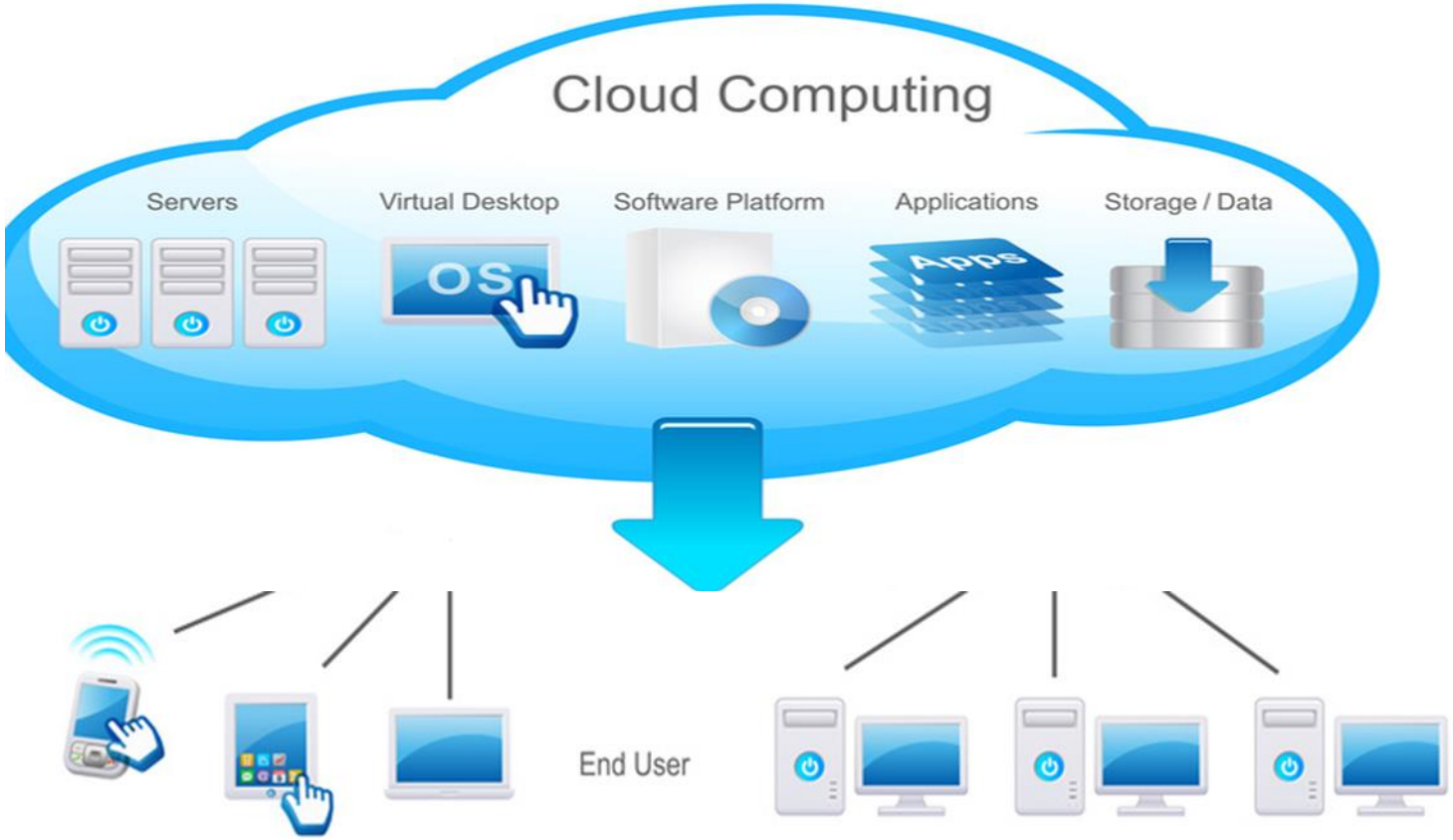
CLOUD COMPUTING

Prescribed Text Book					
Sl. No.	**Book Title**	**Authors**	**Edition**	**Publisher**	**Year**
1	Cloud Computing a Hands on Approach	ArshdeepBahga, Vijay Madisetti	1st Edition	University Press	2013
2	Virtual Machines	James E Smith, Ravi Nair	1st Edition	Morgan Kaufmann Publishers	2006.
3	Distributed and cloud Computing from Parallel Processing to the Internet of Things	Kai Hwang, Geoffrey C. Fox, Jack J Dongarra	1st Edition	Morgan Kaufmann, Elsevier	2013
4	Mastering Docker	Scott Gallagher, Russ McKendrick	2nd Edition	Packt	2017
5	Mastering Kubernetes,	Gigi Sayfan	3rd Edition	Packt	2020

Reference Text Book					
Sl. No.	**Book Title**	**Authors**	**Edition**	**Publisher**	**Year**
1	"Cloud Computing – Principles and Paradigms"	RajkumarBuyya, BrobergAndrzejGoscinski,	1st Edition	WILEY: A JOHN WILEY & SONS, INC.,	2011

CLOUD COMPUTING

- Cloud computing is a **service provisioning technique** where **computing resources** like **hardware such as servers and storage devices**, **software's** and **complete platform for developing applications** are provided **as a service by the cloud providers to the customers.**



