider. However, trust is a concept rather than a measurable element since it is more subjective than objective. Measuring trust must involve several factors that make decision-making and evaluating a decision harder.

Hence, security management must provide any and all quantifiable measures that help the consumer gain trust about a provider and the provider, on their end, must enable such factors.

From this perspective, trustworthy systems have been proposed wherein systems must always perform as expected, even under the influence of hostile disruption. Such trustworthiness combines reliability and security that uses security measures that are visible and measurable at any point in time, thereby allowing consumers to gain trust on a certain provider by testing the provider at unknown frequency and on different aspects.

Since trust is a humane aspect, rather than a machine aspect, a mixture of people-to-machine, people-to-people, and machine-to-machine interactions must be provided in such measures.

*4.1.5. Increased Attack Surface*

An attack surface is vulnerability in a system that malicious users may utilize and exploit. Cloud actually has an increased attack surface since for an organization the data is stored outside the organization with an unknown entity being in charge. Even all the computations on the stored data are typically being done by this entity. Also all the communications during, before and after storing data in cloud take place over the Internet, thereby making cloud more vulnerable through the communication links which are public/insecure. The attack surface further increases since cloud forces consumers to share the infrastructure.

*4.1.6. Difficulty in Auditing and Forensics*

Client does not have direct access to any resources since everything in cloud is virtualized. Hence it is difficult to have a monitoring mechanism. Hence auditing can be done only by the service provider and we have to ask how trust-worthy that is. Further, in case of any problem occurring in a cloud environment involving clients' data and/or computation, the natural recourse is digital forensics. It is a branch of forensic science that is involved in the recovery and investigation of material found in digital devices in case of a computer crime.

### *4.2. Critical Aspects of Cloud Security*

Many industry as well as research bodies have investigated to identify the most critical aspects in the security of cloud. NIST had proposed a large set of items in this category [5]. Below we discuss the most critical ones.

i. Data breaches
ii. Data Loss
iii. Account or Service Traffic Hijacking
iv. Insecure Interfaces and APIs
v. Denial of Service
vi. Malicious Insiders
vii. Insufficient Due Diligence
viii. Shared Technology Vulnerabilities
ix. Loss of Governance
x. Lock-in
xi. Insecure or Incomplete Data Deletion
xii. Availability Chain

i) Data Breaches: When an attacker, even a business rival, obtains access to an organization's sensitive internal data, due to attacks on the virtual machines, data breaches occur.
ii) Data Loss: Many a time, there may be an unintentional error on the part of the storage by which important data get deleted. Other possible reasons for such deletions are loss of encryption key. Natural disaster such as flood, fire etc. may also cause data loss.
iii) Account or Service Traffic Hijacking: there are various ways by which an attacker can access the credentials of the customer such as the identities, social security numbers etc. through software vulnerabilities in the cloud environment. This is known as traffic hijacking.
iv) Insecure Interfaces and APIs: If special care is not taken, then the interfaces used by a service provider may be exploited by an attacker to create an opportunity and this increases the risk of cloud usage.
v) Denial of Service: When a valid customer is unable to access her data and other resources in the cloud due to false artificially created traffic, a Denial of Service (DoS) attack is created. The ultimate form of this is a slowdown to shutting down of the servers keeping authentic customers off the service.
vi) Malicious Insiders: Attacks on the consumer resources always need not come from outside attackers. The system and consumers are at a greater risk when such attackers are system administrators, employees, or others somehow related to the service providers.
vii) Insufficient Due Diligence: When the providers system and mechanisms are not transparent or the consumer's adoption of such services are not based on the appropriate understanding of such mechanisms, a consumer organization may find itself in a critical situation.

viii) Shared Technology Vulnerabilities: As we have already discussed, multiple vulnerabilities come into the picture of cloud due to the sharing of underlying resources.
ix) Loss of Governance: This leads to the problems of data breaches and data loss.
x) Lock-in: Lack of standards among the service providers allow each one to store the data and apply computations on the stored data in a non-transparent manner, making portability of data and computations impossible. This leads to the unhealthy dependence on a cloud service provider due to which a consumer is unable to take advantage of better facilities offered by other providers, even when it is available.
xi) Insecure or Incomplete Data Deletion: Many a time, due to reasons varying from the migration of a consumer away from a cloud, to the consumer not needing certain data any further, data in a cloud storage need to be deleted. However, since a consumer has no way of supervising this process of data deletion and is completely dependent on a trust situation, there is no guarantee that this data is physically removed and would not be made inaccessible to other interested parties.
xii) Availability Chain: When a service provider depends on a third party for various reasons, with or even without the knowledge of the consumers, trust has to be extended to all these third parties involved. A problem to one or more such entities may cause a problem making the service of data not available to the consumers on time.

### *4.3. Cloud Security Requirements*

In this section we will look at the requirements of a secured environment and how these are related to the different service layers and deployment models in cloud. The following is the list of the requirements:

- Identification and authentication
- Authorization
- Confidentiality
- Integrity
- Nonrepudiation
- Availability

The above security requirements are to be present in a cloud environment and a possible consumer must be able to check these in different models of a cloud offering before deciding on the usage.

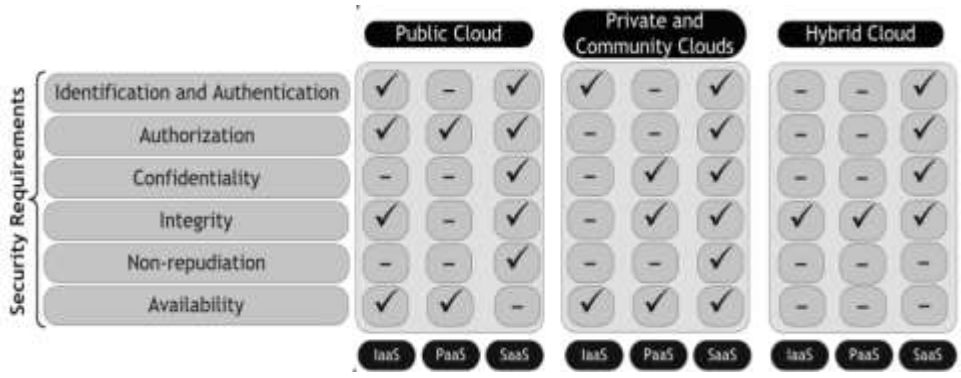



**Figure 38.1: Security Requirements in Different Models**

Figure 38.1 above shows the relationship between these requirements and the different models of cloud.

### *4.4. Cloud Security Issues*

In this section we will look at the various issues that are present at various stages of a cloud architecture. This will explain why we need to focus more on the security aspects in cloud environment.

- Data center or physical security issues
- Virtualization and hypervisor security issues
- Identity and access management issues
- Data and storage security issues
- Cryptography issues
- Governance issues
- Compliance and legal issues
- Network security issues

#### *4.4.1. Data Center Or Hysical Security Issues*

When service providers build data centers, they must have various aspects in mind; the geographical location and the environment thereof, political and governmental restrictions and rules, availability of power and other such requirements, possibilities of energy-savings and so on. All these would help the provider to keep a fault-tolerant cloud available all the time reaching a goal of 100% uptime. Data centers qualities are categorized in tiers with the lowest level being 1 and the highest level being 4. A highly reliable and available data center with special cooling techniques is assigned to tier 4 level. When so much goes in a data center, the physical security of such facilities also is a great concern and attracts attention. Data centers must have appropriate security mechanisms equipped with video cameras and other techniques to prevent any physical violations.

Further, data centers also include the networking mechanisms, which are outside the boundaries of the data center and physical security of these also must be included.

#### *4.4.2. Virtualization And Hypervisor Security Issues*

As we have discussed, virtualization is the core technique of a cloud but this is also responsible for a large number of security concerns. In this subsection, we will look at the possible issues specifically due to virtualization. Hypervisors have been identified as a source of security threats. Let us see what are the specific issues that may come from the use of hypervisors.

##### *4.4.2.1. Hypervisor vulnerabilities*

A hypervisor or Virtual Machine Monitor (VMM) is designed to run multiple VMs and applications concurrently on a single host machine. It is further a requirement that these VMs must be provided with strict isolation. However, a weakness in the design would allow attackers to exploit the presence of multiple applications on the same host simultaneously.

##### 4.4.2.2. VM escape

It is possible to design attacks that would break the isolation between a VM running on the guest OS and the host. This would make the host available to the malicious

program running in the VM. This gives the malicious program access to the underlying systems without going through the Virtual Machine Monitor (VMM) layer.

4.4.2.3. VM sprawl

If the some arbitrary numbers of VMs are created without an appropriate support or management, then these VMs will occupy large amount of resources causing other, regular applications to starve. This situation is called VM sprawl.

*4.4.2.4. Cross VM side channel attack*

When multiple VMs exist in the same host, if the VMs are not very robust, it is possible to have a malicious VM to penetrate the isolation between VMs and accessing the resources available to the other VM. Side channel attack is very dangerous since an unsuspecting application running in an attacked VM has all the confidential data exposed to the attacker.

*4.4.2.5. Single point of failure*

Hypervisors are the single points that control the whole of the host machine along with all the VMs and the guest OSs running in them. Failure of a hypervisor causes very serious problems making the VMs loose the states and hence forcing these to restart.

*4.4.2.6. Mobility*

Typically, in a cloud environment, VMs are copied and reused usually to increase elasticity. This process is called VM cloning. However, this is the cause of concern since a copy may retain an earlier data but may be used for a different purpose. Since the earlier data may be retained, this VM actually may have some confidential information without the knowledge of the original owner. While VM mobility helps, a service provider must ensure strict security procedure before reusing a VM.

*4.4.3. Identity and Access Management Issues*

A cloud can be accessed and used by anyone. Data is stored by customers and the stored data is accessed, computed upon, changed, saved back, and moved all the time by the owners or any other authorized persons. However, the cloud cannot allow a wrong unauthorized entity to access a data. Hence the ability to identify an entity, look at the permissions the accessing entity possesses and to allow only the permissible work on the data is an important part of cloud. A good identity and access management strategy must be adopted in order to ensure that data is safe in the cloud.

In this category, perhaps the most important aspect is the *identity management*. An efficient management of provisioning and deprovisioning of users to systems and applications is a major problem since the roles of users often change when there is a change in business strategy. The next process is *authenticating* the identity of a user or a system in a secure and dependable way. This requires robust credential maintenance and management, password management, use of appropriate encryption technique etc. *Authorization and access control*, on the other hand, implies establishing fine-grained for users to access the systems resources. Last but not the least is the *federation management*. Typically, different clouds must use federated identity management allowing single sign on facility to its customers.

However, since this requires exchange of identity information between the Service Provider (SP) and the Identity Provider (IdP) there is security risk.

*4.4.4. Data and Storage Security Issues*

In a cloud, users are to store their data in the cloud and allow all and any computation to happen in the cloud itself. Since the data is expected to reside in cloud, users can order a computation any time and expect to get the corresponding output soon. In case there is a problem with the data stored or with the storage where the data is stored, this is violated. Cloud providers must make a highly available data store having high integrity and confidentiality.

Particularly the fact that users may not know the current state of their data is a great concern, i.e., a service provider is able to hide if something happens to the data in order to maintain reputation. Further, with provisions like deleting files that are not accessed, a provider may, inadvertently delete an important and sensitive data.

All these bring forward great number of serious concerns about data and the corresponding storage in cloud.

*4.4.4.1. Data confidentiality*

Confidentiality refers to the fact that a piece of data must not be available to all. Only authorized user may access such data. Within an organization, such confidential data would not move out of the computers of the authorized personnel. Unfortunately, in cloud the process is left to the provider to ensure appropriate access to confidential data.

*4.4.4.2. Data integrity*

Since data is under threat in cloud due to various reasons, data integrity is of great concern. Reliability of stored data is its integrity and the apprehension is that the data may have been changed subtly or tampered with by unauthorized persons.

Typically, integrity goes closely hand in hand with authenticity. It must be provable that there has not been any change in the data while the consumer was away.

*4.4.4.3. Data availability*

Cloud outage is the problem of having the data and computes of customers not available for some time and even if a customer is in great need of data at some point, they have no way of obtaining the data till the services are restored. Machines are unreliable and every service provider would use the best possible architecture and design to ensure that outages do not affect the customers. Yet, in almost all the cases, cloud customers experience temporary or permanent partial or complete data loss.

A number of threats affect availability in an environment. Denial of service (DoS) is an example. Service outage due to failures is another concern. Also simple failures such as disk/sector failure may cause an availability problem. With the exponential growth of archival data, a small failure rate can imply significant data loss in archival storage.

*4.4.4.4. Data isolation*

In a cloud, data resides in a shared space and hence is a huge threat to the users. Due to multi-tenancy, resources (i.e., servers, clouds, storage) are shared by multiple organizations and individuals, the decision of which is beyond the control of the user. While this is good for flexibility and economies of scale, the notion of shared infrastructure is the source of concern. Since this is a very real and genuine concern, administrators of cloud must ensure that all data in cloud are completely secure and can be accessed only by authorized users. Hence, even in a shared environment, data must be kept in isolation.

*4.4.4.5. Data sharing*

While shared environment brings threats and challenges, it also brings the attractions of having a collaborative environment like online word processing, using a shared calendar, blogging, and social networking. Typically, such applications allow concurrent access of shared data to multiple users and it is easy to implement scalable and always available environment that is accessible globally. However, while allowing access to multiple users, the system must guarantee that a non-user or an unidentified entity must not gain access to such data.

*4.4.4.6. Data backup and redundancy*

Backing up data is perhaps the main mantra in any digital environment and cloud is not exempted. Just because data is stored in cloud storage does not guarantee that data is also backed up. Even in cloud, data may be accidentally lost, or even be modified by adversaries. It is also possible that the encryption key be lost. Hence as part of security, clients must learn to back up data stored in cloud thereby avoiding accidental data loss and maintaining business continuity.

*4.4.4.7. Data sanitization*

Data sanitization is the process of deleting data permanently from the storage media when user deletes some data. When a user requests to delete some data from a public cloud storage, the service provider must ensure that data is deleted completely from all the log files and backup replicas in such a way that not only no copies are left, it cannot be reconstructed by a malicious users or competitors. However, it is difficult for a user to ensure this on behalf of the client and often it is seen that due to the shared nature of the storage media, a service provider may not be able to completely delete certain data within a specified period. Hence data sanitization is a serious concern.

*4.4.4.8. Data provenance*

Data provenance means that the source of a data is traceable along with information such as who has accessed and modified the data, and what are the sequences of those actions. It is the responsibility of the service provider to ensure provenance of data before a data is stored. However, the fact that the provider is collecting such information is a source of security concern. While provenance is a necessity in auditing and forensic requirements, this data may be sensitive and confidential and hence the service provider must protect the data and must ensure that adversaries of the data provider must not get access to this data. Therefore, while collecting such information is a necessity, the integrity of sensitive provenance data must be ensured.

*4.4.4.9. Dynamic data update*

In cloud, data is stored in some remote location and it is the responsibility of the service provider that the data is secured. It is already a difficult task to ensure security of static data in remote location, although there are some methods for ensuring integrity of static data archives. This task becomes more difficult in cloud since the data is not static. Such data is often updated by clients using various methods such as insertion, updation, appending new data, deleting some part of the data etc. However, data must be made secured in spite of such frequent accesses. This poses a very serious challenge to the service provider.

*4.4.4.10. Unreliable computing*

In cloud, a client can request from anywhere and the promise of the service provider is that such a request will be executed within the time limits as specified by the SLA. This poses a great burden on the service provider. Any malicious outsider may inject an incorrect computing. This becomes more clear when we look at large framework like MapReduce. Such a framework involves thousands of servers and ensuring that all the programs running in that many servers are correct becomes a hard task. Since the attacker may pose as a valid client, each and every computation issued by any client becomes unreliable making provisioning of security from such potential unreliable computation harder.

*4.4.4.11. Cryptography*

Encryption-based security using cryptography ensures authorized consumption of data. Hence, for any security threats, typically cryptographic solutions are resorted to. However, the basic solution in all these mechanisms, be it popular and strong methods like RSA or AES, is that it is computationally infeasible to calculate values that are used as the key. Ironically, cloud provides a cheap source of computation power and it is easier for malicious users to use such computation powers to apply brute-force method to obtain passwords. Perhaps, that is the reason why researchers observe that nowadays, *brute-force* attacks represent a growing threat. Hence, while cryptographic methods may work fine in a normal circumstance, with the advent of cloud such methods may not remain that attractive ant more and caution must be used for implementation of such methods.

However, in spite of being vulnerable, cryptographic encryption is the most popular and important method of security implementation and in the next subsection, we will look at the issues of using such encryption method and see how special cryptographic methods are used in cloud.

*4.4.5. Cryptography Issue*

In cloud, stored data may be of sensitive nature e.g. personal health records, emails, customer information etc. Such data stored in cloud provides the data owners freedom from the worries of the size of data, the growth rate of this data and maintenance issues. Also, increased data availability is an added advantage. However, since there is a trust issue and cloud may, knowingly or unknowingly, allow adversaries to access a sensitive data, data owner typically apply data encryption using appropriate cryptographic method. This ensures safeguarding of the contents and guarantees data authenticity and integrity. Unfortunately, data encryption may reduce the effectiveness of data utilization. The same data is often shared with other users, some of whom may not have anticipated at the time of encryption. Unfortunately, since data stored is very large and users often need to access these data, searching is also a necessary requirement. Typically, a keyword search like Google plaintext keyword search is used to retrieve any specific file of a specific

users' requirement instead of retrieving all the encrypted files. Unfortunately, while simple encryption of data can protect privacy, it poses a serious challenge to the users since it is not possible to apply techniques such as a keyword-based search on encrypted data. Hence in cloud, encryption is a necessary yet difficult proposition. To resolve this problem, researchers use two complementary concepts in the encryption of cloud.

These are:

a. Content aware searchable encryption
b. Format Preserving Encryption

In recent years, searchable encryption techniques have been developed to allow users to securely search over encrypted data. Also format preserving encryption ensures that encryption of data such as credit card number or passport number continues to have the same format even after encryption. This means that the cypher text of a credit card number must also have the same number of digits as the corresponding plaintext. This results in easier discovery of correctness of information in a cloud data without relaxing on the security aspect.

*4.4.6. Governance*

When a user uses a cloud to store and manipulate data in cloud, it is expected that the services will be available to the user as long as he/she pays for the service and the services would continue to be there till the user does not voluntarily wants to withdraw her data from a certain cloud. However, classic case of *vendor lock-in* occurs when, even if a user doesn't wish to continue using the services of a certain provider or wants to utilize various services from different providers, she is simply not able to do that. This happens because the data, when kept in some cloud with one service provider, is converted to a certain format and stored. As long as the services of the same service provider are used, the data owner can use the data without any problem. However, later when the data is to be taken out or shared with other providers, the specific format prevents the data owner or another cloud provider to store or even use the data. This is a governance issue, which refers to losing administrative, operational and security controls.

Another governance issue is the interoperability between clouds. In the absence of any standards, if a few clouds wish to collaborate, such attempts still face security and other issues, regarding protocols, data formats and APIs. Vendor lock-in and governance issues results in customers being trapped to a certain cloud provider and make them vulnerable to data migration, price increases, reliability and security problems and many more.

*4.4.7. Compliance and Legal Issues*

The cloud business model uses SLAs to specify the agreements over a certain service, may that be in the form of IaaS, PaaS or SaaS. An SLA is always signed to formally agree on a price per service and inherent legal matters. Thus, there can be an implied subjectivity on the fulfillment of such agreements giving rise to compliance and legal issues.

*4.4.8. Network Security Issues*

Since networking and communication over the network plays a very important role in cloud, discussion on the security threats in cloud cannot be completed without seeing the threats to the networking. In this section, we will discuss a selected set of security issues associated with network communications and configurations.
The following is a list of security issues associated with network and Cloud environments:
**XML Signature or Wrapping Attack**: XML signatures are used for authentication and integrity of SOAP messages. A part of the SOAP message is signed using XML signature. It is possible for an attacker to modify the message body.
**Flooding Attack:** In this attack, the attacker starts a large number of virtual machines and fake requests thereby making the server waste time to check the validity. During this unnecessary checking, a legitimate request would also be waiting causing starvation, in an extreme situation. This is also known as distributed denial of service (DDoS) attack.
**Malware Injection Attack:** In this attack, the attacker creates a service with malicious code. When this service runs, it tries to harm legitimate requests.
**SQL injection Attack:** Malicious code is inserted into the data fields of a standard SQL query using which an attacker gains unauthorized access to databases.
**Cross Site Scripting (XSS) Attack:** In this attack, malicious scripts or codes are injected into web contents thereby forcing a website to execute the attacker's supplied codes.
**Insecure APIs Metadata:** APIs are the doors into a service. All clients use the APIs exposed by a service provider. These APIs may be used by an attacker to compromise a cloud service if these are not built in a robust fashion.
**Spoofing Attack:** It is possible for attackers to maliciously alter the content metadata files like WSDL file by running malicious codes in client machines.

### *4.5. Standards*

In a collaborative world, having standards and ensuring that all follow these standards definitely ensures better and improved collaborations and exchanges. There have been many attempts and in this section we mention the more popular ones.

**SAS 70:** This is a commonly adopted security standard among CSPs with adoption rate of almost 67%. This is developed by the American Institute of Certified Public Accountants (AICPA). The focus is on a CSP's infrastructure, policies, and procedures. It ensures that a CSP follows best practices ensuring security of clients' data.

**ISO 27001:** This standard was published in 2005 for an Information Security Management System (ISMS). Adoption of this standard is about 33%. It is generally accepted to be a holistic information security mechanism since it examines in great detail how an organization manages information security management.

**SAFE HARBOR:** Typically this standard is used by organizations in the U.S. and European Union.
**NIST:** National Institute of Standards and Technology (NIST) is a body that regulates many things in cloud and they have proposed a set of standards known as the NIST standards.
**HIPAA:** The U.S. Health Insurance Portability and Accountability Act (HIPAA) is a popular standard among CSPs. It is followed by roughly 16 percent of cloud service providers.

**FISMA:** This is the Federal Information Security Management Act used by U.S. federal government processes or programs to measure security for federal IT systems.

**5. Security as a Service**

Since providing security is a problem, large business houses try to outsource their security concerns and let others find a solution. Taking this as a business opportunity, service providers provide integrated security service in corporate infrastructure and allow others to rent this environment. This is useful for the business houses and even smaller ones as well since the subscription would be a much cheaper option in comparison to obtaining a custom-made solution for them. Such a service is called Security as a service (SECaaS) since security is delivered as a service from the cloud. This method of deploying security solution is easy for the clients since they are not required to acquire on-premise hardware avoiding substantial capital outlays. Typical for such security services often are services such as authentication, anti-virus, anti-malware/spyware, security event management etc.

**6. Summary**

In this module we looked at security aspect in cloud. We learnt what is cloud security and why security is a problem in cloud? Critical aspects of cloud security are investigated to provide better perspectives. We then looked at the various threats and requirements of cloud security. Issues pertaining to security provisioning in cloud is looked at next followed by a discussion on the standards in cloud for security. The module is completed by a short discussion of security as a service.

In the next module, we will look at case studies in cloud.

## References


1. Oyegoke, Folusho Abayomi. "Security Challenges of Cloud Computing For Enterprise Usage and Adoption", IOSR Journal of Computer Engineering, 1.16: 57-61, 2014.
2. Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., & Inácio, P. R., "Security Issues In Cloud Environments: A Survey", International Journal of Information Security, *13*(2), 113-170, 2014.
3. Modi, C., Patel, D., Borisaniya, B., Patel, A. & Rajarajan, M., "A Survey on Security Issues and Solutions at Different Layers of Cloud Computing", The Journal of Supercomputing, 63(2), pp. 561-592, 2013.
4. Bond, James, "The enterprise cloud: Best practices for transforming legacy IT", O'Reilly Media, Inc., 2015.
5. W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing" NIST Special Publication. http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf.
6. Reed, Archie, Chris Rezek, and Paul Simmonds, "Security Guidance For Critical Areas Of Focus In Cloud Computing", Cloud Security Alliance, 14-44, 2011.
7. https://en.wikipedia.org/wiki/Information_security

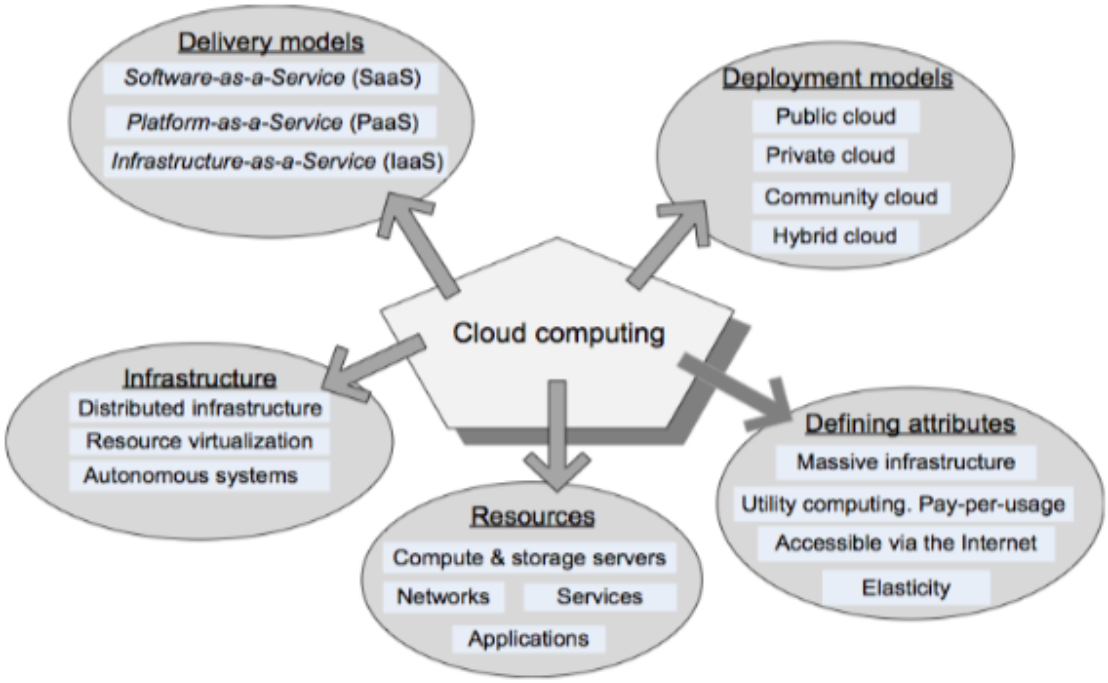
- Oyegoke, Folusho Abayomi. "Security Challenges of Cloud Computing For Enterprise Usage and Adoption", IOSR Journal of Computer Engineering, 1.16: 57-61, 2014.
- Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., & Inácio, P. R., "Security Issues In Cloud Environments: A Survey", International Journal of Information Security, *13*(2), 113-170, 2014.
- Modi, C., Patel, D., Borisaniya, B., Patel, A. & Rajarajan, M., "A Survey on Security Issues and Solutions at Different Layers of Cloud Computing", The Journal of Supercomputing, 63(2), pp. 561-592, 2013.
- Bond, James, "The enterprise cloud: Best practices for transforming legacy IT", O'Reilly Media, Inc., 2015.

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 4: Delivery Models
# Module No: CS/CC/4
# Quadrant 1 —e-text

## 1. Introduction

Cloud computing is a service oriented architecture. The term service in the world of software indicates endpoint of a connection. And, the connection should be offered in an environment supported by the underlying computing system models. In our case, the underlying system is our 'pool of resources', which is an essential characteristic of cloud computing, as we learnt in the last module. Also, as part of the characteristics, we learnt that Internet is an essential component of cloud. Now, the 'connection' here is to be offered through the Internet and the endpoint of this connection is the service. Hence, in cloud, all and any resource that is provided for the customer is a service. Thusstorage, processing, bandwidth, and active user accounts etc. are all offered as a service.Cloud is an environment through which these services (i.e., the pooled resources)are offered in a pay-as-you-go model accessible over the Internet. It is important to understand how these services are delivered to the consumers. In cloud, the mechanism of delivery of a serviceis dependent on the type of the service being delivered. Thus, there can be as many delivery models as there are different service offerings in the cloud. However, NIST (National Institute of Standards and Technology) originally had definedthree delivery models and we will study these here.

Figure 1 below shows the overview of cloud. In earlier modules we learnt the types of resources, Infrastructure, defining attributes, and deployment models. In this module, we'll discuss about the delivery models in cloud. This will complete the discussion on the various components or parts of the overview of cloud computing components.




Courtesy: Dan C. Marinescue [1]

**Figure 1: Overview of Cloud Computing**

## 2. Learning Outcome

A service in general is a transaction between a provider, who will offer the service and a consumer, who will use the service. Since cloud is about software service, the delivery models are typically described from the perspectives of the service providers and the service consumers. In case of public cloud the service providers are the cloud owners, who understandably have built the infrastructures in the cloud stack and are offering the facilities of these through one of the many delivery models possible. In case of private, hybrid or even community cloud, these are the people who are helping the organization(s) to create and offer the services. In either case, there are service providers and service consumers. In this module, we will discuss the benefits of the delivery models from the perspective of both the providers as well as the consumers.

At the end of this module, students will be able to:

1. Understand and appreciate what is Software-as-a-Service.
2. Understand and appreciate what is Platform-as-a-Service.
3. Understand and appreciate what is Infrastructure-as-a-Service.

## 3. Cloud Stack

Figure 2 below shows the cloud stack, which contains all the layers available in cloud. Delivery models allow one or many of these layers to be controlled by a consumer.

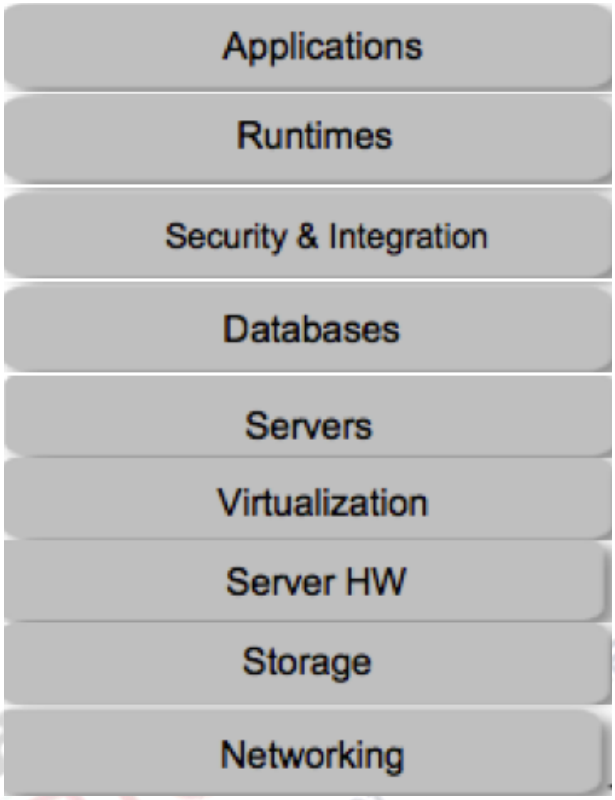



**Figure 2: Cloud Stack**

As is expected, we find that the application is the uppermost layer and, in many cases, this is the primary concern of the users. Runtime is the environment where the application is deployed. Next comes the security and integration. Databases and the servers are next where the data will be stored and the necessary processes will run, while the hardware components of these two software layers, viz., server hardware and storage, are under the virtualization layer that keeps the hardware hidden away from the users.

Details of all the layers of the stack need not be understood at this point in time, which we'll learn in later modules. it is enough to understand only a few concepts at this point of the course.The application is always run at the top and the whole stack is dependent on the underlying server and storage hardware and these are connected using the networking. These are the major assets of a cloud owner. Delivery models will be explained using the stack.

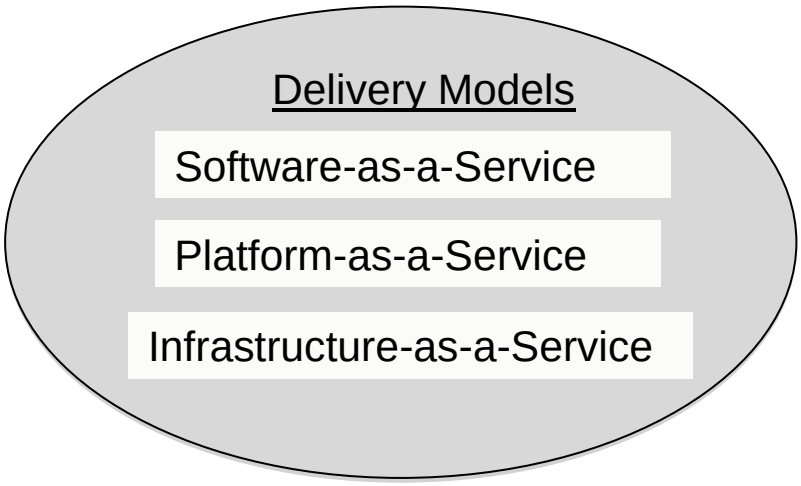
### 4. Delivery Models

As per NIST, there are three possible delivery models. These are:

1. Software-as-a-Service or SaaS
2. Platform-as-a-Service or PaaS

3. Infrastructure-as-a-Serviceor IaaS

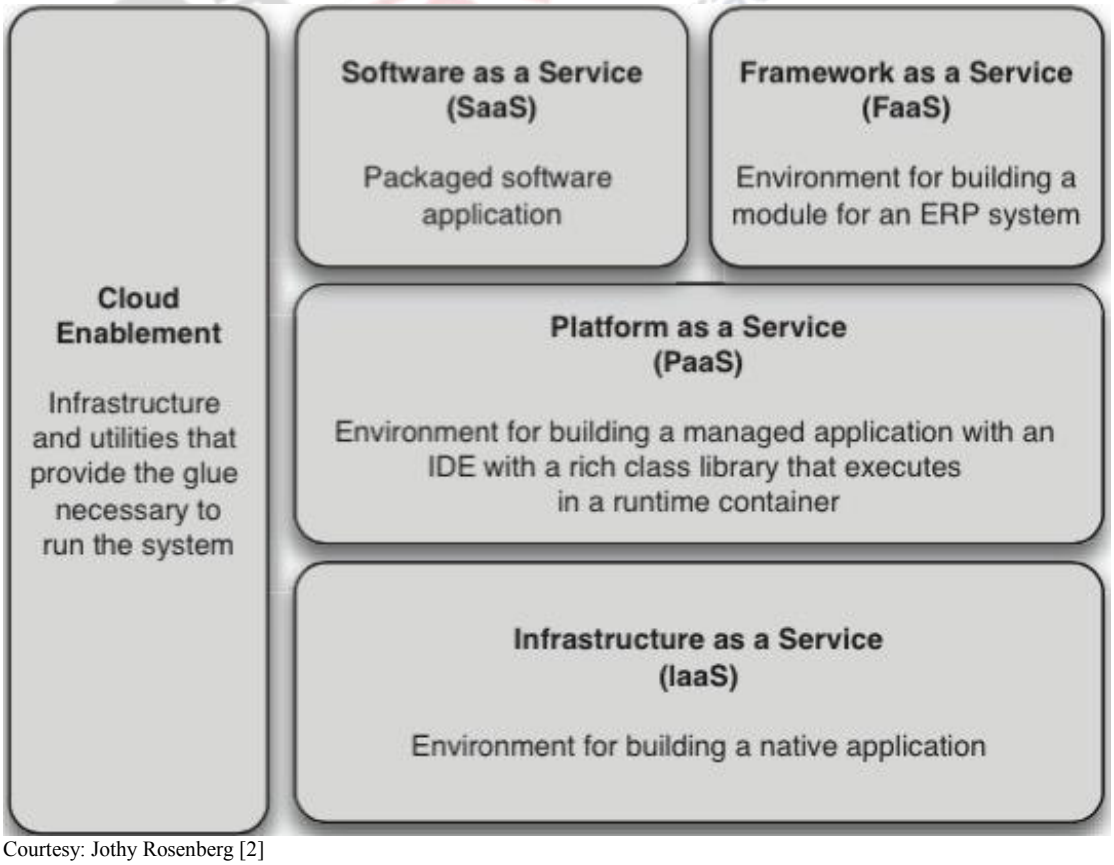
This is shown in Figure 3 below.




Courtesy: Dan C. Marinescue [1]

**Figure 3: Delivery Models**

Figure 4 below shows the layers of the delivery models. This explains the abstraction of the different delivery models from the perspective of what is available in which model.




Courtesy: Jothy Rosenberg [2]

**Figure 4: Models in Layered Architecture**

Software-as-a-Service is at the top along with Framework-as-a-Service, while Infrastructure-as-a-Service is at the bottom and Platform-as-a-Service is sandwiched in the middle. This explains the pre-packaging that is offered at each delivery model. The flexibility of the users using the models increases as we go down the layers, with maximum flexibility provided to the Infrastructure-as-a-Service users and minimum flexibility offered to the Software-as-a-Service and the Framework-as-a-Service users. Now, let us take a look at the details of these three models.

### 4.1. Software-as-a-Service or SaaS

According to NIST, Software-as-a-Service (SaaS) is defined as follows:

> "The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings."

Software-as-a-Service (SaaS) refers to services and applications that are available on an on-demand basis. Itenables the consumer with the capability to use the provider's applications running on a cloud infrastructure. Consumers can easily access the applications running in the cloud using any device at their ends. Providers may provide a web browser based API (e.g., web-based email). Alternatively, a program interface is provided by the provider using which the consumer can access the offered applications. SaaS is a rather old concept, it being in existence long before the concept of cloud came. However, SaaS gained momentum when it was offered through cloud.

SaaS offers software as it is and does not have an opportunity of customization based on the users requirement. In this perspective, Framework-as-a-Service (FaaS) is an environment that is very closely related to SaaS bringing in the required flexibility in SaaS offering. Users using FaaSare allowed to extend existing functionalities of the SaaS applications. For example, the services provided by the Salesforce.com are SaaS and those provided by Force.com areFaaS that extends the Salesforce.com SaaS offerings. It is, sometimes, not sufficient to use specific application in the given form, at the same time it may be too time-consuming to write the whole application. It is in such a scenario that FaaS works. It is possible to enhance the capabilities of the base SaaS system using FaaS offering. Customized and specialized applications may be created out of the general-purpose SaaS applications suitable specifically for a specific organizationthat can be made available to any SaaS customer. However, the restrictions on specific languages and APIs as provided by the FaaS environment must be followed.

SaaSmay be considered to be the most advanced version of all the services provided by cloud service providers in the sense that the specific operating system, the software required and of course the needed applications, all have to be supplied and maintained by the service provider, along with the versions and editions of the applications etc.Applications are hosted in the servers of the service provider. On the other hand, in SaaS, the responsibility of the consumer is the least.

Let us consider the example of an organization ABC that uses Microsoft Excel on a regular basis. ABC has offices in various cities in India and the employees of the organization often need to exchange information to collaborate on projects that are handled by various employees located in different cities. When the office located in Kolkata decided to upgrade the version of MS Excel, if all the other offices also do not do the same, there will be a versioning problem when next time the employees in Mumbai office tries to merge an Excel file with another sent by a colleague in Kolkata office. However, if all the employees in all the offices would use a cloud service offering the service over the Internet, all the employees would use the same version and hence no problem would be there.

The above example shows that the traditional desktop applications such as word processing or spreadsheet, when delivered to customers over the Internet by a third party, reduces the burden of maintaining and upgrading the software, freeing the consumer to spend the time on more important aspect of the business.

From the perspective of the consumer, the SaaS model enjoys the following benefits:

- No infrastructure cost
- No upfront cost of buying any software
- Administration is simple and easy
- All the updates are taken care of by the service provider
- Performance guarantee by the provider
- In case multiple users are collaborating, versions would not be a problem hence, collaboration is easy
- Since it is hosted in cloud, it is accessible from anywhere through the Internet.
- Rapid implementation
- Standalone and configurable applications
- Subscription and pay-as-you-go (PAYG) pricing

However, the service providers are burdened with responsibilities. The providers' concerns are the following:

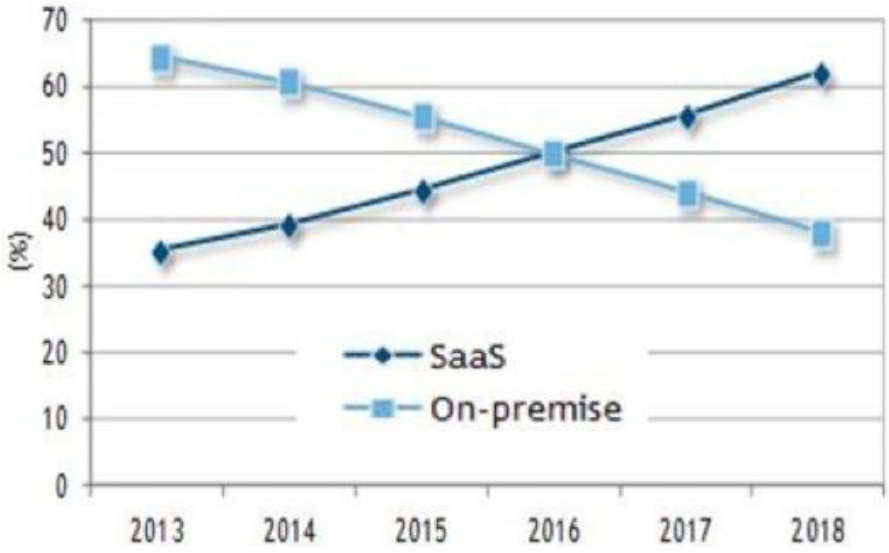
- Installation, configuration and maintenance of all servers (web server, mail server, database server etc.),
- Purchase and maintenance of all the required software in the correct version.
- Integration of the software (if required),
- Patch management, since often software need to be patched due to bugs or problems found in them,
- Monitoring usage and billing,
- Ensuring security,
- And many more.

However, in spite of the ease of use from the point of view of users, all situations are not suitable for SaaS. Typically, there are a few types of applications that are suitable candidate to be used as SaaS model. Suitability of SaaS is limited to the followings:

- When many users use the same product, such as email.
- For applications where significant peak in demand comes only periodically, such as billing and payroll.
- If there is a need for Web or mobile access, such as mobile sales management software.

- The overall need for the application is only short-term, such as collaborative software for a project.

The most important contribution of SaaS is that it allows users to be free of the worries of buying, installing and maintaining the required software, along with the need to keep track of any new version or a patch that may be I the market.




Courtesy: IDC [3]

**Figure 5: SaaS versus On-premise Software**

Thus, having on-premise software can be eliminated or reduced drastically by SaaS. More and more customers are understanding this advantage and using cloud based SaaS for their requirements. Forbes reports that IDC has predicted that the growth of SaaS would overtake the on-premise software acquisition, as shown in Figure 5 above.

Now, we turn our attention to the next delivery model, Platform-as-a-Service or PaaS.

### 4.2. Platform-as-a-Service or PaaS

According to NIST, Platform-as-a-Service (PaaS) is as follows:

> "The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.3 The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment."

PaaS provides the users with an environment where an application can be developed and deployed, saving developers from the complexities of the infrastructure side. This allows the developers concentrate only on the development of the application thereby improving the speed of development, and allowing the consumer to focus on the application itself. In a PaaS model, the infrastructure such

as the servers, databases, networking, operating system etc. are taken care of by the PaaS provider, freeing the application developers from many worries. However, the language or the platform for the application development must be the ones supported by the service provider and offered through predefined interfaces. Therefore, if the environment required is different from the ones being offered by a certain vendor, the developer is forced to move out and look for other vendors who can provide the required environment. The consumer would consume and pay for the CPU, bandwidth, storage and any such items that may be used during the development and deployment of the application. However, in this model there is no scope for any direct interaction or managing of the underlying infrastructure, which is still managed by the service provider. Issues such as how and where to deploy the applications in terms of the underlying infrastructure, issues related to the operating system, how to provision the underlying resources (both hardware as well as software) and how or when to configure the supporting technologies such as load balancers and databases are all under the strict control of the service providers. Developer's job is to design, implement and deploy the application through the use of the interfaces offered as part of PaaS model in the form of a Web-based interface or in the form of programming APIs and libraries.

PaaS consumers are not to directly interact with or administer the virtual environment deployed in the cloud. Instead, they are to concentrate specifically on writing the application. While this brings the benefit of simplification of management and handling of various resources, it also comes at the cost of less flexibility. Further cause of concern maybe from the fact that it is required for a PaaS consumer to necessarily use the languages/platforms supported by the service provider.

Having said that, in a PaaS model, installation and configuring of the servers, operating systems, and databases are all done by the service providers. They are also responsible for obtaining the required licenses for any software being used for the development of the application, althoughunlike in a SaaS model, the consumer here bears some burden. These include the setting up of the applications and operations needed for the development and deployment.

Using PaaS is highly beneficial for the consumer since they have a reasonably flexible and scalable service while being free from the trouble of acquiring all the necessary resources. Flexibility also comes from the fact that, while being under the strict supervision of the service provider, the platform can be adjusted according to the needs of the consumer. For example, if a company using the PaaS model of a service provider suddenly observessomething that could cause a peak load on its client website, it can easily and quickly increase the technological resources (since cloud is elastic, there is no problem to acquire more resources). This can be done seamlessly, without any interruption or impact to its customers.

According to authors of the book "Cloud Computing, A Practical Approach" [5], the following are the characteristics of PaaS:

- *Runtime framework.* This is environment for coding and is the most fundamental offering of PaaS model. Since consumers' requirement is to develop and deploy, the development environment is of utmost importance and this is provided through runtime framework. The runtime framework executes end-user code followingto the policies set by the user and the provider.
- *Abstraction*. The level of abstraction is a way to recognize the involvement of the consumer in the cloud. While PaaS models have lesser levels of

abstraction than its counterpart in SaaS, it enjoys more abstraction when compared to the other model, IaaS since it has more interactions and involvement in the system than SaaS and less in comparison to IaaS.

- Automation. Calculating the percentage of the infrastructural resources needed by the developed application to be appropriately deployed and running in cloud is an automated process in PaaS model. This means that the developer need not worry about the requirement of the infrastructure for the deployment of the software being developed using PaaS model. The amount of requirement is judged automatically and the provisioning for resources are done. This, of course, is at a cost to the consumer and are all part of the service level agreement between the provider and the consumer before the process of consuming starts.
- Cloud services. This again is an essential part of the whole mechanism where everything is being offered as a service. These services generally include specific components for developing applications, advanced services for application monitoring, management, and reporting etc.
-

Perhaps Google AppEngine is the most popular product in PaaS category. It provides a scalable runtime based on different programming languages supported by it. It also offers additional APIs and components provided to the developer that offer further scalability.

The following are the benefits that a PaaS consumer enjoys:

- Lower upfront and operational costs
- No investment in terms of hardware and software maintenance
- Greater and faster scalability
- Improved performance due to access to better systems in cloud
- Integration is seamless
- Development is faster due to easy access to improved environment
- Server-side scripting environment
- Database management system
- Access to any amount of storage

### 4.3.Infrastructure-as-a-Service or IaaS

According to NIST, Infrastructure-as-a-Service (IaaS) is defined as follows:

> "The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls)."

IaaS provides great flexibility and control over the cloud resources being consumed,

but typically more work is required of the developer to operate effectively in the environment.

The 'infrastructure' in the term infrastructure-as-a-Service (IaaS) is the use of infrastructure or virtual machines on demand. An IaaS provider supplies virtual machine images of different operating system flavors. The developers can run any custom or packaged applications by tailoring these images. The applications can run natively on the chosen OS. The service provider is responsible for the network elements, servers and other hardware required. Here the level of abstraction is far less when compared with the earlier two models, viz., SaaS and PaaS, since the consumer is responsible for the installation and operation of its operating system and all the required applications. Thus, IaaS delivers customizable infrastructure on demand. As part of IaaS offering, it is possible to obtain servers as well as network devices, load balancers, and database and Web servers.Storage and bandwidth are also consumable commodities in an IaaS environment, with storage typically charged per gigabyte per month and bandwidth charged for transit into and out of the system. Use of these is typically metered and charged in hour-long increments.

IaaS is good for workload partitioning, application isolation, sandboxing, and hardware tuning. The providers of IaaS model allow consumers to deploy and access virtual machines (VM) with requisite capabilities and use these VMs as required. Therefore, it is an improved utilization of the IT infrastructure.IaaS provides a secured environment where third party applications can be safely executed and cannot cause harm to the rest of the environment. From the perspective of the customer using IaaS allows them to take advantage of the full customization offered by the cloud to deploy their infrastructure without incurring the cost of installation and maintenance of all the underlying hardware.

While PaaS models offer a way to deploy and manage applications on the cloud, IaaS offers a similar scenario in virtual machines on top of which the IT infrastructure is built and configured.Also, in PaaS, the service provider allocates more resources in case the deployed application needs it and the agreement between the provider and consumer supports such an action. However, the same is not true for a IaaS solution. Here, the consumer has to make an estimate of the CPU, memory etc. for the VMs and these only will be accessible. IaaS solutions, only provide ways to provision more resources.

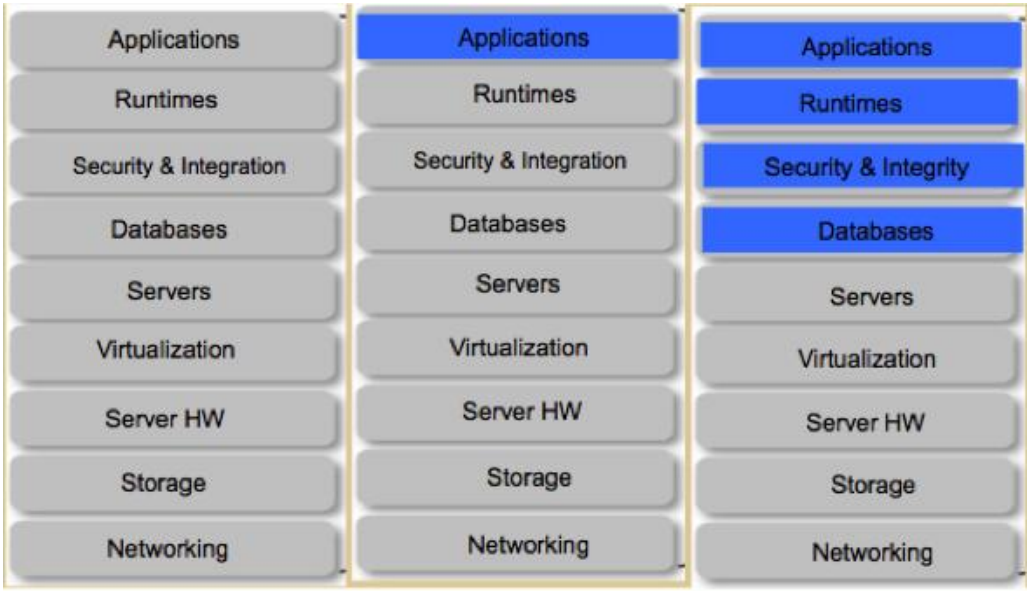
Benefits of the IaaS model include:

- No initial hardware investment
- Highly Scalability
- Pay per Use
- Location independence
- Physical security of data center locations for servers

**5. Comparison of the Delivery Models**

In SaaS users possess the capability using applications provided by the cloud. However, SaaS does not allowthe consumer any level of control over the deployed applications. Nor is there any control of the user over other resources being used such as the platform or the infrastructure. *PaaS*, on the other hand,*enables users* to develop their own applications. PaaS also provides any necessary tools or language

support enabling users to deploy the applications in the cloud infrastructure. The most flexibility is however provided by *IaaS.*It not only allows the user use the virtualized resources, users are able to run new operating system or any arbitrary software. Figure 6 below shows the ownership of the three delivery models. 6(a) represents the SaaS solution, (b) represents the PaaS solution and (c) represents the IaaS solution.




**(a)** **(b)** **(c)**

**Figure 6: Relative ownership of the cloud stack**

The blue blocks in the figures 6(a), 6(b) and 6(c) indicate the control of the consumer while the gray ones indicate the control of the providers. As can be seen from the figure, In SaaS, all the control lie with the service provider and nothing of the stack is controlled by the consumer. As shown in (b) above, in PaaS, while provider controls all the components right upto and including the Runtime, the consumer owns the applications. This scenario changes quite a bit in the IaaS model as shown in the (c) part of the figure above. Here, the consumer controls upto Databases, while the provider controls the rest of the stack.

A user of IaaS is operating at the lowest level of granularity available and with the least amount of prepackaged functionality.A PaaS user operates at a higher level compared to the IaaS user and SaaS user operates at the top-most level where everything comes prepackaged.

Table 1 below summarizes the various aspects of the three models:

| Category | Characteristics | Product Type | Vendors/Products |
|---|---|---|---|
| SaaS | Customers are provided with applications that are accessible anytime and from anywhere. | Web applications and services (Web 2.0) | SalesForce.com (CRM) Clarizon.com (project management) |

| | | | |
|---|---|---|---|
| | | | Google Apps |
| PaaS | Customers are provided with a platform for developing applications hosted in the cloud. | Programming APIs and frameworks Deployment systems | Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse |
| IaaS/ HaaS | Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure. | Virtual machine management infrastructure Storage management Network management | Amazon EC2 and S3 GoGrid Nirvanix |

Now, let us now take a look at the growth rate of these three delivery models. Figure 7below shows how the user ultimately uses the various services. They have used the measure CAGR or the **compound annual growth rate.** It is a useful measure of growth over multiple time periods. It is a way of measuring the growth rate using compounding mechanism over the time period.From the figure it is clear that initially in 2013, IaaS has the majority workload share, but by 2015 SaaS workloads take the majority share, and by 2018 will have 59 percent share of all cloud workloads. PaaS will have the second-fastest growth, although it will lose the share of total cloud workloads from 15 percent in 2013 to 13 percent by 2018.


Courtesey: Cisco Global Cloud Index 2013-2018[4]

**Figure 7: Deployment percentage of the three models**

## 6. Summary

In this module, we have looked at the three major service delivery models used to deploy services in cloud. These are SaaS or Software as a Service, PaaS or Platform as a Service and IaaS or Infrastructure as a Service. While SaaS is used for users that wish to use an application in a provided version, PaaS is used for the development of an application in a customized manner and deploy. IaaS, on the other hand allows the use of the infrastructure of the service provider and create platforms as well as applications in a customized environment. IaaS users enjoy the most from the perspective of flexibility and control, with PaaS users coming as the second while the SaaS users do not enjoy any. However, SaaS users need to be responsible only for their data and need to pay for the rest, while PaaS and definitely the IaaS users shoulder greater responsibility in the cloud environment.