CLOUD COMPUTING (Cont…)

- Customers **can use these resources as and when needed**, **can increase or decrease resource capacities dynamically** according to their requirements and **pay only for how much the resource were used**.

- Customers **no need to invest money to purchase**, **manage** and **scale infrastructures**, **software upgradation** and **software licensing.**

Cloud Service Models

- The services that are provided by the cloud providers are broadly classified into three categories:

  - **Infrastructure-as-a-Service (IaaS)**

  - **Platform-as-a-Service (PaaS)**

  - **Software-as-a-Service (SaaS)**

Cloud Service Models

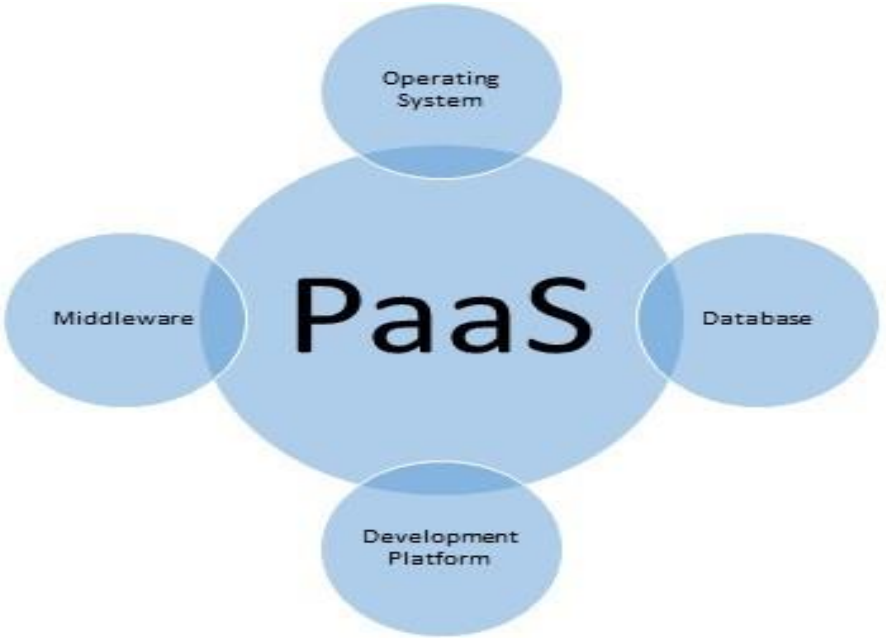
- **Infrastructure-as-a-Service (IaaS):** In Infrastructure-as-a-Service model, the service provider owns the hardware equipment's such as **Servers, Storage**, **Network** and is **provided as services** to the clients. The **client uses these equipment's and pays on per-use basis.**

- E.g. **Amazon Elastic Compute** (EC2) and **Simple Storage Service** (S3).

Cloud Service Models

- **Platform-as-a-Service (PaaS):** In Platform-as-a-Service model, **complete resources** needed to **Design**, **Develop**, **Testing**, **Deploy** and **Hosting** an application are provided as services **without spending money for purchasing and maintaining the servers, storage and software**.

- **PaaS is an extension of IaaS**. **In addition to the fundamental computing resource supplied by the hardware** in an IaaS offering, **PaaS models also include the software and configuration required to create an applications**.



- E.g. **Google App Engine**.

Cloud Service Models

- **Software-as-a-Service (SaaS):** In Software-as-a-Service model, the service provider provides **software's** as a service over the Internet, eliminating the need to **buy**, **install**, **maintain**, **upgradation and licensing** on their local machine.

- E.g. **Accounting, CRM, Google Docs** are all popular examples of SaaS.

Cloud Deployment Models

- Mainly there are four cloud deployment models (4 ways we can create/organize a cloud)
  - **Public Cloud**
  - **Private Cloud**
  - **Community Cloud**
  - **Hybrid Cloud**

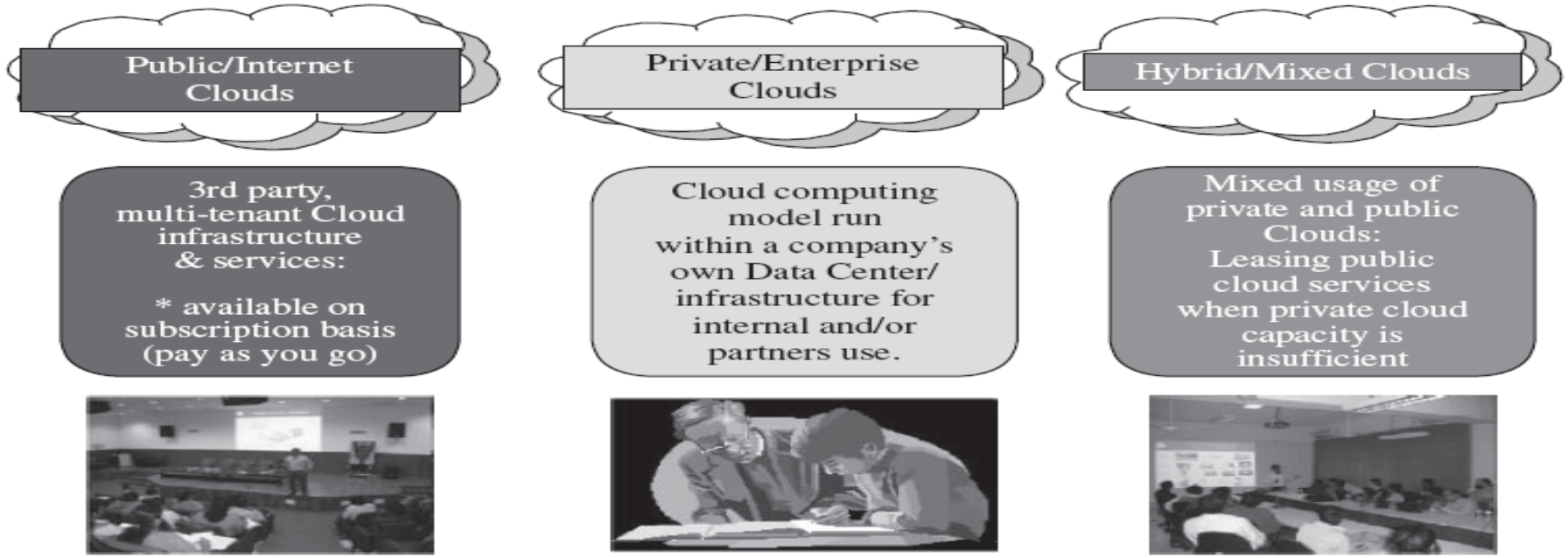


FIGURE 1.4. Types of clouds based on deployment models.