## References


1. Dan C. Marinescu, “Cloud Computing, Theory and Practice”, 1st Edition, Morgan Kaufmann, 2013.
2. Jothy Rosenberg and Arthur Mateos, “The Cloud at Your Service*The When, How, and Why of Enterprise Cloud Computing*”, Manning Publication, 2010.
3. http://www.forbes.com/sites/louiscolumbus/2014/12/20/idc-predicts-saas-enterprise-applications-will-be-a-50-8b-market-by-2018/
4. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf
5. Toby Velte, AnthoniVelte and R. C. Elsenpeter, “Cloud Computing, A Practical Approach”, McGraw Hill Publication, 2009.
6. RajkumarBuyya, James Broberg, A. M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley Publications, 2011.
7. RajkumarBuyya, Christian Vecchiola, S. ThamaraiSelvi, “Mastering Cloud Computing Foundations and Applications Programming”, Elsevier Morgan Kaufmann, 2013.

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 1: Introduction
# Module No: CS/CC/1
# Quadrant 1—e-text

**1. Introduction**

Welcome to the new and exciting world of cloud computing. In this course we will learn what is cloud computing, how to make the most of this great new technology and also how to be a part of the innovative side of this technology. There have been many definitions of cloud computing each describing the phenomena from a certain perspective. Since cloud computing is an amalgamation of many technologies, some old and some new, it is difficult to provide a concise definition that can describe this phenomenon. National Institute of Standards and Technology (NIST) [1] defines cloud as follows:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a sharedpool of configurable computing resources (e.g., networks, servers, storage, applications, and services) thatcan be rapidly provisioned and released with minimal management effort or service provider interaction."

Another, most often used definition of cloud computing is the following:

"Cloud computing is the delivery of computing services over the Internet."

From the simple task of defining the phenomena, one can understand that cloud consists of various characteristics. In this course, we will try and understand how and why these characteristics give a different dimension to the concept of cloud. However, among so many characteristics, perhaps the most important and mentionable is the rate of growth, due to which cloud computing is being treated as a 'phenomenon' by the industry as well as the academia. To understand why the growth rate of cloud computing is called a phenomenon, let us review what Gartner[2] ,an American information technology research and advisory firmproviding technology related insights, observes:

"The use of cloud computing is growing, and by 2016 this growth will increase to become the bulk of new IT spend".

It should be noted that the IT industry is considered to be a major money-churning sector and if this sector is predicted to be spending the bulk of its expenditure on cloud, it is easy to understand the importance of this cloud as a newtechnology. Another pointer towards the growth of the cloud-related exposure is the graph shown in Figure 1.1 below, which represents prediction on the growth of the numbers of subscribers who would use cloud storage.While the number of cloud storage users was 500 million in 2012, it has grown to 700 in 2014. However, the most important point in the graph is the fact that this number would become 1300 million in 2017!

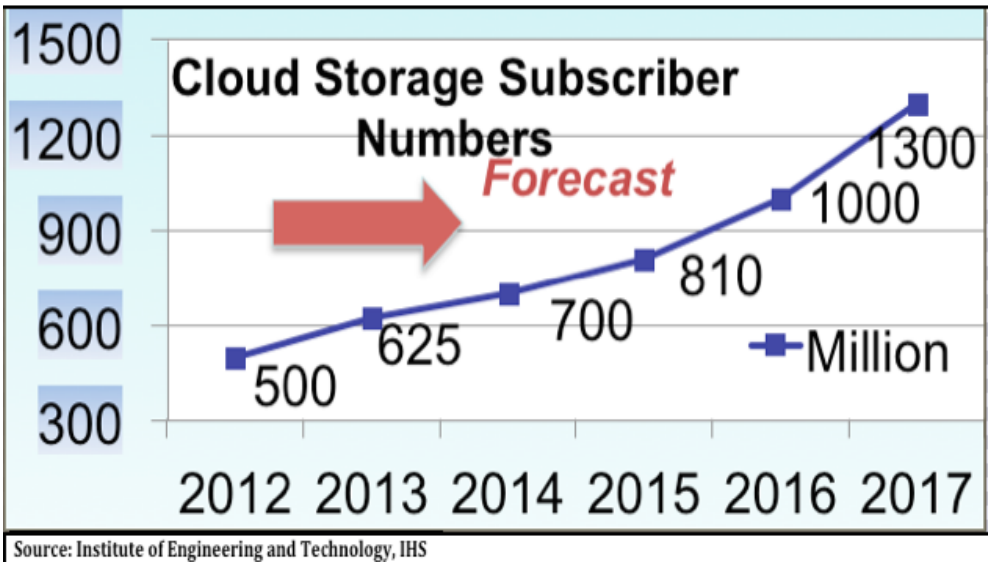



**Figure 1.1: Cloud Storage Subscriber**

So from the above discussion one can conclude that cloud computing is fast becoming an accepted and adopted technology used by many. This in turn serves as a motivation to study and understand cloud computing and its related technologies. Before we start with the concepts of cloud, let us take a look at how cloud computing came about.

Over the past 60 years, there have been many changes with respect to platform as well as the environment. We have seen changes in the architecture of the machine itself. Changes became the norm everywhere in computing. Some examples areOperating Systems, the way we interconnect our systems through networking and of course, the way we write and use our applications. In fact every aspect of the computer has gone through a sea of change. However, with these changes came the problem of scalability. As we grow or increase in one area, the other related areas must also be able to grow accordingly to accommodate the changes. For example, as we increase the size of the registers used in a computer, the memory size also needs to grow and the software should be able to exploit this power effectively. Therefore, changes in hardware, networking and operating systems caused paradigms of computing to evolve reflecting the need to grow. From one computer, we have learnt how to use multiple computers to solve large-scale problems using the Internet as our communicating medium. Our applicationshave become more data-intensive and network-centric. All these have helped to solve larger computational problems that have become the need of the day.The ability to grow, and shrink if required, is known as scalability. With the advent of computing technology, we need to have scalability that will accommodate the changes. The computing that can accommodate by scaling appropriately is known as scalable computing.

This module is devoted to explore the development and changes that computing in general has gone through paving the way to cloud computing.

## 2. Learning Outcome

In this introductory chapter of the course Cloud Computing, basic understanding of the concepts and how the evolution in computing has taken place are to be discussed. At the end of this module, students will be able to:

1. Understand and appreciate what is meant by the term Cloud Computing.
2. Understand the process of evolution in computing.
3. Understand and appreciate the different types of computing paradigms.
4. Learn to use and read the Hype Cycle.
5. Gather knowledge about various companies that are offering services in cloud.

### 3. Evolution of Computing

First let us look at how computing has evolved to the modern incarnation. In early days pebbles were set on clay board to calculate additions and subtractions. In 3600 BC, Abacus was invented in China. Although various other devices were invented since then that aided calculations, the computer era is said to have started around 1940, which marks the beginning of the first generation of computers. Since then various technological developments have been marked in different generations.The whole era of computing till date has been divided into five generations. During each generation there had been some fundamental technological development essentially changed some basic operations of the computers. Almost always the result of such resulting in increasingly smaller, cheaper, more powerful, efficient and more reliable computing devices.

The beginning of this journey started in 1940 and this phase went on for about 15 years till the first change came in 1956. This phaseof 15 years is termed as the First Generation of computing devices. The computers in this era used vacuum tubes for circuitry and magnetic drums for memory. They were very large in size, often taking up entire rooms. They were also very expensive to operate needing a great deal of electricity and generating a lot of heat often causing malfunctions.First generation computers relied on machine language. This is a binary language that is understood by computers. At a point in time only one problem could be solved by these computers. Punched cards and paper tapes were used for inputs while outputswould be printed on papers. Not only were these hard to operate, they were also not very reliable. Two examples of computers of the First Generation are UNIVAC and ENIAC.

The Second Generation of computing devices started in around 1956when transistors replaced the vacuum tubes. While the transistors were invented in 1947, its use in computers came in around 10 years later and started the next generation of computing devices. With transistors, the size of the computers became smaller and the computers became cheaper too. Not only binary or machine language was replaced by symbolic language and the operating system was improved making the machines faster during this generation, high-level language appeared with the early versions of FORTRAN and COBOL. This generation also saw the first computers that stored instructions in their memory.

Around 1964, when transistors became smaller and could be placed on silicon chips in the form ofintegrated circuits, the Second Generation gave way to the Third Generation of computing devices. In this generation, computers became more efficient, input and output devices like the keyboard and monitor appeared and communicationwith the operating system was made possible. The ability of multi programming, *i.e.*, running different applications at the same time, marks an important achievement of this era. This era also made computers available for the mass audience due to two factors: size and price, both of which became lesser.

Around 1971 came the age of microprocessor, with the ability to put a large number of integrated circuits on one chip. Intel 4004 chip marked the beginning of this era that could fit in all the components of a computer in a chip. With this, the concept of home computers appeared. Graphical User Interface (GUI) along with mouse and other handheld devices became part of the computer in this era termed as the Fourth Generation. Many new features were added in this generation and modern computers of today continue to have all these features developed in the Fourth Generation.

Around 1990, development of computing devices was augmented with the appearance of Artificial Intelligence and parallel processing. This age had the aim of using Natural Language Processing and providing the comfort of interacting with the computers using natural languages. This is the current era, which is still developing and is termed as the Fifth Generation.

### 3.1. Evolution of Internet and Internet-Connected Devices

With the advent of computing devices, other areas such as networking, the Internet and computing paradigms also have seen major evolutions. However, these cannot be demarked as clearly as the computing devices. Advances in the concepts of networking made way for the unprecedented growth of the Internet and internet-connected devices. With so much of advancement all around, the requirement today is to provide concurrent computing services to billions of the Internet users. This can be made possible in two ways. The first is that we use supercomputersto provide such services, which is expensive and hard to implement at the required level since this calls for a large number of supercomputers. The second option is to use large datacenters with clusters of commodity computers that can replicate the power of supercomputers at a feasible level. The computing world,by and large, used the second option and since it is financially not so challenging, we saw an unprecedented growth in the internet-connected devices. It is predicted that this number is going to grow more in future. As published by Cisco (Figure 1.2 below), the world is witnessing an unprecedented growth in the number of internet-connected devices in the last two decades. We also observe in this figure that while in the five years between 2015 and 2020the world population is to going to grow from 7.2 to 7.6 billions, both the internet-connected devices and the internet-connected devices per person are going to become almost double! This comparison perhaps provides a perspective to the rate of growth of the Internet and the devices connected to it.

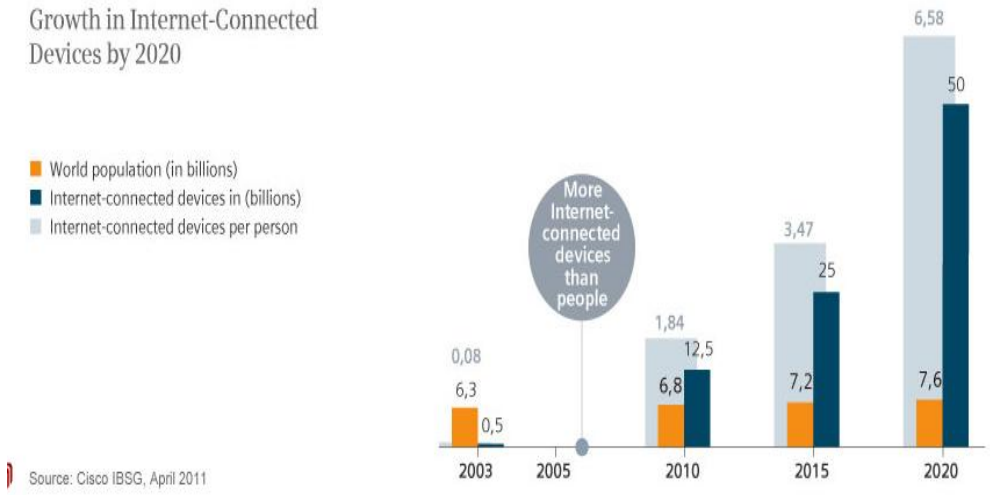



**Figure 1.2: Internet-Connected Devices**

Along with the increase in the number of Internet users and devices, the mechanism of Internet usages also has gone through a major evolution. As shown in Figure 1.3 below, while initially users were happy by merely publishing and obtaining available information, the phenomenal growth in the Internet usage gave rise to greater interaction leading to the need of intelligent and machine-based automatic discovery mechanisms. Large number users in the Internet also meant the use of different terms to indicate same concept and hence the advent of semantic computing came forth. Currently, with the popularity of social networking and data analytics,new challenges have appeared in the horizon and the corresponding development has to match these growth as well in the form of deep web.

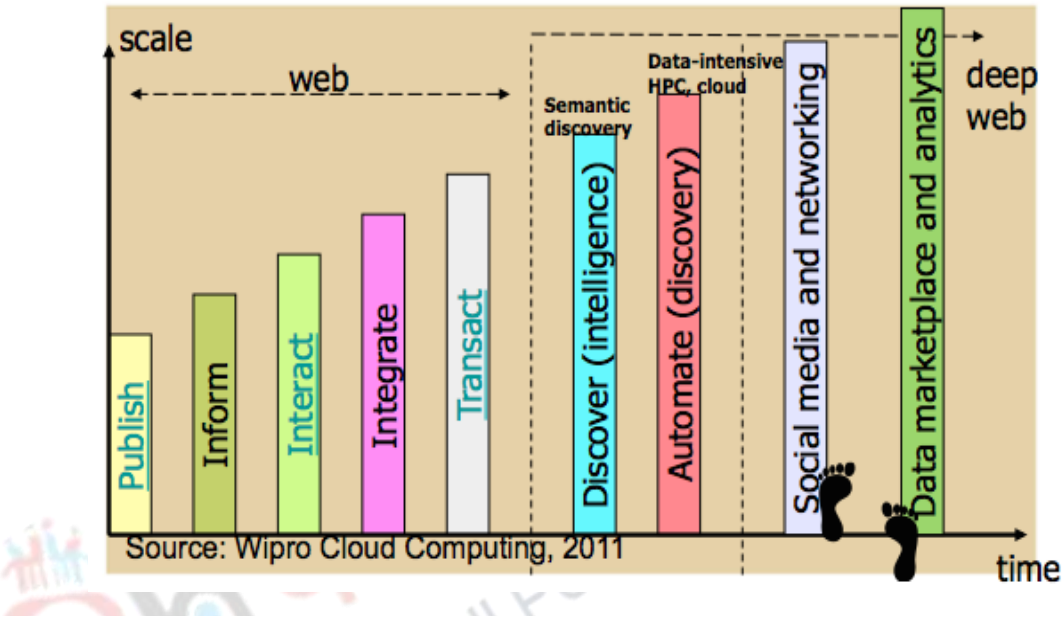



**Figure 1.3: Evolution of Internet Computing**

### 3.2. Evolution of Computing Paradigms

Along with all computing devices and the Internet usage, computing paradigms also developed keeping pace with these advancements. While initially computing was perceived to be centralized, other advanced mechanisms came about. Overall, the computing paradigms can be categorized as follows:

- Centralized Computing
- Parallel Computing
- Cluster Computing
- Distributed Computing
- Cloud Computing

In **centralized computing paradigm**, all computer resources such as processors, memory, storage etc. are typically located in one physical system and controlled centrally. Resources are shared and used by various processes under one operating system. While such systems can offer high security and integrity and low maintenance with high reliability, they suffer from scalability, inflexibility and lack of usability problems.

**Parallel computing paradigm** refers to the mechanism of a computer system,which makes it capable of running computations parallelly. Typically processors share a central memory and communicate through the shared memory in a tightly coupled manner. Alternatively, they can be loosely coupled having distributed memory. Communication in this case,can be via message passing.

**Cluster computing paradigm** refers to a cluster of connectedcomputers. It is a type of parallel or distributed processing system, consisting of a collection of interconnected stand-alone computers cooperatively working together representing a single, integrated computing resource.

**Distributed computing paradigm** consists of a collection of independent computers,each having its own memory and other capabilities cooperating with each other to solve a problem. The participating computers communicate with each other through message passing and the group appears to be a single coherent system to the users.

Let us compare and contrast distributed computing and cluster computing paradigms. While cluster is a collection of tightly coupled independent computers, often kept in the same room, that work in cooperation with each other to achieve a goal with the help of specialized software, distributed computing has no such restrictions. In a distributed paradigm, geographically dispersed independent computers can communicate using message passing in a loosely coupled environment. What is common in both is the ability to achieve the power of a supercomputer by harnessing the power of a large number of workstations.In reality, cluster computing is a special case of distributed computing.

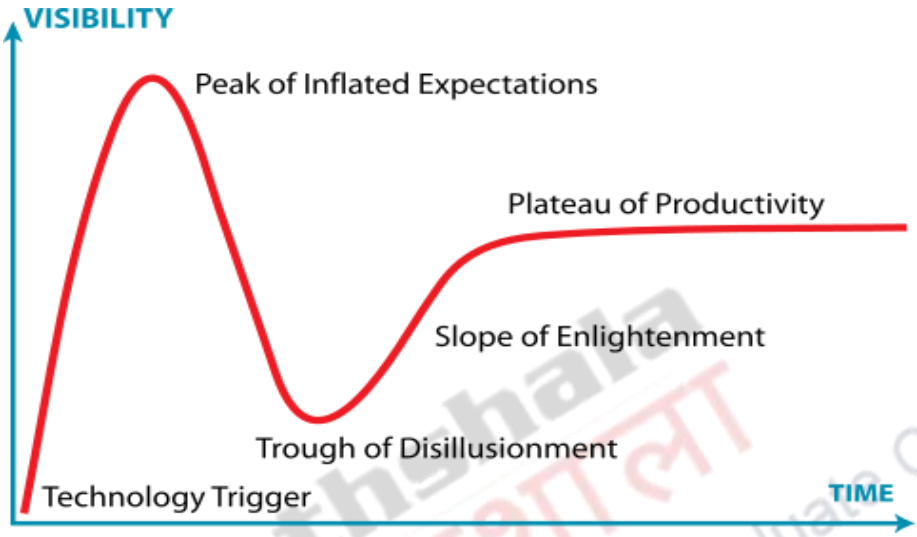
Last but not the least is **cloud computing paradigm**, which is a specialized distributed environment. It is an internet-based paradigm, with shared resources that is capable of providing services on-demand. Cloud helps achieve the concept of utility computing in which computing services is offered as an utility in a similar manner as any other utility services such as electricity.

Even though cloud is a special kind of distributed computing environment, it is interesting to track down the path through which it came about. Web hosting is the mechanism of providing space in the World Wide Web to host one's server(s). Users could be Application Service Providers (ASP-s) who offered applications as service delivered to a rather smaller number of users. However, such service offerings suffered from the problem of scalability, which was mitigated by the use of virtualization. This made this model more effective and financially viable andgave rise to the concept of offering software servicesto a larger number of users giving birth to Software as a Service (SaaS). Eventually SaaS grew in popularity due to the ease of use, giving rise to what is today known as cloud, where not only software, but also platform and infrastructure are offered as a service, called Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) respectively. In the current cloud technology, however, it is possible to provideanything and everything as a service, which gave rise to the term Anything-as-a-Service (XaaS) being widely used.

### 4. Gartner's Hype Cycle

With the continuously changing backdrop of computing, it is important to track the trends and make relevant predictions that would help the industry, researchers as well as the users stay abreast of the advancements coming in the near and far future. In this respect, a very dependable opinion and forecast is provided by the company Gartner. They provide insight with respect to all aspects of information

technology. Among other methods of prediction, Gartner uses a graphical presentation called 'The Hype Cycle' to represent the maturity, adoption and social application of new and emerging technologies. As goes without saying, the journey of cloud from a fledgling SaaS to full-fledged technology also attracted a lot of the predictions and opinions. Industry experts as well as researchers expressed their opinions about the possible future of cloud, as they thought applicable. To understand the pointers provided by Gartner, and to match it with reality, first let us understand the Hype Cycle by Gartner and widely used by the IT industry. Figure 1.4 below shows the basic structure of Hype Cycle.




Courtsey: Gartner[1]

**Figure 1.4: Gartner's Hype Cycle**

The Cycle is divided in five phases, the first one being the Technology Trigger. A technology is positionedin this phase when a new and potential technology comes into the horizon. A technology here indicates that there exists some proof-of-concept and the technology has generated significant interest. While no usable products are expected from a new technology in this phase, there must be enough promise from it.

The next phase if Peak of Inflated Expectations. While a powerful appearance of a technology produces sufficient interests among the industry users, it is highly possible that the concept may fail. This happens especially if the technology generates a very high expectation it may not be able to meet and hence fails completely. Every emerging technology goes through this test of expectation and has to survive to emerge in the next phase.

Trough of Disillusionment is the third phase when through experiments and implementations, users taste an initial disappointedabout the technology. At this phase, the producers of the technology get the verdict, whether the technology would survive or be removed. Investments continue only if the surviving providers who continued to believe in the new technology can improve their products to the satisfaction of the users who have tried adopting the technology initially.

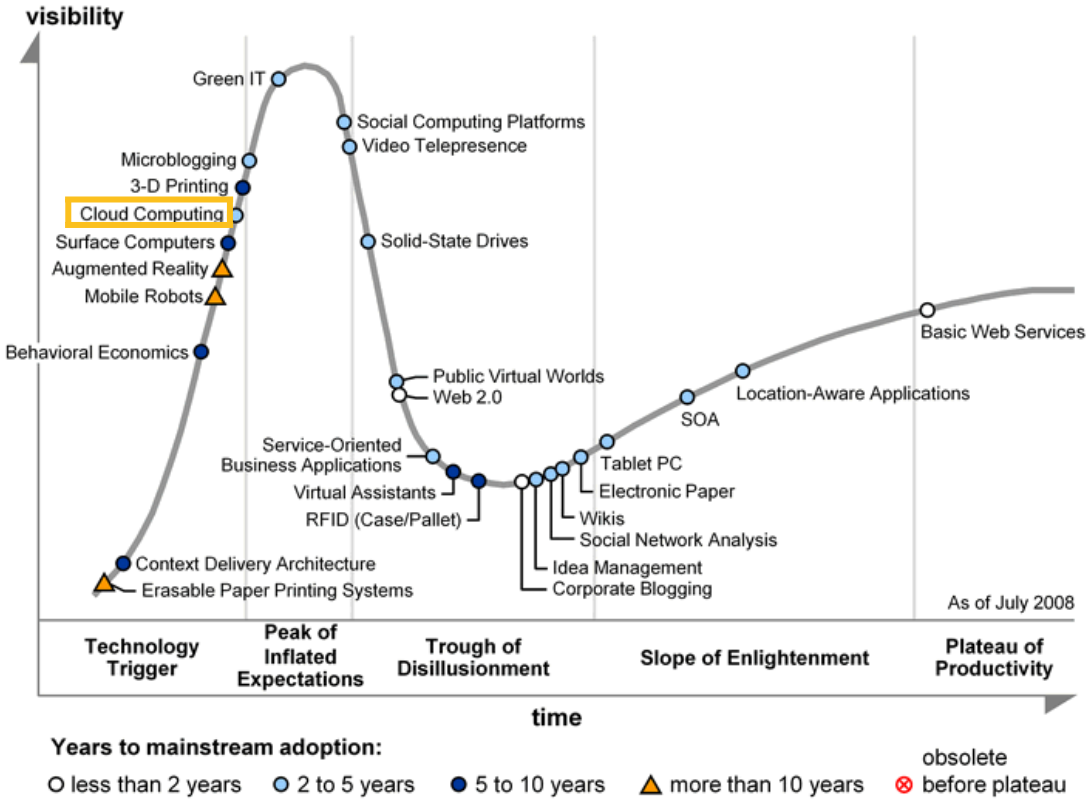
A survivor through the Trough of Disillusionment phase would emerge in the Slope of Enlightenment. In this phase, the technology is believed to be good ready to be adopted by many and hence more funding is channelized. New products as well as

new support appear in the market, although the adoption is not yet complete since the more cautious companies still stay out.

The last and final phase is the Plateau of Productivity. This phase witnesses adoption of the technology at all level. Products mature and providers that pass through the assessment of the matured users continue to flourish. This phase indicates that the technology has matured and is going to be used widely as applicable.

As shown in Figures 1.5 - 1.11, special symbols mark the number of years predicted for an emerging technology to reach maturity and market adoption. The hollow circles indicate technologies that are predicted to be adopted by the industry in two years, the gray circles in two to five years, the solid circles infive to 10 years, and the triangles denote those that would require more than 10 years. The crossed circles represent technologies that are predicted to become obsolete before they reach the plateau.

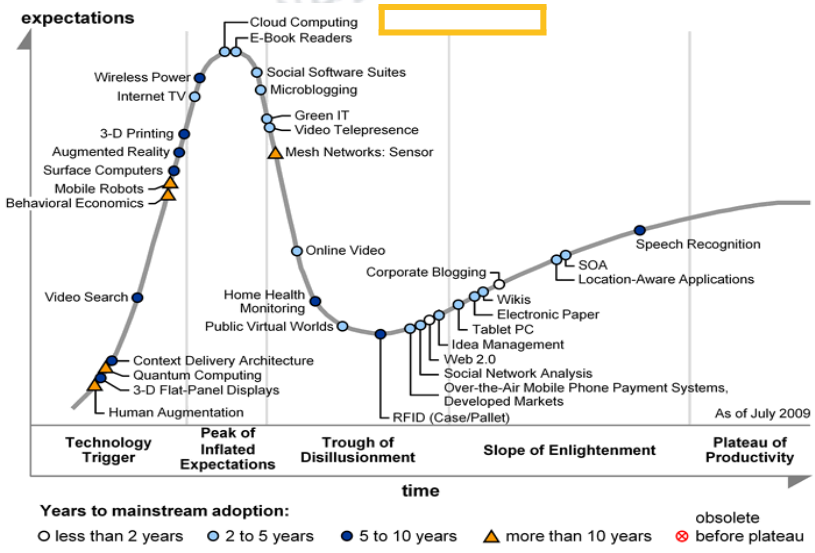
Gartner publishes many articles, graphs, charts and other documents every now and then for the information and knowledge of the users and practitioners of various technologies. Hype Cycle for Emerging Technologies is published every year by Gartner indicating the current position of all upcoming technologies and Gartner's prediction about their future adoption by the IT world. All new and emerging technologies appear in the Hype Cycle based on two perspectives: their position in a specific phase of the Hype Cycle indicate the relevance of the technology and the accompanying indicator describes the number of years the technology would still take to be adopted in the mainstream, if at all. With this knowledge about the Hype Cycle, let us now turn to cloud computing and see how it faired as an emerging technology. Cloud computing first appeared in the Hype Cycle for Emerging Technologies in the year of 2008. As shown in Figure 1.5 below, cloud computing (inside yellow box) was in phase 1 with a promise of coming into the mainstream in 2-5 years indicated by the gray circle.




Courtesy: Gartner[1]

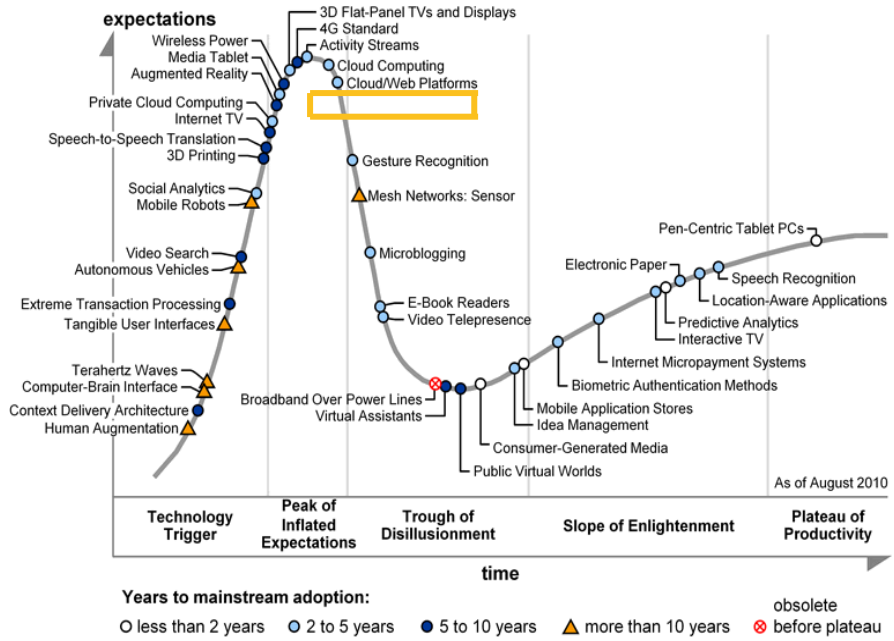
**Figure 1.5: Hype Cycle for Emerging Technologies, 2008**

The next two years saw cloud computing in the Peak of Inflated Expectations as shown in Figures 1.6 and 1.7 below (inside yellow box). However, it started its downward journey through the next four years, as is seen in Figures 1.8 through 1.11.




Courtesy: Gartner[1]

**Figure 1.6 : Hype Cycle for Emerging Technologies, 2009**




Courtsey: Gartner[1]

**Figure 1.7: Hype Cycle for Emerging Technologies, 2010**

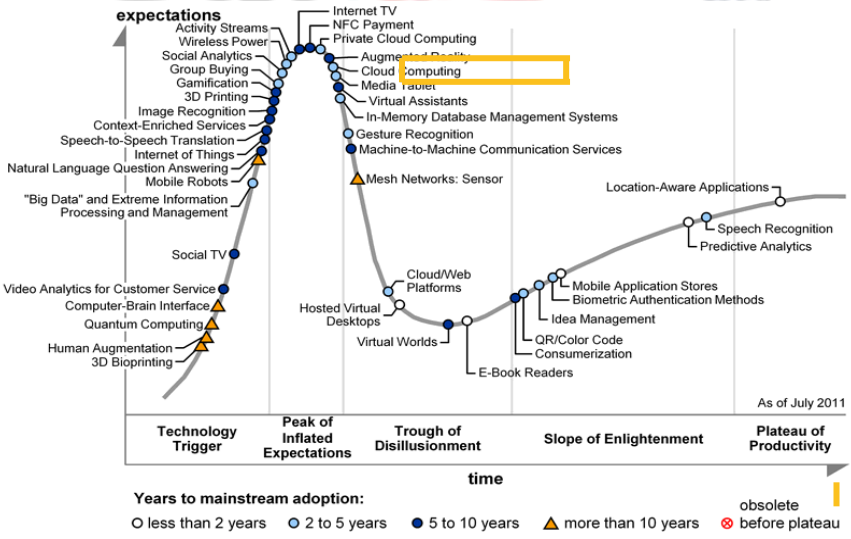
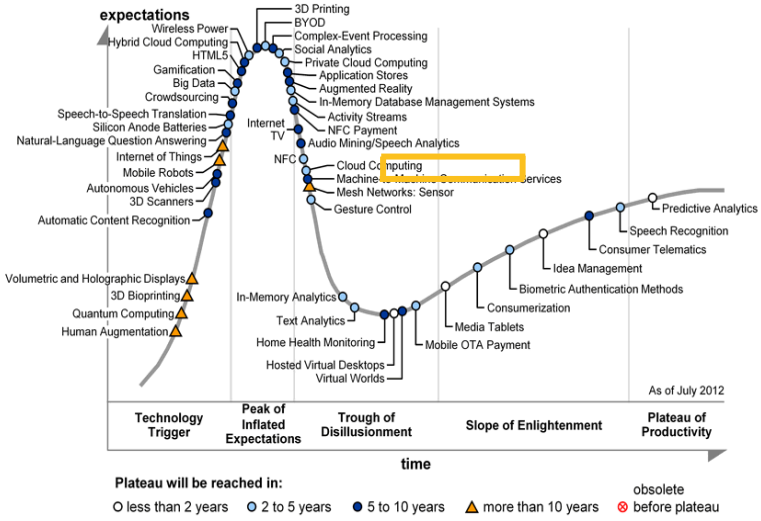
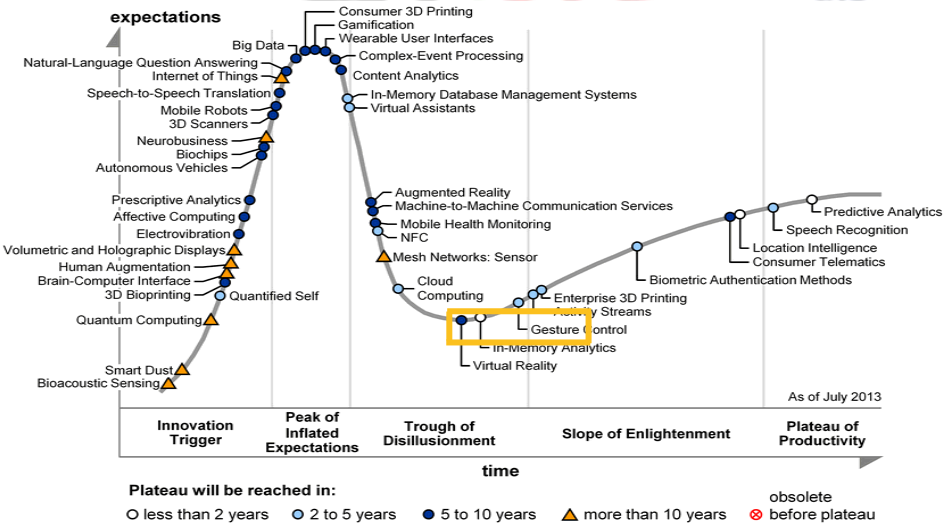



**Figure 1.8: Hype Cycle for Emerging Technologies, 2011**



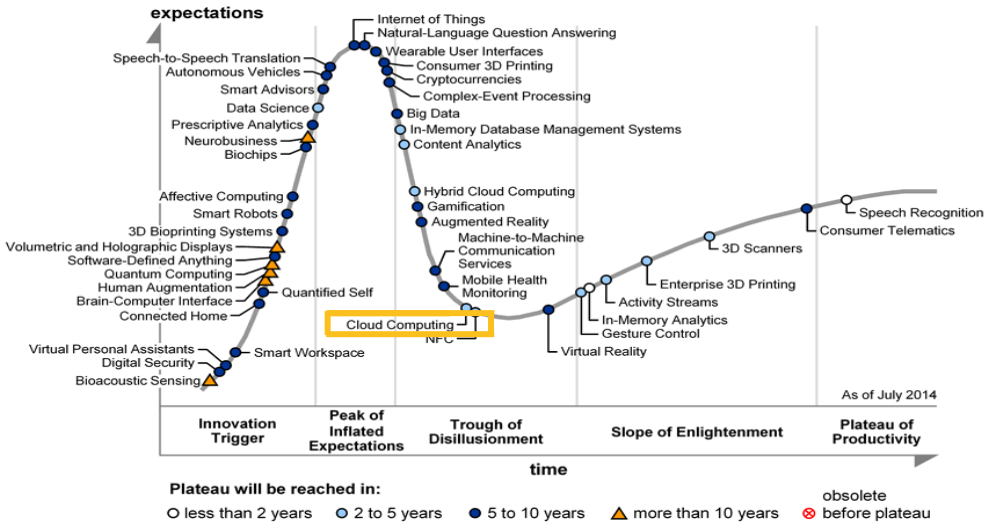

Courtsey: Gartner[1]

**Figure 1.9: Hype Cycle for Emerging Technologies, 2012**




Courtsey: Gartner[1]

**Figure 1.10: Hype Cycle for Emerging Technologies, 2013**




Courtesy: Gartner[1]

**Figure 1.11: Hype Cycle for Emerging Technologies, 2014**

The point to note is that even in the year of 2014, the prospect of cloud computing still is that it will take another 2-5 years to mature completely and be adopted in the mainstream.

It can be noted that in all these years, there are many technologies that appeared in the Hype Cycle and disappeared, while cloud computing has remained and is still growing. This can also be observed from the multiple large organizations that have offered products in cloud. We conclude this unit by taking a quick look at organizations that have lead the way.

Some of the current leaders in the cloud market are shown in Table 1.1 below. Amazon offers its cloud platform called Amazon Web Services, more popularly known as AWS, which is an IaaS platform,since 2006. Microsoft, another big name in the arena of cloud, has offered their platform Azure since the year of 2009. Google has a large presence in the cloud world but is particularly popular among cloud practitioners for their Google App Engine (GAE). GAE was launched in 2008 and allowed free access to their PaaS services for developing web applications. IBM launched Blue Cloud in the year of 2008 as PaaS.

**Table 1.1: Key Players in Cloud Computing Platforms**

| **Company** | Cloud-Computing Platform | Year of Launch | Key Offering |
|---|---|---|---|
| | | | |
| **Amazon.com** | AWS (Amazon Web Services) | 2006 | Infrastructure as a service (Storage Computing), Datasets and Content Distribution |

| | | | |
|---|---|---|---|
| | | | |
| **Microsoft** | Azure | 2009 | Application platform as a service (.Net, SQL data services ) |
| | | | |
| **Google** | Google App Engine | 2008 | Web Application Platform as Service |
| | | | |
| **IBM** | Blue Cloud | 2008 | Proprietary 4GL Web application as an demand platform |

Since cloud is a technology that has very high commercial value, it is important to know the major players in this field. The short discussion in this module of the important offerings by the major companies gives just a little flavor of the the market of cloud. We'll see in details the various offerings of these companies in later units.

**5. Summary**

To summarize this module, we first discussed the evolution of computing devices and the Internet. Then we moved on to explain various computing paradigms. In this discussion we understood the basics of cloud computing as well. We then elaborated the meaning and importance of Gartner's Hype Cycle and discussed the prediction of Gartner about cloud computing starting in 2008 till date. We completed the discussion by mentioning a few popular offerings from some of the large companies dealing in cloud.

## References


1. Peter Mell& Timothy Grance, "The NIST Definition of Cloud Computing", http://csrc.nist.gov/publications/PubsSPs.html#800-145, Sept, 2011.
2. www.gartner.com

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 18: Concurrency Control
# Module No: CS/CC/18
# Quadrant 1— e-text

## 1. Introduction

In the last module, we investigated how to take and use snapshots in a cloud computing environment. In this module, we will take up another important distributed system aspect called concurrency control. Essentially, the whole of a distributed system, and hence cloud, is about accessing data and computation remotely. When there is a remote access, quite obviously allowing multiple such remote accesses would improve the performance of the system. When we allow multiple servers to connect and access data concurrently, we need to ensure that such concurrent accesses are not causing the data to be wrong. The mechanism to do this is called concurrency control, the topic of this module.

## 2. Learning Outcome

At the end of this module, students will be able to:

1. Understand what is the impact of RPC on concurrency control.
2. Learn about the normal transactions and distributed transactions.
3. Understand the ACID properties of transactions.
4. Be aware of the issues of isolation and consistency.
5. Appreciate the problems of concurrent access.
6. Gather knowledge about locking mechanism and its use in a distributed system.
7. Learn what is two-phase locking.

## 3. Revisit the Remote Procedure Call (RPC)

One of the most important building blocks of any data access in a distributed system is RPC. Hence when we want to discuss concurrency control over simultaneous access to shared data, we first take a quick relook at RPC from the perspective of this topic. Particularly since RPC is an important abstraction that is implemented in many distributed systems today and most of the cloud computing systems. Most communication in cloud take place using RPC, an important abstraction that allows processes to call functions in other processes.

We have discussed the basic structure of RPC in our first distributed communication discussion in Module 7. Here, in this module, we will elaborate that discussion with focus on certain aspects of RPC that we have not been discussed in the earlier model. In RPC, two processes, the caller and the callee, can communicate across the

boundaries of machines. In the perspective of cloud, this means that two processes in different datacenters in geographically different locations can easily communicate using RPC. And since RPC does not impose any restrictions on the types of machines the caller and calllee reside in, the datacenters can even belong to two different clouds, as far as the RPC-based communication is concerned. This gives the implementers of cloud a lot of flexibility. Now, let us look at the semantics of RPC.

### *3.1. RPC Semantics*

What is a communication semantic? It is a set of properties of a communication protocol. Communication is assumed to be between a sender and receiver and semantics depend on the arrangement of acknowledgments and retransmissions used in ensuring that the messages sent by the sender is correctly received by the receiver in a communication protocol. Three semantics are commonly used in distributed communication. These are as follows:

At-most-once semantics
At-least-once semantics
Exactly-once semantics

In at-most-once semantics, receiver process either receives a message once, or does not receive it. These semantics are obtained when a sender process sends a message only once and the process receiving a message does not send an acknowledgment. In this semantic, the sender process does not perform retransmission of a message.

In at-least-once semantics receiver process is guaranteed to receive a message albeit several times. In this semantic a sender has to retransmit a message after sending it, till it has not received an acknowledgement from the receiving process before a time-out of every retransmission.

In exactly-once semantics receiver process receives a message only one time and that is the reason the semantic is named exactly once. In this semantic sending of acknowledgments and retransmissions are performed as in at-least-once semantics but the communication protocol must apply an additional mechanism to remove copies of the same messages received at the receiver end and discard these so that the receiving process receives a message only once.

Of the three possible semantics, exactly once is the most desirable one. Of course it is also the most difficult to achieve in a distributed environment. In RPC, we strive to obtain exactly-once semantic since RPC has been devised mimicking the behaviour of local procedure call or LPC, which exhibits exactly once semantic. LPC is the communication behaviour of two processes in the same machine. Since the processor and the memory are the same in LPC, it is easy to have exactly once semantic. However, messages of RPC have to travel through an unreliable network, the sender and receiver process are in different machines with different failure rates, they do not even have a shared memory. Under such circumstance, it is hard to guarantee exactly-once semantics in RPC.

Perhaps the hardest in RPC is the presence of failures. Starting from the messages being dropped by the network, to callee process failing every possible failure would

affect the correct execution of RPC a difficult task, though not impossible.

In a distributed system, the largest use of RPC is in implementing distributed transactions, which are at the heart of a distributed environment, especially cloud. Let us look at what we mean by transaction.

## 4. Distributed Transactions

A transaction is a series of operations between two processes that is treated as a unit for a certain purpose. Even though there are multiple instructions/operations that constitute a transaction, a transaction may either be completed along with all these instructions/operations and all databases are updated or be aborted in such a way that it leaves no trace of its partial execution. There cannot be an intermediate state in a transaction. A transaction whose operations are performed by different processes, which are distributed geographically, is called a distributed transaction.

### *4.1. Transaction Properties*

A distributed transaction has certain properties, the same as a normal transaction. These are called ACID properties. ACID is an acronym as explained below.

A stands for Atomic. The atomic property is the all or nothing property of a transaction. There can never be a partial result of a transaction, distributed or otherwise. Let us take an example of student records with marks of four subjects, along with roll number and address. The task is to aggregate the marks for each student and add the aggregated marks in each record. Let there be 10,000 such records in a cloud environment. Since the environment is cloud, it is expected that all the records are not going to be in the same data servers. Now if this task is executed as a distributed transaction, then the atomicity demands that either the aggregation of marks of the four subjects for all the 10,000 records would be completed and the databases updated or no record would be updated.

C stands for Consistent. It ensures that if database was consistent before the transaction started, it continues to be consistent after the end of transaction.

I stands for isolated. It indicates that even if multiple transactions may be going on concurrently on the same data, the normal execution of one transaction should not interfere with another.

D stands for durable. It ensures that the changes made by a transaction on successful completion will remain in the databases and must not be overwritten till some other correct process changes them.

It is important to note that the ACID properties of a transaction cannot be violated. However, in a distributed transaction, the highest probability of violation of the properties come due to concurrent transactions being executed simultaneously on the same data. When two different transactions are executed by two different processes (most probably in different data centers) access the same data object, maybe in some other geographic locations, this is concurrent access and the solution lies in concurrency control.

In the next section we will see some well-known problems of concurrent transactions

and hence the justification of why concurrency control is needed.

### *4.2. Problems of Concurrent Transaction*

Three well-known problems may occur due to accesses to data by concurrent transactions. These are:

Lost update problem
Inconsistent retrieval problem
Incorrect read problem

#### *4.2.1. Lost Update Problem*

The lost update problem is illustrated by the following pair of clients in Delhi (clients 1) and in Chennai (clients 2) accessing the shared object 'Marks' in server in Mumbai. Initially the server has a value of Marks = 10. If Delhi server executes first and waits for the Chennai server to execute next and then both commit, the database at Mumbai would be kept consistent with a value of Marks = 30, as shown in figure 18.1. Similarly, the other order would also work.

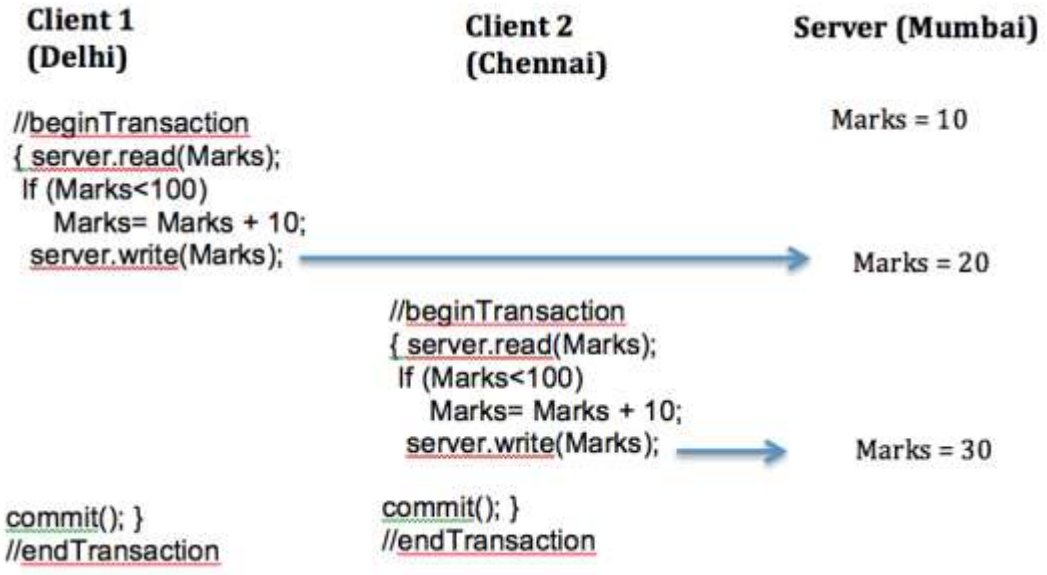



**Figure 18.1: Two Transactions - Example**

However, when these two transactions work concurrently, at the end the value at the Mumbai server is not what it must be as shown in Figure 18.2. This is called the lost update problem.

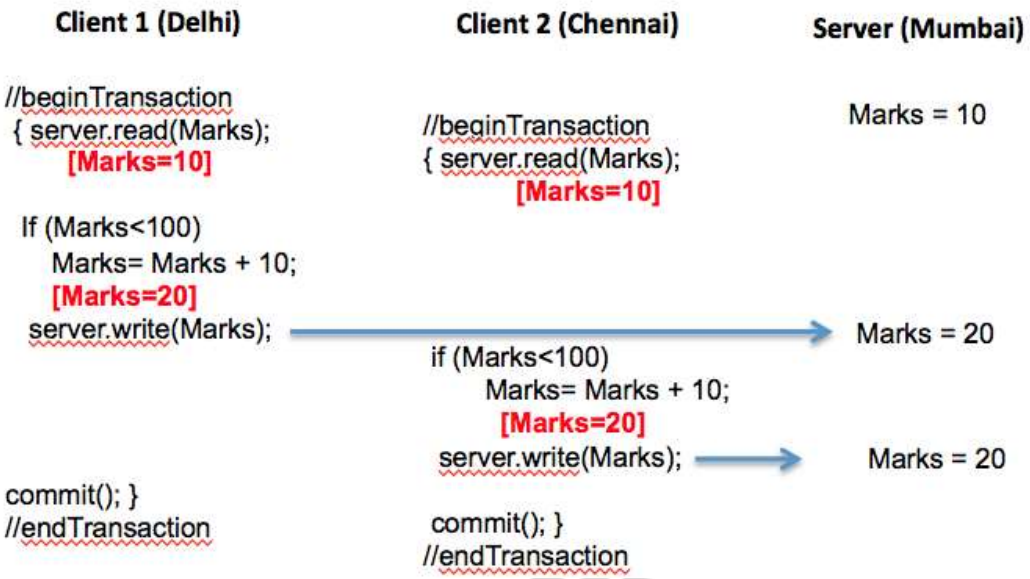



**Figure 18.2: Lost Update Problem**

*4.2.2. Inconsistent Retrieval Problem*

To understand the next problem of concurrency, let us take the same example. Now let us assume that the Mumbai server has an initial value of Marks=90. Delhi server would first execute and add 10 marks making the total to 100. However, by the time the Chennai server applies the code on the data, it Delhi server should have updated the data and the Chennai server should have skipped the 'if' part completely. But because the Chennai server retrieves a data from the Mumbai server, which is inconsistent at this time since the Delhi server has already changed the data but updates are yet to come. This is an example of inconsistent retrieval problem.

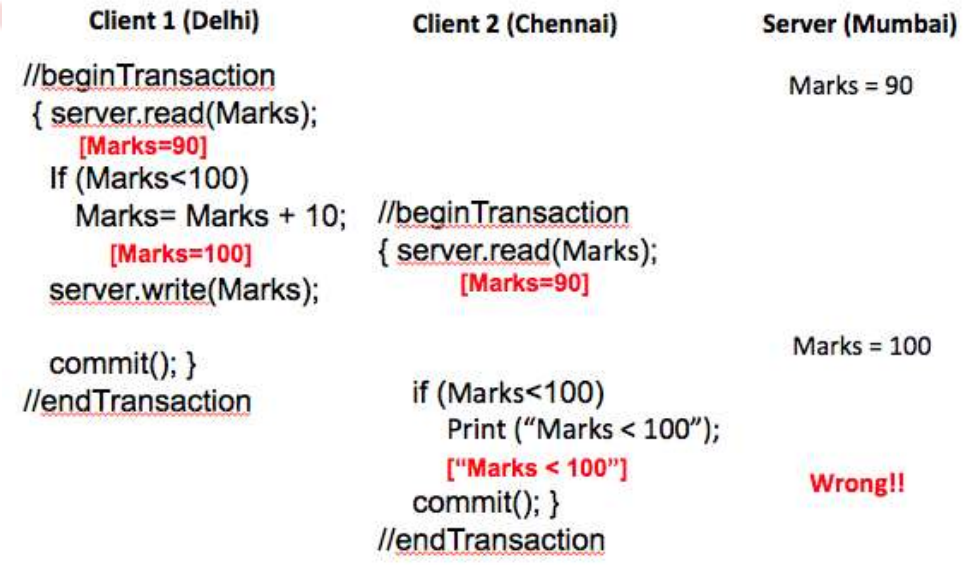



**Figure 18.3: Inconsistent Retrieval Problem**

### *4.3. Solution – Serial Equivalence of Transactions*

If the transactions execute sequentially, neither of the problems would occur! Problem is caused by the interleaving of the operations of the concurrent transactions. Correct interleaving, in which the combined effect is the same as if the transactions had been performed one at a time in some order is called a *serially equivalent* interleaving. The use of serial equivalence as a criterion for correct concurrent execution prevents the occurrence of lost updates and inconsistent retrievals. The lost update problem occurs when two transactions read the old value of a variable and then use it to calculate some data. This cannot happen if one transaction is performed before the other, because the later transaction will read the value written by the earlier one. Similarly, the inconsistent retrievals problem cannot occur if the retrieval of transaction is performed before or after updating the transaction.

Therefore, serial equivalence is used as a criterion for concurrency control. There are a few alternatives to obtain a serial equivalence of concurrent transactions. Out of all the available solutions, the most appropriate that is practical, simple to implement and effective is locking. In this method, the server executing a transaction sets a lock, marked with the transaction identifier, on each object. These locks would be removed at a later point. With a lock imposed, the transaction alone can access the object. However, other concurrent transactions must wait until the completion and hence opening of the lock by the transaction who has imposed the lock. This is shown in Figure 18.4.


**Figure 18.4: Locking**

When we look at this solution of concurrency control, the most important point that arises is how restrictive is it? Will there be ant concurrency at all, or all the transactions would be in sequence. Completely sequential execution of transactions without any interleaving would be costly in terms of time, especially in a system like cloud. Also would locking uphold the ACID properties of transactions? Let us look at all these in the next section.

### *4.4. Implementation of Transaction Properties*

Let us look at how to implement the properties of distributed transaction. The first property, atomicity, is perhaps the hardest to resolve. We need special mechanism to uphold atomicity of transactions. We will discuss these mechanisms in the next module.

Next let us consider the solution for durability. To implement durability, no major efforts are needed. The implementer of a transaction has to ensure that the resulting data after a transaction has committed is saved in stable storage.

How to obtain solution for Isolation and consistency? These properties ensure that a transaction in progress does not interfere with another transaction in progress hence does not allow concurrent access to shared data. This brings us back to the mechanism of concurrency control, which is locking.

## 5. Locking

To achieve concurrency control, and to ensure isolation and consistency of shared data a distributed system, locking is proposed. The idea used in locking is very simple. Let us call a data item as an object and let us assume that each such object has a lock associated with it. Before a transaction T accesses such an object, it must first ask for a lock. The lock of the object is examined to ensure that already another lock is not present on the object. If no other ongoing transaction holds a lock to the data item, then the transaction obtains the lock. If another transaction is already holding a lock on the data object, then the asking transaction has to wait. This way, only one transaction can access the data item at a time. So we lock a data item or an object. In fact, all shared data for a transaction must belong to this category of lockable object.

However, not allowing one transaction completely to access a shared data while another transaction is in progress, is too restrictive and solutions like this would make a system very slow. In cloud, solutions that affect the performance adversely are not encouraged generally. Also, is it required to have such strict solutions? To answer this question, rather than putting a whole data object under lock, let us ask which operations are really important to be put under lock? In reality, with a data item, there are only two operations that must be locked: read and write. Even all read/write operations are not required to be locked. Only certain the conflicting operations, whose *combined* execution may cause problems are to be put under lock.

If x is a share data object, then the following combinations are conflicting:

read(x) and write(x)
write(x) and read(x)
write(x) and write(x)

In the above list, the read(x) and read(x) combination is missing. Actually, this is a combination that does not require the restriction of locking. Also, if we take two different objects, x and y, then none of the pairs involving one x and one y would need to be locked.

Now, with this understanding, let us look at the model: Before reading and/or writing a

data object O, acquire lock(O). This will not be allowed if another transaction has acquired a similar or a conflicting lock already. After acquiring an appropriate lock, T can perform the related operation(s) multiple times on O. When done (*i.e.* at commit point), T calls unlock(O). One of the waiting transactions (if any), will be allowed to acquire the lock released by T.

**Table 18.1: Read-Write Locks**

| | READ | WRITE |
|---|---|---|
| READ | ✓ | ✕ |
| WRITE | ✕ | ✕ |

There are two types of locks, viz., read lock and write lock. While read lock is non-exclusive lock a write is an exclusive lock. On a data object, multiple non-exclusive locks may be allowed but only one exclusive lock is allowed as shown in Table 18.1.

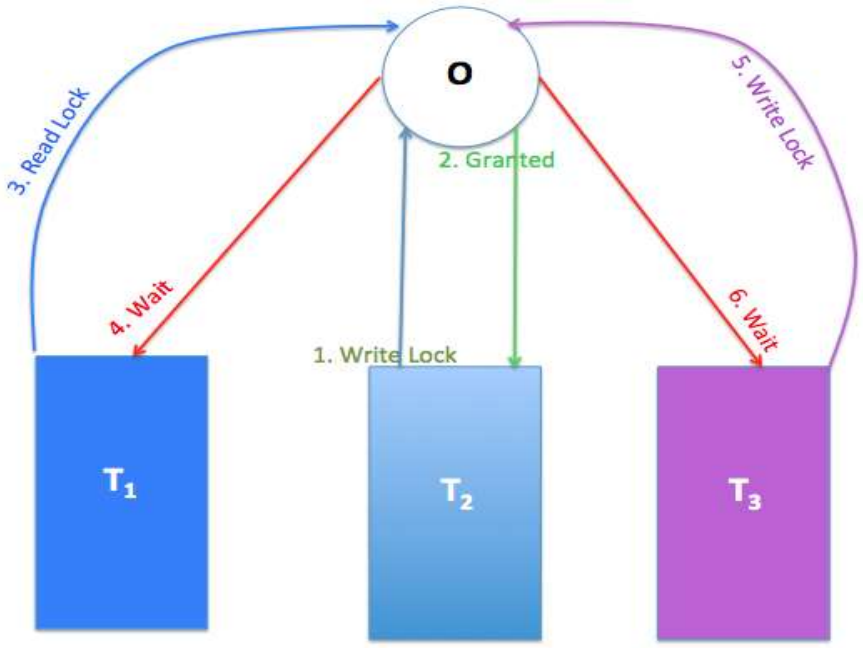



**Table 18.5 : Permitting a Write Lock to $T_2$**

The idea of read and write lock is that we should be able to allow concurrent transactions to access the same item X if they want to acquire only read lock. However, if a transaction wants to write/update an item X, it cannot share access with other transactions, neither read nor write. This is shown in Figure 18.5. That is the transaction must have an exclusive access to X. This is implemented using multiple-mode lock. In this scheme, there can be three types of operations: read_lock(X), write_lock(X), and unlock(X), where x is the data item. While a read-locked item is share-locked, a write-locked item is exclusive-locked, because a single transaction exclusively holds the lock on the item.

### *5.1. Problems of Locking*

A schedule is a list of operations (including locking), ordered by time for a set of concurrent transactions. A correct schedule must possess two characteristics:

Legality: This tells that no conflicting locks have been granted to two transactions.
Serializability: A schedule must possess an equivalent serial schedule, i.e., the result of a schedule is the same as would happen f the transactions were to execute sequentially.
Figure 18.6 shows an example of what is meant by serializability. Let us start with initial values of X=20 and Y=30. If we execute $T_1$ followed by $T_2$, at the end of both the transactions, the result is X=70 and Y=50. On the otherhand, if we have $T_2$ followed by $T_1$, at the end of both the transactions, the result is X=50 and Y=80. Any schedule that provides either of the two results has serializability. However, even in the absence of conflicting locks, it is possible to have a schedule that does not have serializability.

| T1 | T2 |
|---|---|
| read_lock(X) | read_lock(Y) |
| read(X); | read(Y); |
| unlock(X); | unlock(Y); |
| write_lock(Y); | write_lock(X); |
| read(Y); | read(X); |
| Y=X+Y; | X=X+Y; |
| Write(Y); | Write(X); |
| unlock(Y); | unlock(X); |

**Figure 18.6: Two Transactions $T_1$ and $T_2$**

This is shown in Figure 18.7. One inter-leaving of the transaction lock operations, which does not produce the correct result.

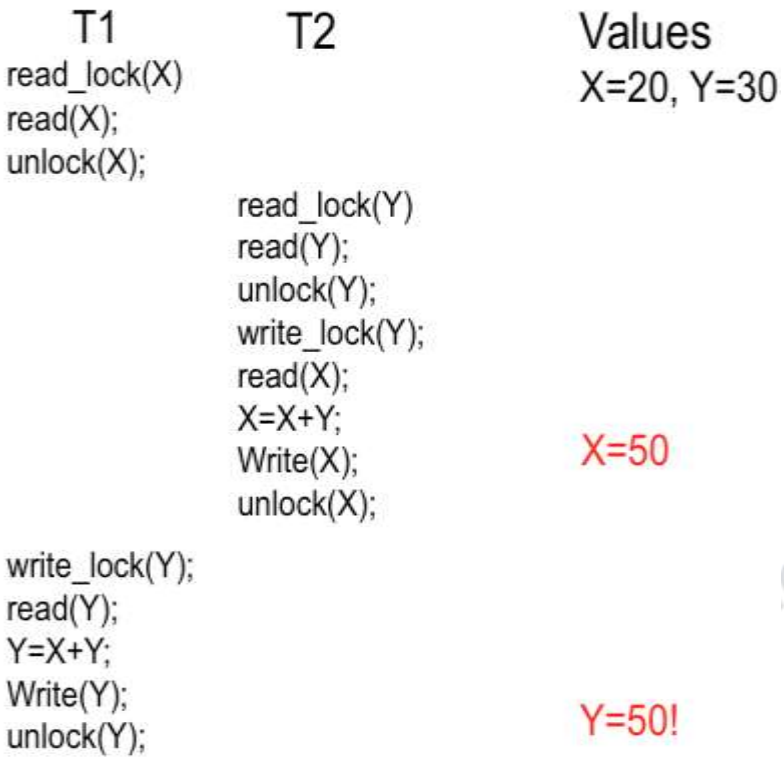



**Figure 18.7: Not Serializable Transactions $T_1$ and $T_2$**

Another problem of unrestricted use of locking is deadlock. A deadlock occurs when a pair of transactions each having a locked object, needs to acquire the lock on another object held in the other transaction. Neither will be in a position to release the lock it has acquired leading to a deadlock. Deadlock is time consuming to detect and is completely undesirable to happen in a cloud. In the next section, we discuss a method of locking that doesn't allow these problems to occur. This is known as two-phase locking.

### *5.2. Two-Phase Locking*

A special case of allowing and permitting locking is done using two-phase locking protocol. In this mechanism, all lock-related operations in a transaction are executed in two phases:

1. Growing phase: the transaction obtains locks.
2. Shrinking phase: the transaction releases locks

All the locks must be acquired in the growing phase and only after the lock acquiring is completed, locks can be released. The lock point is the moment when transitioning from the growing phase to the shrinking phase. This is shown in Figure 18.8.

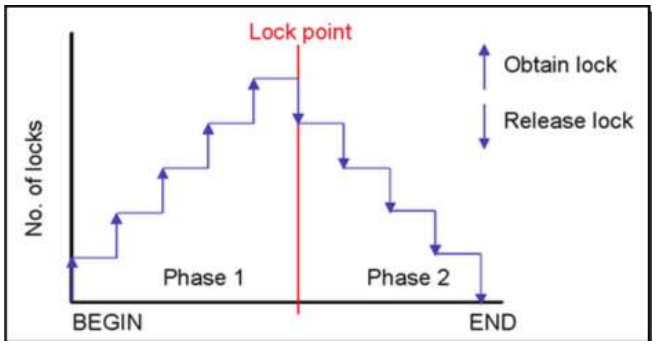



**Figure 18.8: Two-Phase Locking**

This way, a transaction cannot acquire any locks after it has started releasing locks. Two-phase locking protocol (2PL) generates conflict-serializable schedules. When an operation accesses an object within a transaction and a lock exclusively has not been set for the object, it is first locked and then the operation proceeds. Also in 2PL, if the object is trying to acquire a conflicting lock already set by another transaction, the transaction must wait until the other transaction has not unlocked.

The problem of 2PL is that on abort after releasing a lock, a transaction may cause other transactions to abort as well. This is called cascading aborts. A version of 2PL, called strict 2PL holds all the locks acquired till the end of the transaction thereby avoiding cascading aborts.

## 6. Summary

In this module, we first looked at the most important communication mechanism, the Remote Procedure Call or RPC. We discussed what is a distributed transaction and also the properties of a transaction. We looked at the problems of concurrent transactions and discussed these with appropriate examples. We discussed how locking can be a simple but elegant solution for achieving isolation and consistency as well as ensuring concurrency control. However, no solution is free of problems. Locking is no exception and we looked at the possible problems that can occur due to locking. As a measure to control the problems, two-phase locking or 2PL is discussed.

While locking can solve the problems of concurrency while upholding consistency and isolation, it cannot guarantee atomicity of transactions. In the next module, we will look at how atomicity is handled in distributed transaction?

## References

1. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, “Distributed Systems: Concepts and Design”, 5th Edition, Pearson publications, 2011.
2. Tanenbaum, Andrew S., and Maarten Van Steen, “Distributed Systems: Principles and Paradigms”, Prentice-Hall, 2007.
3. Kenneth P. Birman, “Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services”, Springer Science and Business Media, 2012.

4. D. M. Dhamdhere, “Operating Systems: A Concept based Approach”, Third Edition, Tata McGraw-Hill Education, 2012.
5. Singhal, Mukesh, and Niranjan G. Shivaratri, “Advanced Concepts in Operating Systems” McGraw-Hill, Inc., 1994.
6. http://research.microsoft.com/en-us/people/philbe/chapter3.pdf
7. **https://www.cs.uct.ac.za/mit_notes/Database/Jul2007/html/ch09s08.html**

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 22: Virtualization I
# Module No: CS/CC/22
# Quadrant 1— e-text

## 1. Introduction

Earlier, we mentioned virtualization to be one of the two enabling technologies of cloud. In this module we start the discussion on virtualization. While virtualization concept is not new, its reincarnation happened with the advancement of cloud technology and today it has become a widely used mechanism. Virtualization affects all parts of a data center providing cloud-based services. All elements of computing, such as hardware, runtime environments, storage, and networking get affected by virtualization.

Virtualization particularly affects the services that provide IT infrastructure on demand. It is due to this technology that a greater control and more appropriate customization is possible in a cloud environment. Virtualization helps providers by enabling sustainable solutions in cloud by creating different computing environments in a single computer being accessed by multiple users simultaneously. Essentially, virtualization allows presenting a simulated hardware system to an operating system, thereby running multiple operating systems simultaneously on the same hardware, a phenomena called hardware virtualization. This is responsible for the coexistence of more than one stack of software that hitherto could not be run together on the same hardware.

A natural fall out of virtualization is the ability to create multiple virtual machines. This, on the other hand, allows isolation of the different environments, thereby providing security in a virtualized environment. The number of such virtual machines would depend on the server in which they are run but each of these instances allow an optimization on the usage of the hardware. This is the essence of sharing in cloud computing environment. From this ability comes the terms such as elasticity and on-demand. Success of Amazon EC2, VMware vCloud, and many other cloud providers, in fact the success story of cloud computing is largely due to this technology.

In this module and a few modules after this, we will learn about this mechanism in details.

## 2. Learning Outcome

Virtualization is at the core of cloud computing and the understanding of cloud is not completed unless we understand how the magic of virtualization works. The objectives of this module and a few modules after this are to provide a clear understanding of virtualization mechanism what is Virtualization and to learn various mechanisms of Virtualization.

At the end of this module, students will be able to:

1. Understand the concept behind virtualization through the use of a real-life example.
2. Understand and appreciate the need for virtualization
3. Get a clear idea of what is termed as virtualization.
4. Appreciate the architecture of virtualization.
5. Learn about the goals of virtualization.

**3. Problem with Traditional Systems.**

The traditional computing system suffers from many serious to less serious problems. Let us discuss these in this chapter. We use the analogy of a library to understand this.

***3.1. A Conventional Library System***

Let us take an example of a library. Let us consider a typical library consisting of multiple a large number of books under a number of categories such as Engineering, Literature, History, Drama, Fiction and so on. How does one manage this environment?

Let us assume that the entire library is managed by a chief librarian. Each sub-section is managed by individual assistant librarian, who is aware of position of various segments of books and is responsible for the issuing and returning of all the books in that section.

Let us assume that the library has its registered members who visit the library to borrow books from different categories and return back the books after a stipulated period, just as we would expect in a library.

a. Issue in current library scenario

How well does this environment work? Is there any problem or restrictions in the workings of the above-explained scenario?

The described system needs every registered member who visits the library to be aware of which assistant librarian to approach for hiring the desired books. If one is an active member studying wide range of books, she needs to know all the assistant librarians. This situation will be quite difficult for a new member who wants to get books from different categories.

Further, any modification within the library such as merging two categories of books or merging the racks is not possible without causing inconvenience to members for a similar argument. If any such internal modification is done, it has to be informed to all the members and all the assistant librarians, a process definitely tedious.

Also, if on any one day, one of the assistant librarians is not available for the day, how would the head librarian solve the problem? The answer is easy; she will ask another assistant librarian to take-up the additional work of issuing the books on behalf of the absent assistant librarian.Unfortunately, this simple task becomes a challenge for the members who would expect the same sub-librarian to serve them. Intimating the temporary change in the responsibility of the assistant librarian duty to the library members who visit the library on that particular day would be difficult. For each such member identifying and locating the new person would be a tedious job. A similar situation occurs when some assistant librarianis to be permanently replaced by new assistant librarians. The members need to be acquainted with the new assistant librarians thereby causing inconvenience to the members.

Moreover, the time of the sub-librarians are not fully utilized all the times. The average demand of the books in any category will be very less in a day and hence a sub-librarian would be idle for most part of the day, Also, the demand of a certain category of books may increase during a specific time. For example, during board examinations, the demand of text and relevant reference books would be high. This would make the sub-librarian attending to the schoolbook section over-worked. However, under this circumstance increasing the number of sub-librarians cannot be done transparently without causing inconvenience to the users. Also reducing the number of sub-librarians later when the examinations get over, would pose difficulty.

Neither of the tasks of increasing the number of sub-librarians nor the task of decreasing the number of sub-librarians is easy, given the structure of the library.

Figure 22.1 shows the conventional library scenario.

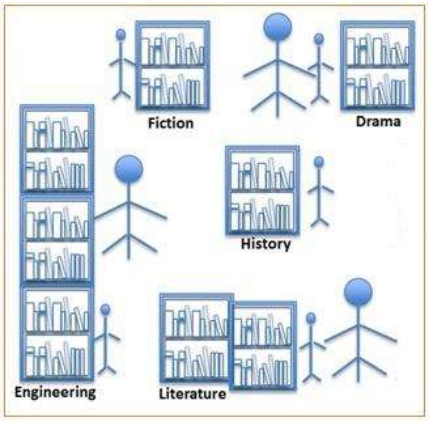



**Figure 22.1: Conventional Library Scenario**

### *3.2. A Traditional Server Room Scenario*

The traditional server room is a place where any organization having medium to large networked environment would keep their servers. This physical space is typically designed for placing and operating servers, storage and network equipment. For larger organizations this room will be fully equipped with IT and non IT facility like primary and backup power supply, air conditioners, surveillance systems, fire safety equipment and so on.

So how would a server room look like?

In a traditional server room, there may be many applications running on heterogeneous environment. Generally the operating environment or typical server room or core functional components are servers, storage and network. The heterogeneous environment may have different types of servers, storage and network equipment employed for different purposes running on variety of platforms as described below.

While a web server may be a rack server type of hardware running on Windows, an app server may be a blade server running on Linux, a database server may be a

Tower server with external storage running on Linux, and an email server may run on any Windows work station.

Storage attached to the servers may be of different architecture such as Storage Area Network (SAN) or Network Attached Storage (NAS) with any RAID configuration.

The network in traditional data center/server room may be controlled by different architecture and devices such as Routers, switches, firewall.

All these components would be further grouped to make different servers for different applications. For example, we can expect any combination of the following servers:

A proxy Server is used to filter requests, improve performance, and share connections whereas a mail server is configured to move and store mail; mails may be over the Internet or a dedicated intranet. Web Server is employed to serve contents to a web browser by loading a file from a disk and making it available to a user's web browser that may be in a remote location. Application server is to run various applications. An FTP server helps in transferring files between far off computers in a secured and controlled environment while a database server hosts the databases and a telnet server that enables users to log on to a remote computer.

The question is, how would these servers be procured by an organization? Would a manufacturer sell a specific server? No. Traditionally, a user organization would buy the hardware, depending on the number of different servers they need, and would configure these hardware as per their need. So the organization may configure various server machines to act as specific dedicated servers for specific causes and traditionally would store these servers in one or more rooms, called the server room. A resulting scenario may look as shown in Figure 22.2.

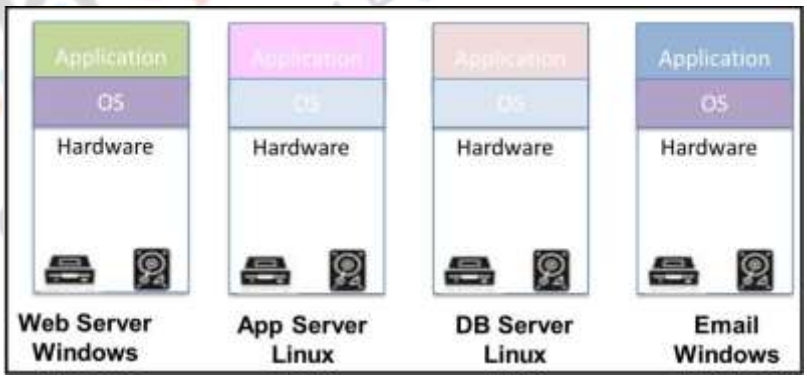



**Figure 22.2: Conventional Server Room Scenario**

### *3.3. A Traditional Server Room vs. A Conventional Library*

The scenario of a traditional server room looks a lot similar like the library scenario. Quite obviously, the problems of this traditional server room are also a lot similar to that of the conventional library organization.

Just like in the library system, the situation in the server room becomes critical if one or more server, storage or network fails. For example, if the mail server fails, we cannot make the web server act also as mail server, since users are unaware of the change and will be unable to log in the mail server.

Moreover, the servers are not fully utilized at all the times. The average utilization of the server, storage and network resources or applications in the server will be very less compared to the capacity of the servers but still all the resources will be running continuously. Also, when the demand of a certain server increases (say the App server due to a need in the organization), increasing the number of physical machines for that service cannot be done transparently without inconveniencing the users. In a similar manner, reducing the number of servers on less-load times also poses difficulty. All these issues lead to under-utilization of underlying servers most of the times thereby wasting the resources and over-utilization of resources at certain times. This leads to poor quality of service.

### *3.4. A Data Center*

Simply put, cloud is like a very large organization with huge computing and storage requirements. Hence we can draw an analogy between a server room and a data center. Quite logically expect a data center, which is the heart and brain of cloud, is also be organized like a big server room. In fact, data center is an advanced server room in fully secured environment with backup electricity, cooling, customized floor and roofing, sound proof cabin for placing servers.

Therefore, the problems of a data center are the same as those of a server room, which is similar to that of a library system. Now, let us look for a solution to these problems.

### *3.5. A Solution*

So now we turn towards a possible solution to the problems that are faced by our systems that we have described.

#### *3.5.1. Solution for the Library Scenario*

Some typical issues in the library scenario include difficulty in identifying an assistant librarian who can manage different categories of books, finding suitable alternatives for handling the additional load for certain category of books, developing a mechanism by which the members be updated about changes in the assistant librarians etc. Even upgradation of the library system would also similarly suffer from such difficulties. An optimal solution for such issues in the library scenario is to place a front desk in the library, which has the details of all the categories of books with an assistant librarian taking charge of this desk. The detail is as shown in Figure 22.3.

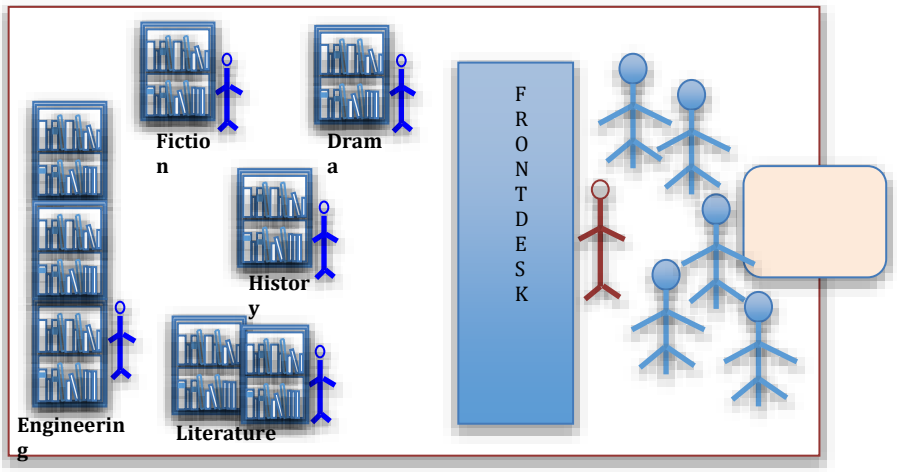



**Figure 22.2: Front desk in Library**

In this arrangement, the members will approach only the front desk for their desired books and need not approach the sub-librarians directly, which makes it far easier for the users accessing the library facilities. The front desk will, in turn, pass on the requirements to the sub-librarians who now are behind the Fron Desk and hence not in direct contact with the members. The placement of Front desk between the sub-librarians and the members will eliminate all the difficulties for both the librarians and the members. Any alteration in the book category or in the sub-librarian will not provide any inconvenience to the members and all such situation will be handled by the front desk, which makes the management and maintenance of library much simpler.

### *3.5.2. Solution for the Server Room Scenario*

Can we apply a similar solution in the server room problem by placing a layer of indirection? In the server room/data center scenario a simple alternative to the front desk would be to run two or more services in one machine thereby solving the under-utilization problem, as shown in Figure 22.4. However, this solution suffers from two aspects. The first is that it does not resolve the over-utilization/overload issue. In case one of the applications becomes more demanding, this compact server running multiple applications in one machine will definitely not be able to handle the load. The second problem is that the proposed solution suffers from limited implementation scope. Let us understand this point.In the server room/data center scenario, different services may need different operating systems, as seen in the Figure 22.2 with email and web server running on Windows whereas the App sever and the DB server running on Linux.Combining these services in one machine would require both the operating systems running concurrently in the same machine. Unfortunately, the traditional architecture of servers does not allow multiple operating systems to run concurrently in the same machine, as shown in Figure 22.5.

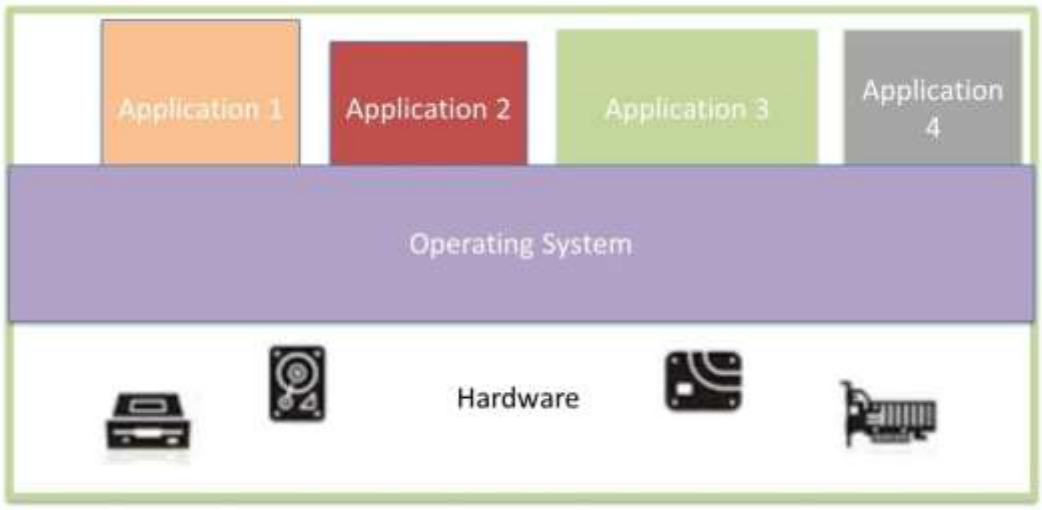



**Figure 22.4: Services/Applications in one Machine**

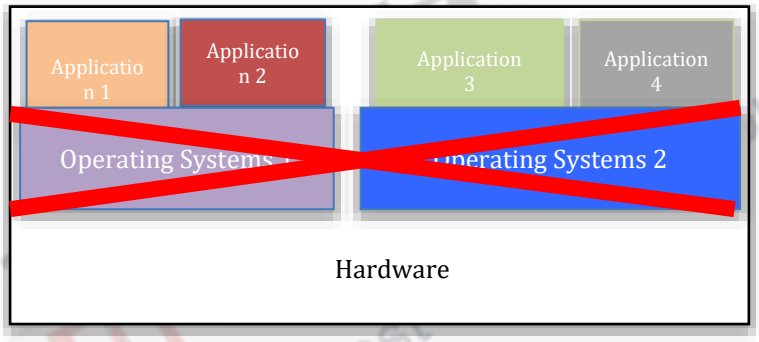



**Figure 22.5: Running Multiple OS on One Machine**

This constraint will restrict the grouping also very severely, since only services running on the same operating system can be run on the same machine.To understand why multiple OS-s cannot be run simultaneously, let us understand the layered architecture as is used in our computers.

*3.5.3. Layered Architecture*

Our computer systems have layered architecture with lowest layer being the hardware layer and the top most layer is the application. The hardware layer comprises of Processor, Mother Board, Memory – RAM and ROM, Storage – Hard disk, Network hardware – Network interface card, Input & Output devices. Application layer runs the applications. Operating system (OS) runs in between the hardware and the application layer. OS interacts with the user applications and executes the instructions from the applications on the hardware. The need for the OS layer is to reduce the complexity of the higher layer programmers by taking away the inherent complexity of handling the hardware directly. However, even handling the complete hardware by the OS is not feasible due to the existence ofa variety of hardware. Hence OS also needs assistance to understand and interact with the underlying hardware. This assistance is provided by the Instruction set architecture (ISA). ISA is an abstraction of the abilities of the underlying hardware. However, since this ISA is what interacts with the hardware on one side and an OS on the other, OSs arecompiled for specific ISA-s. We cannot load and execute a Windows kernel compiled for x86 onto another ISA, e.g., PowerPC or SPARC. Hence on top of one

ISA, only one OS can run that can interact through the ISA to the underlying hardware. The position of ISA is depicted in Figure 22.6.

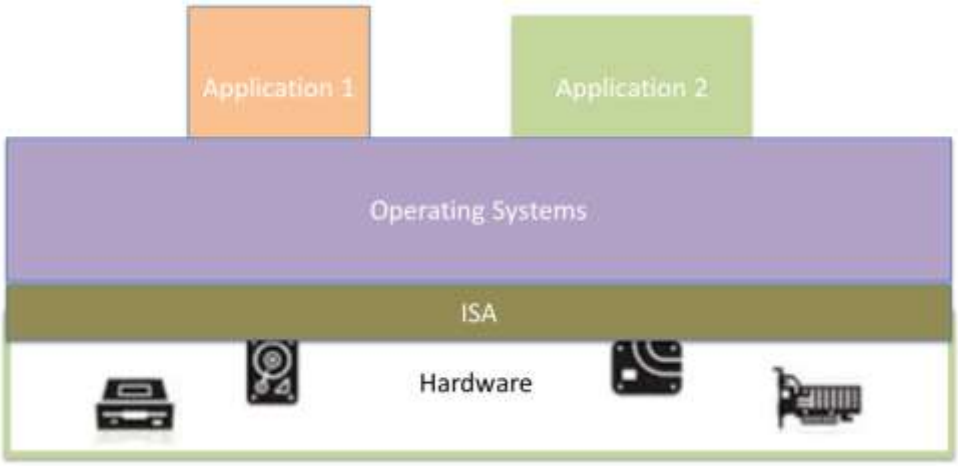



**Figure 22.6: Layered Architecture of Machine**

*3.5.4. Solution of Running Multiple OSs*

As explained in the earlier sub-section, since it is not possible to run multiple operating systems in a machine due to the restrictions in the architecture, how do we add the front desk-like layer in server room /data center scenario to solve the problem of under-utilized resources? The necessary solution of indirection is implemented by placingan interface layer between the server hardware (Processor, memory, storage disk, network interface) and the operating system. This is called virtualization, as is explained in detail in the rest of this module and in the forthcoming modules.

**4. Virtualization Technology**

The additional layer mentioned in the earlier sub-section solves the issue of under-utilization of resources, making the task of maintenance and management much easier. This additional layer is called Virtualization layer. This virtualization layer comprises the software deployed on the hardware, using Virtualization technology.

Virtualization technology divides and combines the underlying resources of server to execute one or more operating system simultaneously and independently using the combination of various methodologies such as hardware and software partition or aggregation, partial or complete machine simulation/ emulation, time sharing and many other methods.

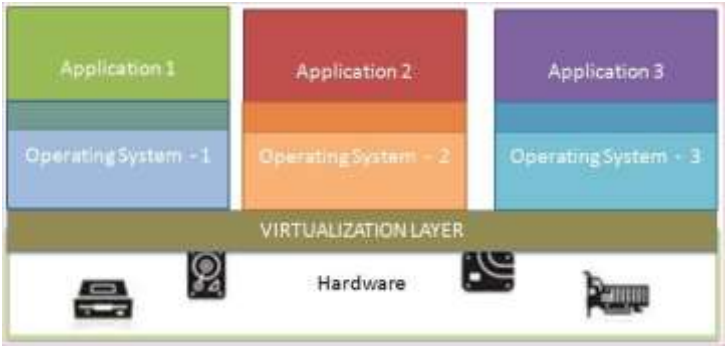



**Figure 22.7: Virtualization Scenario**

### *4.1. Definition*

Virtualization can be defined as follows:

"Virtualization is a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and many others"

### *4.2. Virtualization*

Essentially, virtualization provides a way of relaxing the constraints provided by the dependencies of operating systems on the hardware and the applications on the operating systems. It is a way to abstracting away the underlying hardware and presenting a logical or virtual view. It is worthy to note that this logical or virtual view may be very different from the actual physical resource.

Let us understand the phenomena called virtualization from the Figure 22.7.

With Virtualizations layer software deployed on hardware, more than one operating system of different variety can be executed simultaneously as can be seen in the Figure. Each and every operating system running on top of the virtualization layer will virtually have its own isolated hardware. The virtual hardware for the virtual machines comprises the followings:

1. Virtually partitioned and virtually isolated processor and its corresponding core
2. Virtually partitioned and virtually isolated memory (RAM)
3. Virtually partitioned and virtually isolated storage (Local hard disk or external storage)
4. Virtually partitioned and virtually isolated network interface card.

This hardware for every operating system is virtual, which means that the real hardware is only one on which the hypervisor is running and all the OS above virtualization layer. However, every OS above the virtualization layer considers that it runs on a dedicated hardware. This is why these are called virtual machine.

Each of the OSs on top of virtualization layer along with its virtual hardware and application(s) is called a ***virtual machine (VM)***. Application is to be executed on top of the OS in a VM. The virtualization environment with VMs can bring in optimal

utilization of the resources. In case of any operation maintenance activity, the VMs can be easily migrated to other server hardware.

### *4.3. Virtualization Terminologies*

The software responsible for bringing in virtualization technology is called Virtual machine monitor or Virtualizing software or Hypervisor. The operating system running on physical machine is termed as host or base operating system. The operating system running on virtual machines is termed as guest operating system. The virtual machines comprising of guest OS, virtual hardware is also termed as instances.

### *4.4. Goals of Virtualization*

The primary goals of virtualization are as follows:

1. Optimal utilization of the resources – Single server is used to run multiple varieties of OS simultaneously thereby reducing the idle time of the hardware and underutilization of the hardware.
2. Allows any network-enabled device to access any network-accessible application over any network, even if that application was never designed to work with that type of device.
3. Isolate one or more workload (referred as VM) along with its application from another VM along with application to enhance the security and manageability of the entire environment.
4. In spite of isolation of VMs among themselves, the applications are isolated from the OS, and allow for execution of the application on VMs even though the application was designed for different version of OS.
5. The number of users accessing the application on virtual machines can be increased to larger extent since multiple instance of the workload/VMs withapplication can run on same hardware or multiple hardware simultaneously.
6. Decrease the time it takes for an application to run, by segmenting either the data or the application itself and spreading the work over many systems
7. Increases the reliability or availability of an application or workload through redundancy (if any single component fails, this virtualization technology either moves the application to a surviving system or restarts a function on a surviving system)

### *4.5. Hype of Virtualization*

Virtualization technology has hype because of the following factors.

1. The average utilization capacity of every machine is increased to maximum.
2. Reduces the number of servers required for running multiple applications thereby reducing the cost.
3. As the number of servers required is reduced; the space required is also reduced.
4. As the number of servers and space is reduced, the required power and cooling parameters will also get reduced. This leaves less carbon footprint leading to green initiatives.

## 5. Summary

Virtualization is a new software layer of indirection is introduced between the hardware and operating system. In this module, we described how virtualization solves the issue of non-optimal utilization of resources and makes the management, maintenance, operation and accessibility of systems, applications easier. Virtualization technology provides optimal utilization of the hardware resources. Virtualization technology is used to virtualize the servers, storage and network which mean the underlying hardware components are divided into multiple virtually isolated execution units. The virtualizing software is called the virtual machine monitor or Hypervisor and is responsible for bringing in the virtualization functionality into the servers. VMM lies between the hardware and operating system and hides the underlying hardware from the operating system or application.

In the next module we will look into the various types of virtualization.

## References


1. Smith, J. E., and Ravi Nair, "Virtual Machines: Architectures, Implementations and Applications", Morgan Kauffmann, 2004.
2. Figueiredo, Renato, and Peter A. Dinda, "Guest Editors' Introduction: Resource Virtualization Renaissance", Computer 5: 28-31, 2005.
3. Hwang, Kai, Jack Dongarra, and Geoffrey C. Fox, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things”, Morgan Kaufmann, 2013.
4. Buyya, Rajkumar, Christian Vecchiola, and S. ThamaraiSelvi, “Mastering Cloud Computing: Foundations and Applications Programming”, Newnes, 2013.

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 23: Virtualization II
# Module No: CS/CC/23
# Quadrant 1— e-text

## 1. Introduction

Virtualization is a necessary mechanism in a data center providing cloud-based services. All elements of a data center, such as servers, storage, and networking are affected by virtualization. While In the last module we've explained about the need of virtualization in the cloud environment, in this module, we will see the fundamentals of virtualization in greater details.

The idea of virtualization is far from being new. In fact as early as the 60's, the concept of virtualization was conceived that would allow multiple users to have full control on the access of the underlying hardware. In 1972, the concept was brought into practice by IBM in the System/370 machine in the form of VM virtual machine operating system. However, virtual machines did not become a generic mechanism till around 1998,which saw VMware proposing strategies to build virtual machines on x86 architecture. We will learn about all these in the forthcoming sections in this module.

Since virtualization is a necessity in cloud computing, we must know the mechanisms of virtualization to really understand cloud. In this module we will look deeper into virtualization.

## 2. Learning Outcome

Virtualization, the core concept in cloud computing is the focus in this module. The advantages and the difficulties of virtualization are the main objectives of this module.
At the end of this module, students will be able to:

1. Understand the concepts of virtualization.
2. Obtain an in-depth knowledge about the advantages of having a virtual machine.
3. Learn and appreciate the core strengths of a virtual machine in the form of the primitive operations performed by the virtual machines.

## 3. Virtualization

Virtualization is a concept used to create a virtual environment on a real physical environment. Thus created virtual environment is virtually segmented and isolated system with the same characteristics as that of real system. The segmented virtual components act as if these are independent systems themselves.

With the virtualization technology, any physical or logical components in an operating environment can be virtualized. The following is a list of components that can be virtualized:

a. Network interface card and its functionalities
b. Hardware
    i. Processor and its functionalities
    ii. Memory and its functionalities
    iii. Storage and its functionalities
c. Software
    i. Operating System – Kernel space and User space, Desktop
    ii. Database, Application

Virtualization can be categorized in different ways and many researchers have done that. We will describe all the divisions, as far as possible, in this module.

**4. Types of Virtualization**

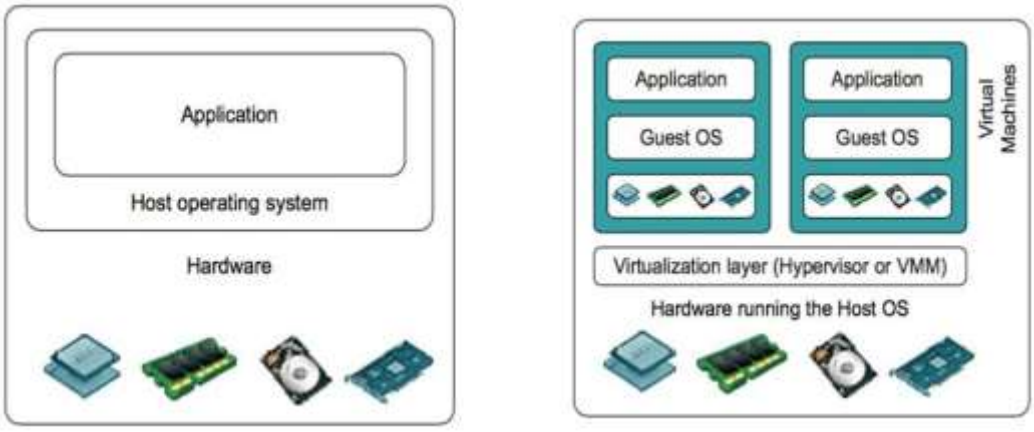
Based on the components that are virtualized and the functionalities offered by the virtualized components, the virtualization concept is categorized into multiple hierarchies. The topmost in this list is the machine.

1. Machine-level virtualization
    a. Processor virtualization
    b. Memory virtualization
2. Network virtualization
3. Storage virtualization
4. Desktop virtualization
5. Other types

Of the different types of virtualization, the most important is the machine-level virtualization, which is being discussed next. Other important types are network virtualization, which virtualizes the underlying network interface to LAN and storage virtualization that virtualizes the underlying storage components such as primary storage (hard disk) and secondary storage (such as SAN or NAS), will be discussed in later modules.

***4.1. Machine-Level Virtualization***

The first and foremost type of virtualization is machine-level virtualization also called the hardware virtualization. Generally, virtualization environment is often referred to as machine level virtualization. In this approach, the underlying hardware components of an operational system (Processor, Memory) is segmented, isolated and operating systems are deployed on top of the isolated segments to make independent operational environments. Each of the segments is called as virtual machine. We see the traditional computer in Figure 23.1(a) where an operating system that is specifically compiled for the underlying hardware architecture is present. Contrast this with a virtualized environment as shown Figure 23.1(b) where user applications are running on top of their own operating systems, also called the guest OSs, in a virtualized environment. A software layer called the virtual machine monitor or VMM or Hypervisor virtualizes the host machine into different virtual machines (VM) and these can run on the same hardware. The VMs are not dependent on the host OS.




**(a)** **(b)**

**Figure 23.1: Machine Level Virtualization**

However, this is one way of virtualizing. The virtualization layer can be implemented at various operational levels in a machine, thereby creating different abstraction layers.

Common virtualization layers include the instruction set architecture (ISA) level, hardware level, operating system level, library support level, and application level [5]. This is shown in Figure 23.2. In the next few sub-sections, we will describe these virtualizations.

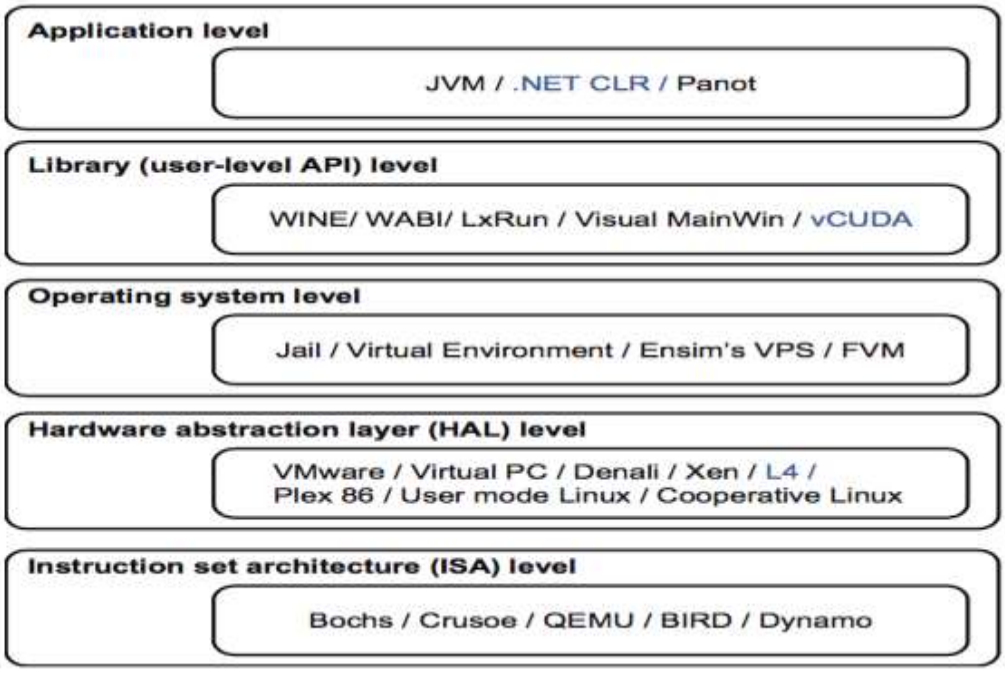



**Figure 23.2: Virtualization at Different Implementation Levels**

*4.1.1. Application Level Virtualization*

In application level virtualization, the end user application is virtualized and executed as its corresponding virtual machines. A generic application will be implemented with 3-tier architecture such as

1. Database layer running on a database engine or database server
2. Application layer running on Application server
3. Web layer running on Web server

In application level virtualization, all the three above-mentioned services are virtualized and made to work as individual virtual machines. Multiple virtualized web server, application server and database server will be running as independent units on a single application server, web server and database server. These parallel execution services forms application virtual machine which constitutes of virtualized web service, virtualized application service, virtualized database service. Each independent user is provided access to application VM where the functionalities of web server, application server and database server can be customized by corresponding users. This feature is called multi-tenancy, where same service is virtualized and functionalities are customized independently for multiple users.

This methodology is also called as process-level virtualization since the web server, application server and database server is virtualized into individual process and every process is executed in parallel independent of one another. Java Virtual Machine is an example.

This virtualization is applicable to the application and database implemented with web service compatibility.

### *4.1.2. Library-Level virtualization*

Certain Application Programming Interfaces (APIs) of the user-level libraries are commonly used by many applications and databases. These API are good candidates for virtualization. API hooks are used controlling the communication between applications and the rest of a system. WINE is a tool that has implemented this mechanism and can support Windows applications on Unix machines. vCUDA is another example.

### *4.1.3. OS Level Virtualization*

Operating system comprises of Kernel space and user space. In operating system level virtualization, an abstraction layer between the traditional OS and user applications is introduced. This layer virtualizes the operating system user space components thereby abstracting it from user applications. This individual virtualized space is called container. Multiple containers are executed in parallel on single operating system kernel with each container customized according to its users. The containers show the behavior of real servers. Jail is an example, which is a FreeBSD-based system. It helps partition a computer into mini-systems. These mini-systems are called jails.

### *4.1.4. HAL Level virtualization*

Hardware abstraction layer or HAL is a set of routines in that behave the same way as the underlying platform, such that programs can access the hardware directly.

Virtualization at the HAL-level generates a virtual hardware environment for a VM.

This virtualization is applied on the bare machine. A computer's resources such as processors, memory, and I/O devices are virtualized so that many users can share the same hardware. In the 60s, IBM's VM/370 implemented the idea first. Recently, Xen, VMware and a host of other hypervisors use this approach.

#### *4.1.5. ISA Level Virtualization*

As we already have discussed, ISA is the instruction Set Architecture that represents a hardware architecture. Virtualization is the core technology that offers cloud services. At the Instruction Set Architecture level, virtualization is performed by emulating a given ISA by the ISA of the host machine. Emulation process is the backbone for virtualization. In simple terms, emulation is the process, which enables one system to behave as another system. QEMU and Dynamo are examples of this category.

### *4.2. Network Virtualization*

Network virtualization is the process of virtualizing the overall network employed for establishing the cloud environment and its components, primarily the network interface card that is directly attached to the servers.

In an advanced network virtualization the hardware and software network resources along with its corresponding functionalities are virtualized, implemented using software and combined into a single, software-based administrative entity.

The network underwent virtualization called Virtual networks which may be Local area network or Wide area network. The advanced network virtualization functionality and its method of operation are termed as Software defined networks or SDN. We will discuss Network virtualization in a later module.

### *4.3. Storage Virtualization*

Storage virtualization is the process to virtualize the storage components. The storage system can be of various categories such as the local storage directly attached to the servers (hard disk) or secondary extended storage such as Storage Area Network. In the cloud perspective, the computation units and storage units are virtualized and controlled as independent entity. Storage virtualization process isolates the storage units according to the desired user capacity and creates the virtualized storage space as mountable volumes. These volumes are identified by certain identification numbers and are mounted to the virtual machines.

Storage virtualization decouples the physical organization of the storage from its logical representation. The users accessing their own data in cloud either directly or through virtual machines are not aware of the specific location of their data. A logical path is provided to their data volumes and the path will be mounted to their VMs or to their devices through which the users will get access to their storage volumes and to the data. Multiple storage facilities can be represented by a single standard file system.

The details of storage virtualization will be discussed in a later module.

### *4.4. Other Virtualizations*

There are other virtualization techniques depending on the level of implementation. We discuss about to such methods briefly in this section.

### *4.4.1. Desktop Virtualization*

Desktop Virtualization is the technique, which virtualizes the user space of Operating system whereas the kernel space of the operating is same. This method makes multiple desktops of operating system to be running simultaneously on the same system with single operating system. Each desktop instance is allocated to different users and the users in turn will access their corresponding desktop through remote desktop connectivity. The applications and files stored by corresponding users will be hidden from other users sharing the same hardware and OS.

### *4.4.2. I/O Virtualization*

I/O Virtualization is the process of managing the Input and output of the instructions to the virtualized environment.

Emulation is a key factor for implementing I/O Virtualization. This technique emulates the underlying devices to Guest OS for flow of input and output in both directions. The instruction/data/signals from the devices such as network, graphic devices, audio devices, display devices are managed by its corresponding device drivers. These instructions/data from device drivers are then passed on to Input/Output stack. The device emulator fetches the instruction from the stack and emulates the instruction according to the guest OS running in the guest operating system. The emulated instructions are passed on to the Guest drivers that reach the Guest OS. This procedure is followed in reverse direction also. The base drivers, I/O Stack and device emulators forms the virtualization stack and the Guest drivers, Operating system forms the guest VM.

## 5. Hardware Virtualization

While there are various ways to virtualize a computer, the most popular way of virtualizing a machine is by the way of deploying a virtualization layer. Now we are ready to explore this virtualization in greater detail.

The idea of virtualization originated in 1960 and with gradual improvements virtualization today has become an integral part of Cloud computing. In 1972, IBM introduced virtual machine operating system. The current version is called a Virtual Machine Monitor (VMM) or Hypervisor. VMM runs on the physical machine.

The typical layering of hardware, OS and application is modified in a virtualized environment. The modified virtualized environment can be either a three or a four layers environment depending on the type of hypervisor being used.

Virtual Machine Monitor (VMM) or Hypervisor is the piece of software that provides the abstraction of a virtual machine. This is responsible for the entire virtualization process. In the current text, we will use the two terms VMM and hypervisor interchangeably.

The operating system running on top of the Hypervisor in virtual machine catering to the needs of the application is the guest OS and the OS that sits on top of the hardware is called the host OS.

### *5.2. Virtual Machine Monitor*

VMM is the software that plays a major role and is responsible for the entire virtualization process. Virtualization software or Hypervisor is categorized into two different types:

1. Type 1 hypervisor
2. Type 2 hypervisor

#### *5.2.1. Type 1 Hypervisor*

Type 1 hypervisor has three layers. The lowermost layer is the hardware layer and networking layer comprises CPU, storage and network components. On top of this is the operating system layer. The next layer is the virtualization layer where the hypervisor resides and virtualizes the underlying resources. On top of the virtualization layer reside the guest environments, which are the virtual machines. The architecture is depicted in Figure 23.3.


**Figure 23.3: Type 1 Virtualization Architecture**

The hypervisor is installed on bare metal, *i.e*, directly on a hardware environment in contrast to the four-layer architecture and hence this hypervisor is also called as bare metal hypervisor. The guest environment contains the guest OSs that are installed on top of the hypervisor. Guest OSs run in less privileged mode. The privilege level of guest OS is emulated by the Virtual Machine Monitor or Hypervisor. We will explain the workings in greater details in the next module.

#### *5.2.2. Type 2 Hypervisor*

Type 1 hypervisor is difficult to use and implement, hence the other type called Type 2 hypervisor is employed. In type 2 hypervisor there are four layers, as depicted in the figure 23.4. Here also the lowermost layer is the hardware layer. On top of this is the operating system layer. Next layer is the virtualization layer with the guest environment on top. Here the hypervisor is installed on top of a host operating system. This is also called hosted environment.

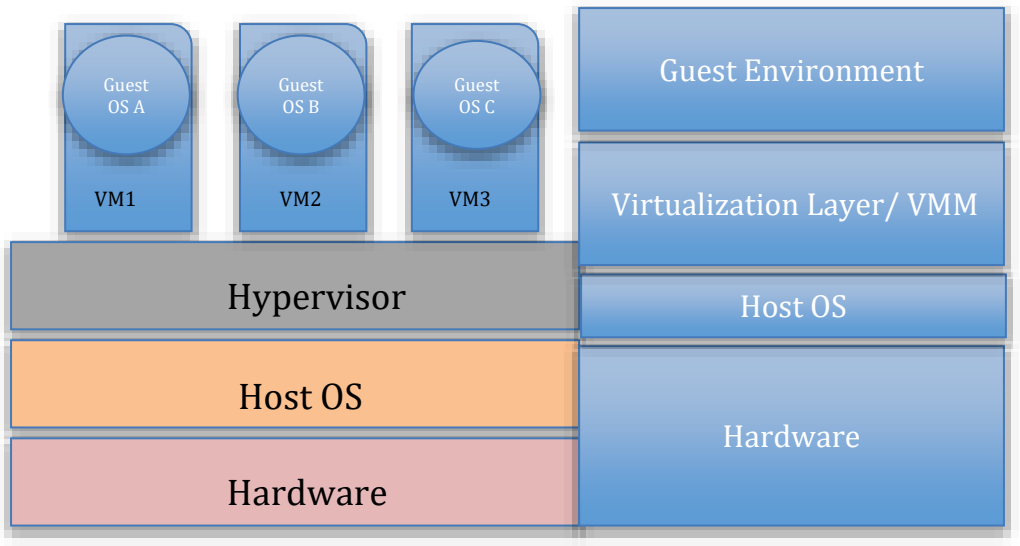



**Figure 23.4: Type 2 Virtualization Architecture**

The Hypervisor utilizes the functionalities available on host OS to control and manage resources desired by each of the virtual machines. These can support broadest range of underlying host hardware configurations.

*5.2.3. Hybrid Hypervisor*

While the hosted/ Type 2 hypervisor is easier than bare metal/Type 1 hypervisor, the efficiency of virtualization in Type 2 is not available in Type 1 since the hypervisor works like an application. Hence, it is desirable that some part of the VMM must be in direct control of the hardware. This is achieved by creating a hybrid model hypervisor as shown in Figure 23.5.

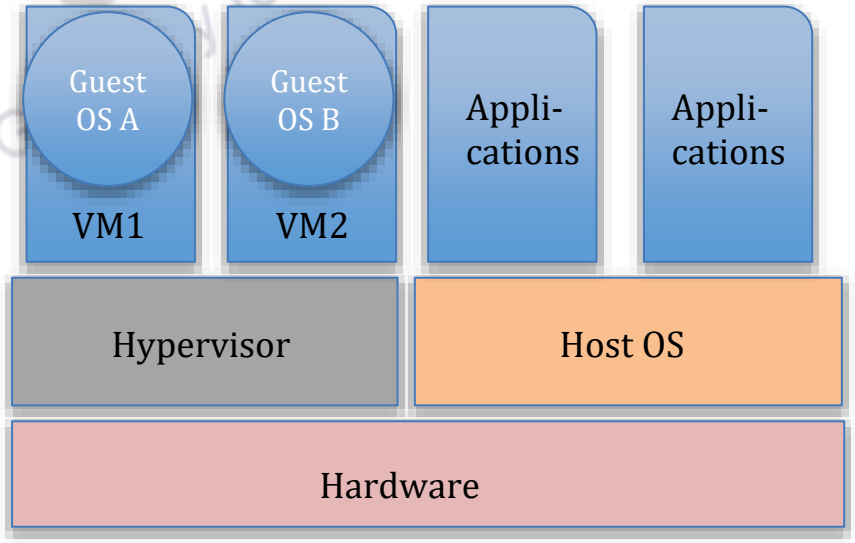



**Figure 23.5: Hybrid Hypervisor**

The VMM here shares the hardware with a host operating system. This is done through mechanisms commonly provided to extend the functionality of an operating system such as kernel extensions and device drivers. Applications that run on top of

the VMM are run in the VM environment. The hybrid system supports multiple virtual machines. On the other hand, there are applications that can be run on the normal system on top of the operating system itself. This system is referred to as a *dual-mode* hosted VM system.

### *5.3. Virtual Machines (VMs)*

The virtual machine (VM) is the core component formed as a result of virtualization. The basic cloud services such as IaaS (Infrastructure-as-a-Service) and PaaS (platform-as-a-Service) are delivered to the end users as virtual machines.

Like the rest of the system, the guest environment in the virtualized system, or the virtual machines, also follow a layered approach, where the bottom layer within the virtual machines is the virtual hardware and layer on top of the virtual hardware is the operating system layer. On top of the operating system layer resides the user applications, as shown in Figure 23.6.

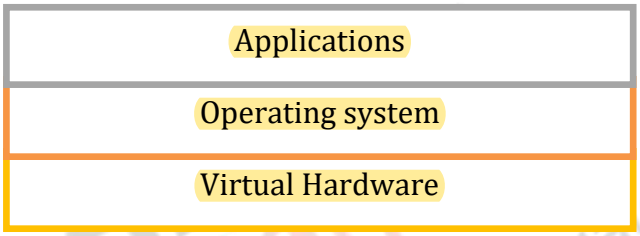



**Figure 23.6: Layers of a Virtual Machine**

#### *5.3.1. VM Taxonomy*

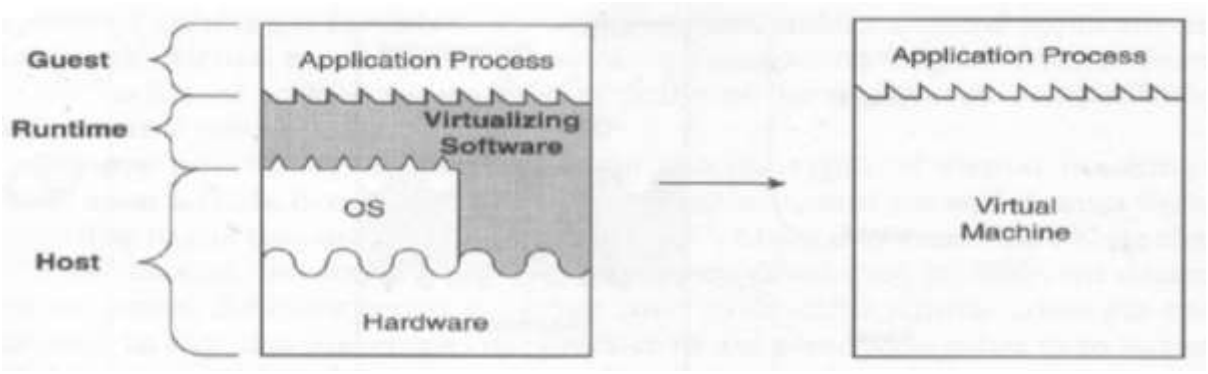
From the perspective of the scope, we can divide the virtual machines into two major categories:

1. Process VM
2. System VM

##### *5.3.1.1. Process VM*

These VMs are platforms created by operating system specifically for the process. These VMs are created when the application is initiated and they are destroyed when the application finishes execution. These VM support binaries compiled on different instruction set. For example, Java Virtual Machine.

Figure 23.4 shows a process VM.



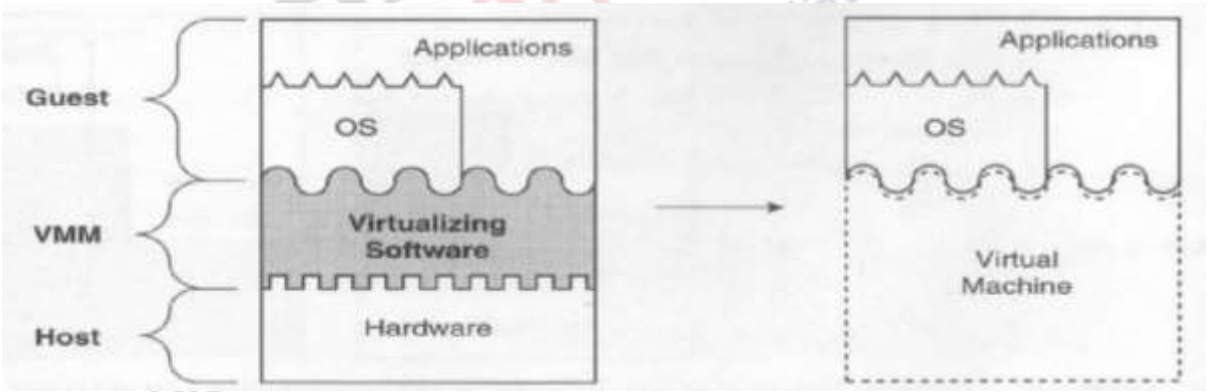

Courtesy: [1]

**Figure 23.7: Process VM**

The virtualizing software in the process VM environment translates one instruction from one platform to another platform. With the help of this, programs developed for variety of operating system or with different instruction set architecture can be executed. These VMs terminates automatically once the process terminates.

*5.3.1.2. System VM*

System VM is the VM that is generally created for virtualizing the underlying hardware and networking resources. Figure 23.5 depicts the system VM.




Courtesy: [1]

**Figure 23.8: System VM**

These VMs provide a complete operational environment, comprising guest operating system, user process, networking components, input output environment, graphical display components and so on. These VMs are tied to the system and not any specific process and will be running as long as the host hardware is running or user terminates the VMs.

These VMs have their own guest operating systems, which are made bootable from OS template called images. These support multiple images simultaneously. Each image runs its own OS and is associated with specific application programs at any point in time. Each guest OS controls and manages its own virtualized hardware resources. The hardware environment is shared among the VMs running simultaneously. Virtual Machine Monitor manages the allocation of, and access to, the hardware resources of the host platform.

*5.3.2. Operations on VM*

There are four basic operations that can be defined as the primitive operations of a VM as follows:

VM Multiplexing
VM Suspension
VM Provision
VM Migration

*5.3.2.1. VM multiplexing*

The word multiplexing here indicates the ability of using multiple instances. In a non-virtualized environment, while multiple applications may be run on top of a given hardware/machine, there are severe restrictions in the number and types of the applications. In a virtualized environment, these applications are deployed in VMs and hence restrictions regarding the number of VMs that can be launched, the type of application that can be run in the VMs, the amount or share of the hardware that can be consumed by these applications become irrelevant. Hence this is called VM multiplexing.

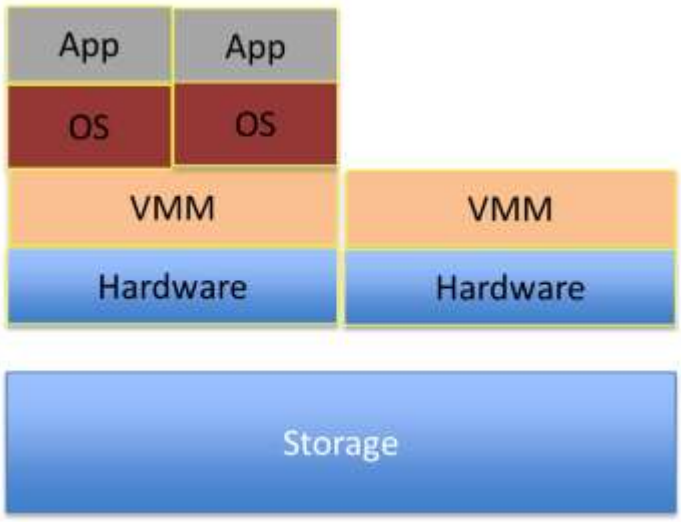



**Figure 23.9: VM Multiplexing**

*5.3.2.2. VM suspension*

VM Suspension is the process of moving the virtual machines to a paused state from running state. Any running or waiting VMs may be suspended from the current state and moved to storage, as shown in Figure 23.9. The VMs reside in storage until revoked back to execution.

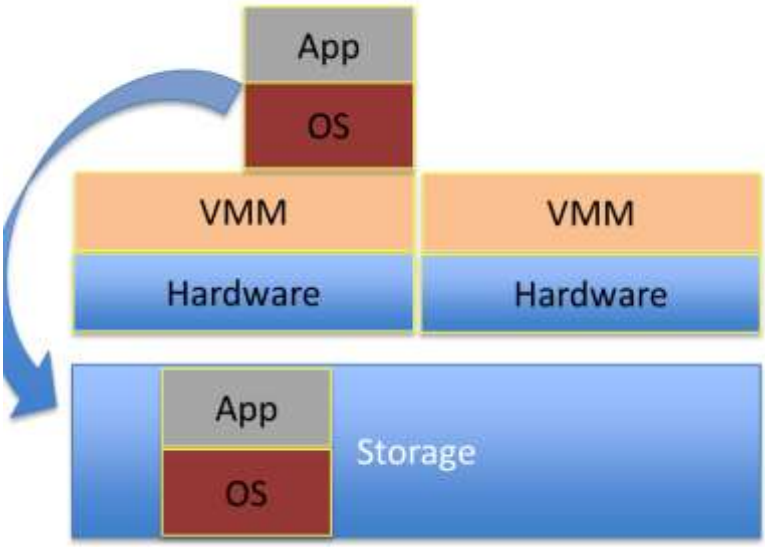



**Figure 23.9: VM Suspension**

*5.3.2.3. VM provisioning*

When needed, a suspended VM can be brought back to the execution environment and be scheduled on the same hardware or on a different hardware. This is shown in Figure 23.10.

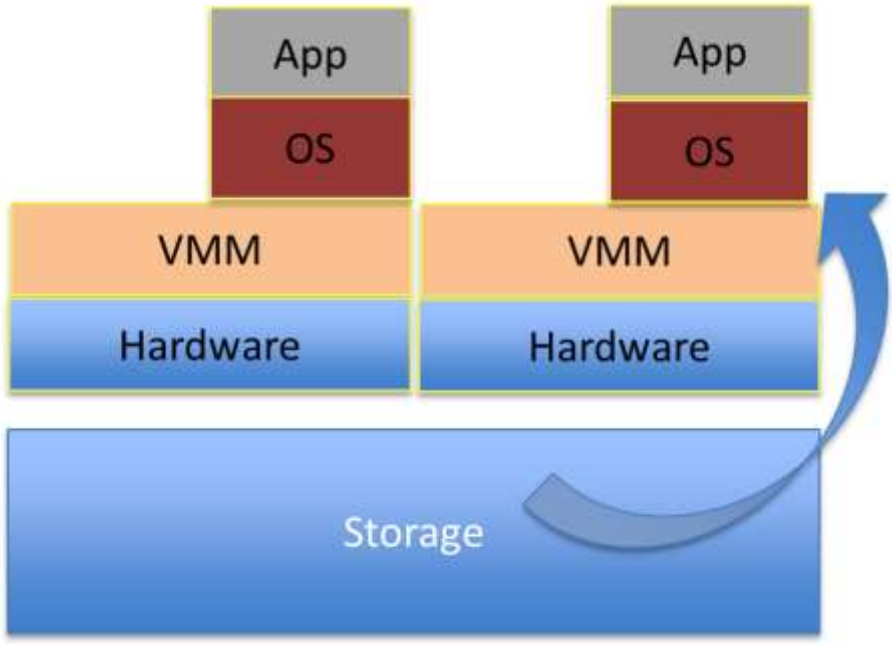



**Figure 23.10: VM Provisioning**

*5.3.2.4. VM migration*

The VMs can be migrated directly from one server to another server either as live migration without shutting down the virtual machines or as cold migration by shutting down the VMs and migrating the VM and rebooting the VM at destination. Figure 23.11 depicts this mechanism.

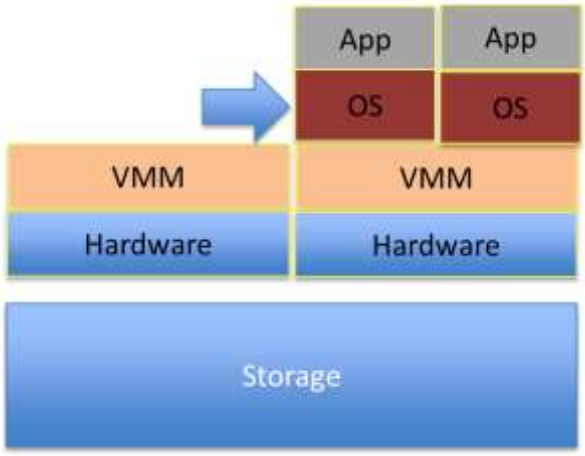



**Figure 23.11: VM Migration**

*5.3.3. Benefits of Virtual Machines*

We can intuitively understand the benefits of using VMs and hence using virtualization. Let us point out some of these benefits in this section.

1. Virtual machine is a way of consolidation of servers. Various hardware can be brought together and these hardware can be better utilized by deploying and executing multiple VMs. Thanks to the various operations that can be performed on the VMs, the different hardware can be utilized to their fullest.
2. The point above directly makes optimal utilization of server and storage resources possible, which otherwise would not be possible.
3. With guest OSs running inside the VMs, a VM environment looks the same as a normal OS environment provided to an application running in a non-virtualized machine.
4. Perhaps one of the most important advantages of virtualization is the ability of the VMs to run applications in different OS environment using the same hardware.
5. Higher levels of security are possible since the VMs run applications in isolated environment.
6. High level of availability is possible since, in case of the failure of a certain server, migration can be applied to seamlessly move the VMs from the failed machine to another machine.
7. Since servers can be added and removed without affecting the running VMs, a virtualized environment supports scalability and portability.
8. It is easy to backup all the data, which promotes faster recovery as a result.

## 6. Summary

In this module, we first explored various types of virtualization. Among these, the hardware or machine level virtualization is further categorized based on the implementation. Since this is the most important and common method of virtualization, we further investigated on this part and looked at the practical implementation of this type of virtualization. We discussed the virtual machines in detail along with the types and benefits thereof.

We continue our discussion on virtualization in the next two modules as well.

## References


1. Smith, J. E., and Ravi Nair, "Virtual Machines: Architectures, Implementations and Applications", Morgan Kauffmann, 2004.
2. Figueiredo, Renato, and Peter A. Dinda, "Guest Editors' Introduction: Resource Virtualization Renaissance", Computer 5: 28-31, 2005.
3. Hwang, Kai, Jack Dongarra, and Geoffrey C. Fox, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things”, Morgan Kaufmann, 2013.
4. Buyya, Rajkumar, Christian Vecchiola, and S. ThamaraiSelvi, “Mastering Cloud Computing: Foundations and Applications Programming”, Newnes, 2013.
5. Chiueh, Susanta Nanda Tzi-cker, and Stony Brook, "A Survey On Virtualization Technologies”, RPE Report, 2005.

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 24: Virtualization III
# Module No: CS/CC/24
# Quadrant 1— e-text

## 1. Introduction

Virtualization holds the magic called cloud. We learnt the basic method of virtualization and the different types of virtualization in the last two modules. In this module we continue the discussion by looking deeper into what is virtualization and its requirements, properties followed.

In this module we also look at the constraints of a virtualization method in the form of Popek Goldberg theorem which governs the virtualization process. We provide how the world of cloud have come up with elegant solutions to the problems.

## 2. Learning Outcome

At the end of this module, students will be able to:

1. Understand the concepts of hardware virtualization.
2. Obtain an in-depth knowledge about the difference between full emulation and virtualization.
3. Learn the details of what is required in a hardware to make it virtualizable.
4. Learn Popek-Goldberg theorem and understand the characteristics of virtualizability.
5. Study the details and the problems of x86 architecture with respect to virtualization.

## 3. Virtualization

As we have already mentioned, CPU virtualization or hardware virtualization is key virtualization process among all other virtualization processes such as memory virtualization, storage virtualization, network virtualization, desktop virtualization. CPU virtualization offers one or more of the following features:

- Emulate underlying physical environment in the virtual software environment through the hypervisor.
- Provide an illusion to all the software processes, whether system process or user process, that these processes are running on top of the hardware and instructions are executing on top of hardware. However, in reality, the process and instruction are not executed directly on hardware. They are executed on emulated hardware environment.
- This makes the system environment capable of running multiple operating systems of different variety on top of the hypervisor simultaneously.

Often we ask the question whether difference between virtualization and emulation? Let us try and answer that question here.

**4. Virtualization and Emulation**

Emulation is the common phenomenon and is an oft-used method. It also forms the base for virtualization process and represents underlying principle behind virtual machine operation. However, the two techniques are not the same.

Emulation, or more precisely full emulation, typically refers to the process of presenting the behavior of underlying hardware through software. In other words, emulation is a process by which, programs written for one computer can be executed on another computer.

One of the most commonly used emulation is at the ISA level. As shown in Figure 24.1, the host hardware could be of any type and tied to a specific ISA, but by creating an emulation layer other ISAs can be executed on top of the hardware. Emulator is a mechanism that is able to replicate the functionality of the underlying processor and corresponding hardware systems completely.


**Figure 24.1: ISA Emulation**

Virtualization technology, on the other hand, is one that divides and combines the underlying resources of server to execute one or more operating systems simultaneously and independently using the combination of various methodologies such as hardware and software partition/aggregation, partial/complete machine simulation, emulation, time sharing and many others. Typically, in virtualization, the ISA of the applications is the same as that of the underlying hardware. The virtualized environment generally consists of physical hardware acting as host either with base operating system or without base operating system. Hypervisor, which is the virtualization layer, is deployed either on top of the hardware layer or on top of the host OS. In either case, the virtual machines are deployed on top of the hypervisor, i.e., the virtualization layer. Figure 24.2 shows the different layers with the virtualization layer sitting directly on top of the host physical layer.

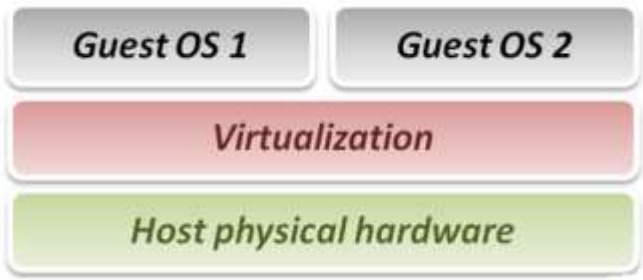


**Figure 24.2: Virtualization Technique**

To implement a hypervisor, emulation is used as one of the methods. The emulation process is implemented by a special software called emulator. If used in virtualization, emulator is integrated with the hypervisor layer. Virtualization typically involves only one ISA. Although emulator is the basic technique for virtualization, virtualization is not confined to emulation. In this module we discuss various methods of virtualization including emulation. Virtualization has specific requirements failing which it is not possible to virtualize a system. Now let us look at the requirements.

### *4.1. Requirements*

The first and foremost requirement of virtualization is that the guest must be able to exhibit the near-native behavior as that of real hardware environment and almost equivalent performance of real environment when executed on a virtual environment. The ISA of the hardware on which the virtualization software is deployed and the ISA of the multiple heterogeneous operating systems run in the VMs are typically the same. In this environment, what is the requirement for performance? What are the problems that affect the performance of such machines? Let us understand this by delving a little more into the architecture of the underlying hardware.

### *4.2. Third Generation Computers*

As we have seen in an earlier module, although we are in later generation of computers, today's computers are essentially the same as the third generation computers. These computers are integrated with circuits, which were introduced during 1964. These circuits interface with applications through specialized software called operating system. In this environment, multiple user processes are executed on a time-sharing basis. However, these concurrent applications impose certain restrictions on the processes that are run in the CPU. In addition to the user processes, certain OS processes also have to use the CPU. The OS functions are critical processes that are empowered with special abilities and are to be allowed to access certain portions of the primary memory, that are out of bounds for user processes. However, user processes can neither be allowed to do these critical operations, nor are these allowed to access such areas of the primary memory. How does the system ensure that there are no violations? This is why the CPU runs in two modes: privileged mode and user mode. The system critical processes run in the privileged mode or the supervisor mode.

When a user executes in a computing system, the processor runs in the user mode and only a generic set of instructions are executable. When an operating system process executes, the processor switches to privileged mode when all the instructions are executed without any restriction. Only an OS process can execute special instructions.

#### *4.2.1. Trap in Conventional and Virtualized Environment*

In any computing environment, when a user process needs to execute an instruction that can be done only in privileged mode, it seeks the help of the OS. This causes the CPU to change mode and bring in the relevant OS process that is capable of executing the instruction on the behalf of the user process. After execution of the user mode instruction as special instruction, the results are passed on to the

respective calling user process. This method of calling a corresponding OS process by a user process is called *trapping to OS*.

Therefore, an instruction in user mode is trapped when it tries to execute any special instruction. The trapping may occur for an I/O operation or for any other special action. The trapping to OS is depicted in Figure 24.3.

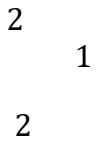



**Figure 24.3: Traditional Environment**

An application or user process sends its instructions on the underlying hardware to be executed by CPU in user mode. This is shown by path no 1 in Figure 24.3. When this application process needs to execute a special instruction, it traps to the OS and a corresponding special routine OS process gets executed by the CPU in privileged mode as shown by path 2 in Figure 24.3.

A question that arises is why we are discussing the third generation computers here? This is because there has been hardly any change in today's computer architecture when compared to that of the third generation. For example, the hardware still essentially consists of a processor and memory. The processor can still operate in either the privileged mode or the user mode. Even in today's machine, a subset of the instruction set can be executed only in the privileged mode. Memory addressing is still done relative to the contents of a relocation register. Hence, we see that whatever was relevant in the third generation computers are still relevant in today's computers.

Now let us see, if, in this machine architecture, when we try to install a hypervisor, how does things change? What happens to the two modes of CPU execution when a hypervisor sits instead of the host OS with a guest OS sitting on top? This is shown in Figure 24.4. Can the execution of hypervisor instructions in the virtualized environment be treated in a similar manner as that of the special instruction with trap to OS as in the original environment?

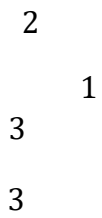



**Figure 24.4: Trap in Virtualization Environment**

Let us consider the new environment with a virtualization layer or VMM as we have considered in Figure 24.2 earlier. This is a three-layer system with a VMM or virtualization layer on top of the hardware. Applications here, as in any virtualized environment, run on top of the guest OSs. This is shown in Figure 24.4. Here also the normal executable instructions of the application continue to execute in the CPU directly, via path no. 1. However, when the application is trying to execute special instructions, it will trap to the guest OS via line 2, since the guest OS is the OS which is visible to the application. However, since the guest OS is not running directly on top of hardware but on hypervisor, it, in turn, must behave like an application and instead of executing the special instruction of the machine, after checking the parameters, the guest operating system must now issue a similar special instruction execution request to the hypervisor. The hypervisor, in turn, executes the special instruction in the privileged mode.

One of the key functionalities of a hypervisor is that it should act like a normal operating system along with additional capabilities. For example, a VMM must act as a scheduler with capability to run multiple operating systems on top of it. Can we implement a VMM on all types of hardware? Let us now explore this by looking at the requirements for a machine to allow virtualization. This is called virtualizability.

### 5. Virtualizability

Virtualizability is the ability of a hardware to allow an operating system to run on top of virtual machine monitor. Popek-Goldberg first proposed the concept and the requirements of virtualization in their path breaking paper in 1974 [1], [2]. There are three properties that are relevant with respect to a VMM:

1. Equivalence / Fidelity
2. Resource control / Safety
3. Efficiency / Performance

Let us understand these properties.

#### Equivalence / Fidelity

This property states that a program running on top of a guest OS under the control of VMM should exhibit a behaviour that is similar to that in a non-virtualized environment.

#### Resource control/Safety

This property states that the virtual machine monitor (VMM) takes complete control of the entire virtualized environment which includes all virtual resources.

**Efficiency/Performance**

This property states that of all the machine instructions that a process would execute, a large part must be executed on its own by the process directly accessing the hardware without VMM help/ intervention. This is to ensure the performance.

### *5.1. Popek-Goldberg Theorem*

Researchers Popek and Goldberg formally derived the conditions sufficient for an ISA to efficiently support virtual machines. In other words, virtualization process that can be supported by an ISA is discussed in the theorem proposed by Popek and Goldberg. Although they had talked about machines such as IBM System/370 and Honeywell 6000, which are the third generation machines, the same analysis is still relevant for today's machines as we have seen earlier in this module that the basic architecture of today's machines are very similar to that of the third generation machines.

The theorem:

"*For any conventional third generation computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of privileged instructions*".

The theorem states that for all such computers for whom sensitive instructions is a subset of the privileged instructions, virtual machine monitor can be deployed. We have already seen the priviledged instructions. So what is a sensitive instruction? Let us understand the instruction behaviour of computer systems.

#### *5.1.1. Instruction Behaviour*

There are three ypes of instructions:

1) Privileged instructions: These are instructions that trap in user mode but do not trap in privilege or supervisor mode of execution.

2) Sensitive instructions: There may be a set of instructions in a computer that behave differently depending on the mode of the processor. That is, if it is executed in the user mode, it takes an action, which is different from the action taken by the same instruction when executed in privileged mode.

3) The rest of the instructions are non-privileged instructions.

Now, the first two sets, viz, the privileged and the sensitive instructions, have no fixed relation. It varies from computer to computer. According to the Popek-Goldberg theorem, when the set of sensitive instructions of a computer is a subset of the set of privileged instructions, it is possible to virtualize it by constructing a VMM. The Popek Goldberg theorem is explained in the Venn diagram given in Figure 24.5.

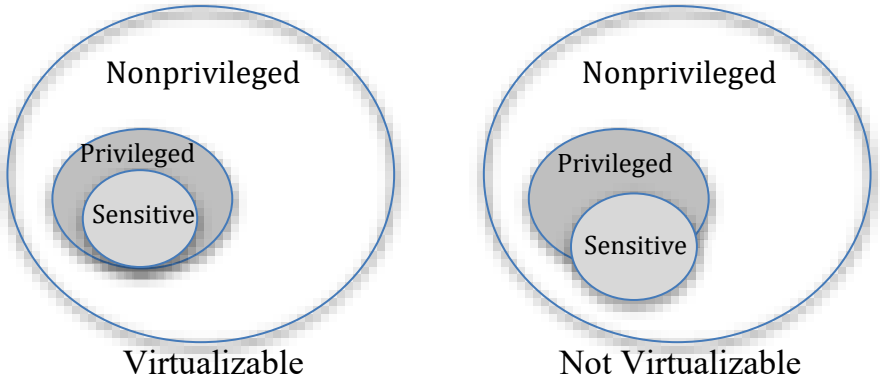



**Figure 24.5: Popek-Goldberg Theorem**

As seen in the above figure, in the 'virtualizable' machine, sensitive is a subset of the privileged. However, in the other part, although a part of the sensitive instructions is indeed a subset of the privileged instructions, there are some sensitive instructions that are not privileged instructions.

So, what problem does a machine face? In simple terms, a computer would not behave in accordance to the three properties of virtualization if the corresponding architecture is not virtualizable. This revelation would perhaps not have mattered under a different circumstances, but it is important to the manufacturers as well as users and researchers, since x86 falls in the 'not virtualizable' category and this is the most used architecture in our computing history. Let us now look at the x86 architecture.

### 6. x86 Architecture

The x86 architecture is an instruction set architecture (ISA) series for computer processors. This is developed by Intel Corporation. x86 is the single most used ISA by major players. It has a ring architecture, as shown in Figure 24.6.

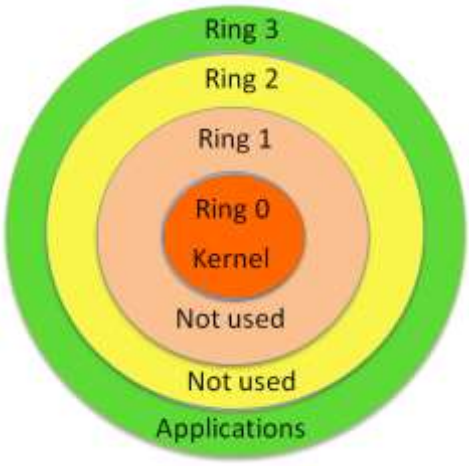



**Figure 24.6: x86 Architecture**

As can be seen in the above diagram, there are four rings that define the architecture. The structure of x86 architecture is as follows:

Ring 0: forms the core of the architecture and lies in the centre, comprising of operating system kernel.
Ring 1: Surrounds ring 0 and is generically not used.
Ring 2: Surrounds ring 1 and is generically not used.
Ring 3: Forms the outermost layer of the architecture and surrounds ring 2. User applications are run in this layer.

### *6.1. Problems in x86 Architecture*

Since x86 is the most used machine popularized by Intel, let us look at the problems that will be faced when a computer with x86 architecture is to be virtualized. There are two issues that have to be resolved. The first is with respect to its instruction set behavior and the second is with respect to its ring formation.

Problem 1:

X86 architecture is not virtualizable according to the Popek-Goldberg theorem, since in x86, there are some sensitive instructions that are not part of privilege instructions. These sensitive instructions do not trap when executed in the user mode, but they change certain values and even fail silently in user mode. Hence x86 is not virtualizable as per Popek Goldberg theorem.

Problem 2:

The second problem is related to the ring formation. OS runs in Ring 0 in supervisor or privilege mode, while applications run in Ring 3 in user mode. If we need to run VMM, which ring should it run in? Since VMM now runs in privilege mode, it must be run in ring 0 obviously, but the existing OS necessarily need to run in ring 0. With two contenders for one ring, and the constraint that only one process can be accommodated in a ring, there has to be a special solution that must be devised for x86 virtualization.

### *6.2. The Difficulty*

The specific problem here is that, in a dual-mode CPU, there are three different entities that need to be accommodated, viz, the application, the guest OS and the VMM. How can we resolve this problem?

In general, user applications is run in the user mode and VMM is run in the kernel mode. While it is not safe to let the guest OS run in the kernel mode and it is not possible to let the kernel of the guest OS in the users mode, virtual machines are run in two modes:

1. Virtual kernel mode
2. Virtual user mode.

Usually, VMMs implement virtual CPU (vCPU) to represent state of CPU in host machine per guest machine. This process is called depriviledging the OS.

#### *6.2.1. Deprivileging the OS*

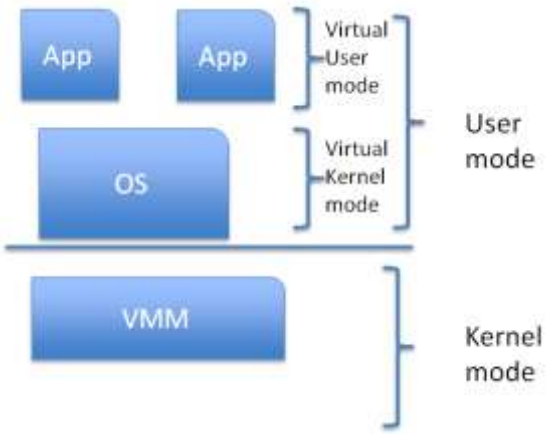



**Figure 24.7:Deprivileging the OS**

Deprivileging the OS is explained in the above mentioned figure where the virtual machine monitor is placed in the kernel mode and the entire guest operating system of virtual machine is placed in the user mode. The guest OS of VM will have its own kernel mode and user mode placed in the user mode of the base OS of physical machine. Within the VM, virtual kernel mode and virtual user mode is created with respect to the guest OS. The guest OS kernel runs in virtual kernel mode and the applications in the VM run in virtual user mode.
How exactly this deprivileging works will depend on the type of virtualization being used and we will learn about this in the next module.

**7. Methods of Virtualization**

After understanding various types of virtualization and pointing to the real problem of the most popular ISA, x86, let us now turn our attention to some of the generic methods using which virtualization can be achieved. Definitely, in a virtualized environment, the privileged state of a virtual machine differs from that of the underlying VMM. The VMM's basic function is to provide an execution environment that meets the guest's expectations in spite of this difference. There are three general methods of virtualization. These are:

1. Emulation
2. Trap and Emulate
3. Binary Translation

In the next module we will elaborate these generic methods.

**8. Summary**

In this module, CPU virtualization along with its features and functionalities is explained followed by an overview of virtualization and emulation. The requirements of virtualization are explained. The structure of third generation computers with respect to virtualization is explained along with what is meant by a trap to OS. The process of Virtualizablity and properties is explained. The Popek Goldberg theorem governing virtualization is explained followed by the instruction behavior such as priviledged and sensitive instructions. The virtalization procedure of x86 with respect to Popek-Goldberg theorem and the problems in virtualizing x86 are also explained. Finally the process of deprivileging OS for virtualization is explained in this module.
In the next module we will elaborate the generic methods introduced in this module. Further, we will also look at some of the practical mechanisms that are being used for the x86 virtualization today by the industry.

## References


1. Popek, Gerald J., and Robert P. Goldberg, "Formal Requirements For Virtualizable Third Generation Architectures", Communications of the ACM, 17.7: 412-421, 1974.
2. Smith, J. E., and Ravi Nair, "Virtual Machines: Architectures, Implementations and Applications", Morgan Kauffmann, 2004.
3. Figueiredo, Renato, and Peter A. Dinda, "Guest Editors' Introduction: Resource Virtualization Renaissance." Computer 5: 28-31, 2005.
4. Hwang, Kai, Jack Dongarra, and Geoffrey C. Fox, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things”, Morgan Kaufmann, 2013.
5. Buyya, Rajkumar, Christian Vecchiola, and S. ThamaraiSelvi, “Mastering Cloud Computing: Foundations and Applications Programming”, Newnes, 2013.

# e-PG Pathshala
# Subject: Computer Science
# Paper: Cloud Computing
# Module 25: Virtualization IV
# Module No: CS/CC/25
# Quadrant 1— e-text

**1. Introduction**

In this module we complete our discussion on virtualization. As we have seen already virtualization is a necessary mechanism in a data center providing cloud-based services. In the last module we've explained the fundamentals of virtualization, the objective of this module is to provide an overview on methods of virtualization, especially the procedure to implement CPU virtualization. We conclude this module, as well as our discussion on virtualization with an overview of a free and open source tool for virtualization named is Xen, and its architecture as case study.

**2. Learning Outcome**

At the end of this module, students will be able to:

1. Understand the general methods of implementing virtualization.
2. Understand how emulation is used for virtualization.
3. Get knowledge about the method of binary translation.
4. Learn the importance of paravirtualization.
5. Get knowledge about the hardware-assisted virtualization.
6. Learn the basic architecture of Xen, an open-source hypervisor.

**3. Methods of Virtualization**

As mentioned in the earlier module, there are three generic techniques that can be used for virtualization. These are:

1. Emulation
2. Trap and Emulate.
3. Binary Translation

Before explaining the methods in practice for virtualization, it is important to understand these techniques.

### *3.1. Emulation*

Emulation is the process where the virtualizing software mimics that portion of hardware, which is provided to the guest operating system in the virtual machine. The presented emulated hardware is independent of the underlying physical hardware. Emulation provides VM portability and wide range of hardware compatibility, which means the possibility of executing any virtual machine on any hardware, as the guest operating system interacts only with the emulated hardware.

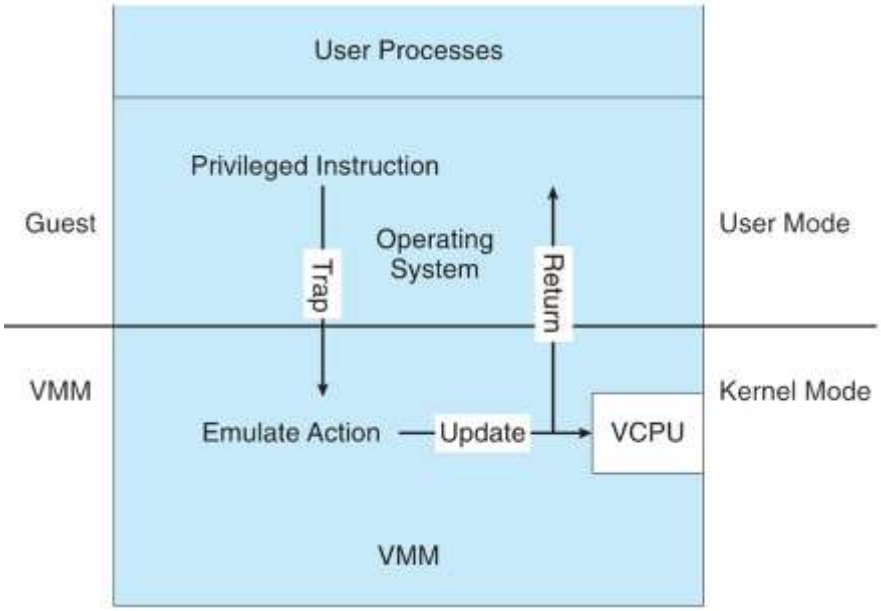
In an emulated environment, both the application and guest operating system in virtual machines run in the user mode of base operating system. In simple terms, the behavior of the hardware is produced by a software program. Emulation process involves only those hardware components so that user or virtual machines does not understand the underlying environment. Only CPU & memory are sufficient for basic level of emulation.

Typically, emulation is implemented using interpretation. The emulator component takes each and every instruction of user mode and translates to equivalent instruction suitable according to the underlying hardware. This process is also termed as interpretation. This means that the guest OS remains completely unaware of the virtualization. Also, in interpretation, each and every instruction issued by a VM is trapped in the VMM and interpreted for execution in the hardware. Goes without saying that computationally it is a very expensive method. However, in some cases,, it is needed to use an interpretation technique. However, due to the huge disadvantage of performance, emulation using interpretation is hardly used in virtualization.

### *3.2. Trap-and-Emulate*

Trap and emulate is a technique that takes the basic of the emulation but improves performance by using interpretation selectively. In this method also, both the user applications and guest operating system of virtual machines run in the user mode and the hypervisor runs in the privileged mode. However, here the application runs the instructions natively on the hardware. Only when a privileged instruction is to be executed in virtual user mode, a trap to the virtual kernel mode occurs. This causes a trap to the VMM in turn. The hypervisor gains control of the execution. All the special instructions that need to be managed, are managed by the VMM and not by the guest OS.

When hypervisor traps, it executes the necessary equivalent operations in the underlying instruction set architecture and returns control to guest in user mode. User mode code in guest runs at normal speed. There is no change from running in non-virtualized environment. But kernel mode privileged codes run slower due to trap-and-emulate. The mechanism works fine, except that the CPU is slower in kernel mode. Unfortunately, it becomes a performance issue when there are several guests and each guest needs to trap to VMM for all privileged mode instructions.




Courtesy: [4]

**Figure 25.1: Trap and Emulate**

The privileged instruction from the guest operating system is trapped by the hypervisor and emulated. This emulated instruction is then passed on to the kernel. On return the privileged instruction from the kernel is emulated and returned to the user mode where the operating system of virtual machine is running.

There are issues prevailing in Trap and Emulate methodology. Not all ISAs can be emulated using trap-and-emulate method. Those that do not follow Popek-Goldberg theorem cannot be emulated. Sensitive instructions are not Privileged instructions. Let us consider the example of Intel x86 popf instruction. The CPU flags are loaded from the contents of stack. In privileged mode, all flags are replaced whereas in user mode only some flags are replaced, hence no trap is generated.

The following is an example to understand why trap and emulate does not work.

Let us take the example of the popf instruction. This instruction loads a set of flags from the stack into a register and can be executed in both the user mode and privileged mode. When executed in privileged mode, popf loads all flags, which include a mix of ALU flags and system flags. In this mode, both the set of flags get modified, as per requirement, since the OS has the permission to make necessary changes in the system flags. However, when the same popf gets executed in user mode, the processor simply does not allow modification of the system flags even if the situation demands it. But, no error or exception is generated due to this. This is what is meant by ‘going totally unnoticed’ by the underlying OS.

This works fine in a non-virtualized environment. Now, let us see why this would cause a problem in a virtualized environment using a trap and emulate kind of mechanism. The guest user mode runs the popf instruction without altering the system flags and this is just a normal action. What happens when the guest OS executes the popf instruction? It is ‘expected’ to modify the system flags as per requirement. But since this itself is running in the user mode, it will not be allowed to modify the system flags! However, a question here is that the same is applicable for all other privileged instructions in this scenario yet there were no problems with them. The explanation to this question is as follows: Since, in all other privileged

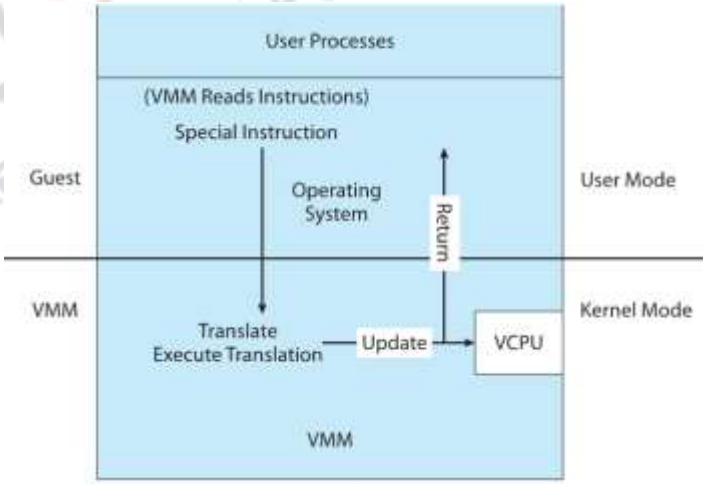
instructions when the guest OS attempts to execute a privileged instruction, the processor will detect this 'unnatural' behaviour and will wake up the VMM. However, since processor is supposed to allow the execution of popf in user mode as well, it will not raise the VMM when the guest OS attempts to execute the instruction. It just does not allow the guest OS to modify the system flags since it is expecting the guest OS to 'behave' like an application and not as an OS. As a result, the outcome of this execution is not the same as the expected outcome of this instruction.

### *3.3. Binary Translation*

The issue of popf and other such instructions that occur in the trap and emulate method can be taken care of if the VMM gets more proactive. This additional activity of VMM is the Binary Translation. Although the concept of binary translation is simple, the complexity lies in implementation of binary translation.

If guest vCPU is in user mode, guest can run instructions natively, whereas if guest vCPU is in kernel mode, VMM checks every instruction (and does not wait for a trap!). Non-sensitive instructions run normally but sensitive instructions are translated appropriately. Performance of this method is worse than trap-and-emulate since all codes of the guest kernel is inspected by the VMM. However, many optimizations have been proposed. Let us look at such an optimization.

One such available optimization for improving the performance is caching. In caching technique, a block of instructions are translated once and stored in cache. When guest executes the code with sensitive instructions, the cached translation is first checked. If instruction is available in cache as translated instruction, then no more translation is necessary. Translation takes place otherwise. This improves the efficiency of Binary Translation.




Courtesy: [4]

**Figure 25.2: Binary Translation**

## 4. CPU Virtualization

With the above methods being available, it is time to see what the cloud industry has actually used to virtualize various hardware environments. As expected, the industry responded to the need of virtualization and came up with multiple solutions, some of

which took advantage of the methods discussed above, yet others came up with unconventional solutions. Over all, there are four methods in CPU virtualization:

1. Emulation with Interpretation.
2. Full virtualization with dynamic binary translation.
3. Para virtualization.
4. Hardware-assisted virtualization.

### *4.1. Emulation with Interpretation*

This is the virtualization at the ISA level as we have discussed in our earlier discussion on the implementation of hardware virtualization. Although they are called virtualization, this mechanism is largely used when a program that is written for one ISA needs to run on a different ISA. For example, one may need to run an android application written for Android platform in a Windows-based machine. An appropriate emulator may be run on a Windows box. Some of the very popular platforms available are Bochs and QEMU.

Of these two, QEMU, besides being an emulator, even has a virtualizer, which attempts to bypass the emulator and run some of the x86 codes natively.

### *4.2. Full Virtualization with Dynamic Binary Translation*

The requirement in a virtualized environment is to run an existing operating system along with applications in an isolated virtual machine. We should be able to run many such VMs in such a way that these do not affect each other's performance running on the same hardware. In an x86 architecture, the guest OSs and VMs are not executed in ring 0. There are two issues out of this action. The first is that these guest OSs should be supervised by a host OS running in ring 0. Hence VMM is run in ring 0. Where do we run the guest OS? These must run deprivileged in either ring 1 or 2. Typically the application is run in ring 3 and guest OS in ring 1.This methodology is termed as *full virtualization*.

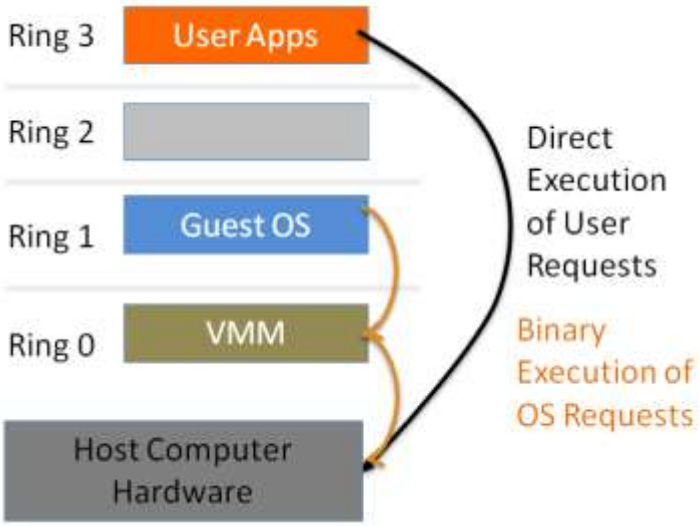
However, deprivileging an OS comes with an associated cost. Since the guest operating systems are written for execution on top of the hardware (ring 0), many instructions may be unsafe or even potentially harmful to be run in user mode. Simple traps for all these 'sensitive' instructions may not work (as per Popek-Goldberg). To ensure safety of the virtualized environment, all those instructions that can cause problems must be intercepted and rewritten, if required. Hence complete binary translation of the kernel code of the guest OSs is required to ensure the safety of the processor and the machine while allowing the user code to run natively on the hardware. Further, a guest OS could easily determine that it is not running at privilege level 0. Since this is not desirable, the VMM must take an appropriate action. Another problem of deprivileging the OS is that, as normal OS, the guest OS program expects to enjoy unrestrained access to the full memory but as a mere application, running in a higher ring, it cannot enjoy the same privilege. Hence the VMM must make way for ensuring that this is taken care of.

This method is called the full virtualization with binary translation. In this virtualization one or more guest operating systems of virtual machines share hardware resources from the host system. The presence of the hypervisor beneath is not known to the guests.

However, the issue that restricts full virtualization with binary translation is the performance. Translation takes time and translating all the kernel codes of the guest OS is expensive in terms of performance. This problem is resolved by using dynamic binary translation.

In dynamic binary translation, a block of code is used. These blocks may or may not have critical instructions. For each block, dynamic BT translates critical instructions, if any, into some privilege instructions, which will trap to VMM for further emulation. Full virtualization technology uses and exploits dynamic binary translation.

The execution of instructions in full virtualization using dynamic binary translation is explained in the Figure 25.4.




Courtesy: [2]

**Figure 25.3: Full Virtualization with Dynamic BT**

The virtual machine monitor or hypervisor executes in ring 0, guest OS in ring 1 and application in ring 3. The black arrow shows the direct execution of user request on the hardware. The orange line shows the binary translation of the OS code. The VMM translates the guest OS's instructions from ring 1 and passes to hardware for execution.

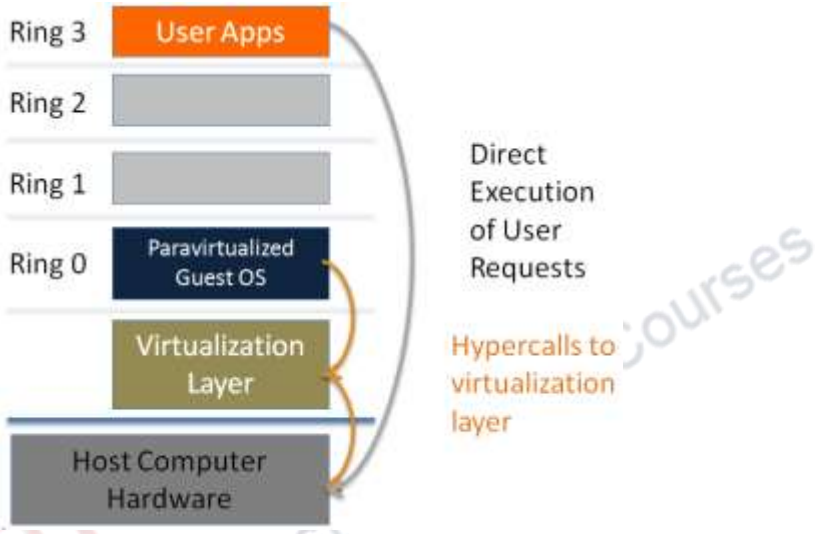
Although this methodology provides performance, there are shortcomings in this methodology. Binary translation does not work for certain cases where the guest OS may be using self-modifying or self-referencing codes. Real-time systems also cannot be virtualized since such systems cannot tolerate the delays caused by the translation.
For the above-mentioned scenario, other methods of virtualization are to be adopted.

### *4.3. Para Virtualization*

The problem of full virtualization is that the guest OS is unaware of the fact that it has been diprivileged and hence its behaviour continues to be the same. In para virtualization, the guest OS is modified or patched for virtualization. Hypervisor sits as the base OS or in ring 0 in case of x86 and guest OS resides on top of VMM. Here, since the Guest OS is aware that it is running above VMM rather than on top of the physical machine, many problems of full virtualization is taken care of.The

modified kernel of the guest OS is able to communicate with the underlying hypervisor via special calls. These special calls are provided by specific APIs depending on the hypervisor employed. These special calls are equivalent to system calls generated by an application to a non virtualized OS. Xen Hypervisor is an example that uses paravirtualization technology.

The Guest OS is modified and thus run kernel-level operations at Ring 1. The guest OS is now fully aware of how to process both privileged and sensitive instructions. Hence the necessity for translation of instructions is not present any more. Guest OS uses a specialized call, called “hypercall” to talk to the VMM. VMM executes the privileged instructions. Thus VMM is responsible for handling the virtualization requests and putting them to the hardware.




Courtesy [2]

**Figure 25.4: Paravirtualization**

Paravirtualization mode in the Figure 25.4 shows the hypercalls using orange lines, while the grey line shows the native execution of the user process as usual. Hypercalls are passed onto the hypervisor in the virtualization layer and VMM passes the calls to hardware after processing. Theissue hereis the fact that the guest OS must be modified. This causes problems in OS maintainability and supportability. Also, since the guest OS and hypervisor are tightly coupled, compatibility problem arises. Each time the hypervisor is updated, guest OS must also be recompiled.

### *4.4. Hardware-Assisted Virtualization*

Legacy processors are not designed for virtualization. Hence we observed that whatever the methods that may be applied for implementing virtualization, each has its own problems. However, if the processors are made virtualization-aware, the VMM design will be more efficient and simple. Many issues mentioned in the earlier sub-section can be easily taken care of with such a processor.

This is the reason why hardware vendors rapidly embraced virtualization and developed new features to simplify virtualization techniques. The two giants in the hardware arena, Intel and AMD came up with came up with designs of new CPU execution mode that allows VMM to run in a new root mode below ring 0. This is the way to handle the privileged mode.In this new design, both privileged and sensitive calls automatically trap to the hypervisor. Hence, in this new design, there remains

no need for either binary translation or paravirtualization. Now, the latest x86's meet Popek & Goldberg requirements, hence they can be virtualized without any complexities. Examples of this new design are Intel VT-x (2005) and AMD-V (2006).

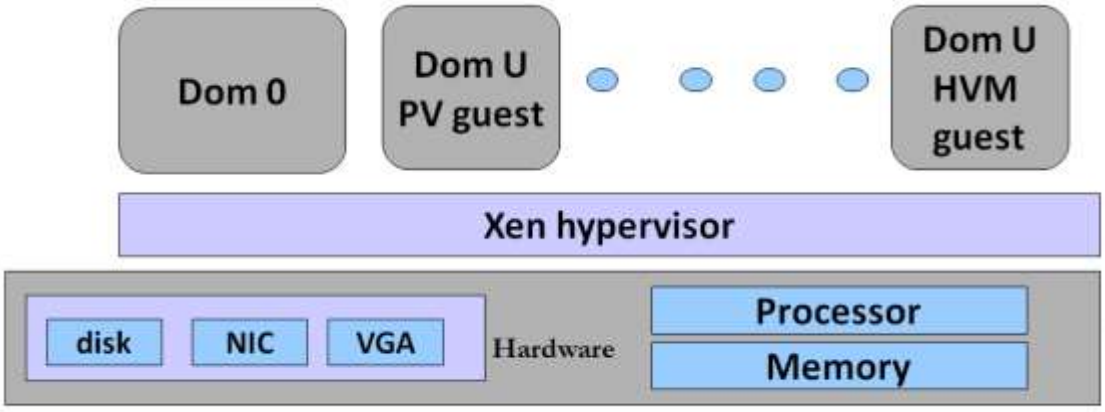
Intel VT-x have two modes of operations: VMX root and VMX non-root. While “VMX root” mode operation executes the hypervisor / VMM in the ring 0, “VMX non-root” mode operation executes the guest OS, also in ring 0, thereby removing the need to deprivilege the guest OS. Both the modes support all privilege rings and are identical. Unmodified guest OS runs in ring 0 in non-root mode and traps instructions to root mode. The privileged and sensitive calls automatically trap to the hypervisor. VMM controls the execution of the guest OS.

While it is possible to run unmodified guest OS in the above set up thereby allowing execution of legacy/unmodified operating system, the challenge in hardware assisted virtualization is that an unmodified OS cannot take the advantage of virtualization. The OS is unaware of whether it is running in a virtualized environment or conventional environment. Partial implementation of paravirtualization may solve this problem.

Now that we have seen how the four types are used in four different environments to implement virtualization, let us look into one such implementation, i.e., paravirtualization by Xen.

### 5. Case Study: Xen

Xen is a free and open source hypervisor widely employed in virtualization environment. The name of this software, Xen, is derived from **neXtgENeration.** Xen Hypervisor adopts paravirtualization and Hardware-assisted full virtualization. Figure 25.6 explains the architecture of Xen.




Courtesy [5]

**Figure 25.6: Xen Architecture**

Xen hypervisor resides on top of the hardware and is responsible for handling all the low-level functionalities. The hardware comprises storage in the form of disks, network interface card, graphics-display adapters, processor and main memory. The whole environment is divided into domains or virtual machines in which the guest OSs reside and these are called Doms or domains. Dom 0, which hosts the most important operating system, is a privileged domain and is responsible for the creation of other new domains. The OS in Dom 0 is a modified privileged guest operating

system inbuilt with the hypervisor. Domain 0 is designed to access hardware directly and manage devices.

All other virtual machines are Dom U which is the guest operating system. These also interact with hardware through the hypervisor. The physical hardware is directly inaccessible for the guest OSs and VM. Linux, Solaris, FreeBSD, UNIX are categorized as paravirtualized VMs whereas Windows VMs are categorized as full virtualized VMs – DomU HVM guests.

Xen is widely employed as a hypervisor with KVM – Kernel Virtual Machine as an alternate free and open source hypervisor. Xen is widely adopted in cloud and virtualization environment. Some of the enterprises supporting Xen are Sun Microsystems, Hewlett-Packard, Novell, Red Hat, Intel, Advanced Micro Devices, Voltaire, IBM.

**6. Summary**

In this module, we first explored various ways of implementating virtualization. We observed that emulation technique is the basis for virtualization, which uses interpretation and translation approaches. In this module, the generic methods of virtualization are discussed first with greater emphasis on the practical implementation of virtualization as adopted by the industry. Among these, full virtualization with dynamic binary translation and paravirtualization are the most used methods. However certain issues still persists, hence at certain instance hardware assisted virtualization. In hardware assisted virtualization, the VT-x approach is followed in VM-x root and non-root mode. As a case study, we briefly looked at Xen, an open-source free hypervisor based on paravirtualization.

In the next two modules we investigate the next topic that is also very important in cloud environment, called web services.

## References


1. Smith, J. E., and Ravi Nair, "Virtual Machines: Architectures, Implementations and Applications", Morgan Kauffmann, 2004.
2. Marshall, David. "Understanding Full Virtualization, Paravirtualization, and Hardware Assist", VMWare White Paper, 2007.
3. Figueiredo, Renato, and Peter A. Dinda, "Guest Editors' Introduction: Resource Virtualization Renaissance", Computer 5: 28-31, 2005.
4. Greg Gagne, Peter B. Galvin, A. SIlberschatz, “Operating System Concepts”, 9th Edition, John Wiley & Sons, 2012.
5. Hwang, Kai, Jack Dongarra, and Geoffrey C. Fox, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things”, Morgan Kaufmann, 2013.