Cloud Deployment Models (Cont…)

- **Public Cloud**: A public cloud is a cloud in which **services and infrastructure are hosted off-site by a cloud provider** (owned by an organization selling cloud services) and easily **accessible to general public via internet**.

- **Private Cloud:** Private Cloud is a cloud where **services and infrastructure are operated for a single operation accessible via private network**, managed internally or by a third party. It is greater level of security.

Cloud Deployment Models (Cont…)

- **Community Cloud:** Community Cloud is a cloud where **services and infrastructure are accessible by a group of organizations**.

- **Hybrid Cloud:** Hybrid Cloud is a cloud which is a **mixture of private and public cloud**. In this type of cloud **all critical and sensitive applications and data are stored in private cloud** and **non critical and non sensitive applications and data are stored in public cloud**.

Features of Cloud Computing

- **It is elastic:** Cloud computing is flexible in nature, where users can **scale up** and **scale down** the resources as needed.
- **Pay per use:** Usage is metered and user **pays only for how much the resources were used**.
- **Operation:** The services are completely **handled by the provider**.
- **Reduce capital cost:** No need to **invest money** on purchasing and maintaining of hardware and software, software licensing, training required for IT staff.
- **Remote accessibility:** Users can access **applications and data stored on cloud** from anywhere any time worldwide through a device with internet connection.
- **Better use of IT staff:** Staff with in enterprise need not worry on purchasing and maintaining of servers, softwares, up gradation of servers and softwares, software licensing etc., instead they can concentrate more on work.

Cloud Services Examples:
IaaS-Amazon EC2, Google Compute Engine, Azure VMs

Amazon EC2

- **Amazon Elastic Compute Cloud** is an Infrastructure as a Service offering from Amazon.
- EC2 is a web service that **provides a computing capacity in the form of virtual machines**.
- Amazon EC2 allows **users to launch instances on demand using a simple web based interface**.
- Amazon provides **pre-configured Amazon Machine Images** (AMIs) which are templates of cloud instances.

	Small	Large	Extra Large	High CPU-Medium
Compute unit	1	4	8	5
Memory	1.7 GB	7.5 GB	15 GB	1.7 GB
Storage	160 GB	850 GB	1690 GB	350 GB
Platform	32 bit	64 bit	64 bit	32 bit

- Users can also **create their own AMIs with custom applications, libraries and data**.

Cloud Services Examples:
IaaS-Amazon EC2, Google Compute Engine, Azure VMs

Amazon EC2

- Amazon EC2 also provides **instances with high memory, high CPU resources, Cluster Compute instances, Cluster Graphical Processor unit (GPU)instances and high Input/Output instances**.

- Instances can be launched with a **variety of operating systems**.

- Users can **load their applications on running instances** and rapidly and easily **increase or decrease capacity to meet dynamic application performance requirements**.

- With EC2, **users can even provision hundreds or thousands of server instances simultaneously**, manage network access permissions and monitor usage resources through web interface. (Create 2 VMs at 3 different places and running applications, creating network among 3 and allowing data transfer among 3 VMs and monitoring resource usage)

Cloud Services Examples:
IaaS-Amazon EC2, Google Compute Engine, Azure VMs

Amazon EC2

- Amazon EC2 provides **instances of various computing capacities ranging from Small Instances** (Eg: 1 Virtual core with 1 EC2 compute unit, 1.7 GB memory and 160 GB instance storage) to **Extra Large Instances** (Eg: 4 Virtual cores with 2 EC@ compute unit each with 15GB memory and 1690 GB instance storage).

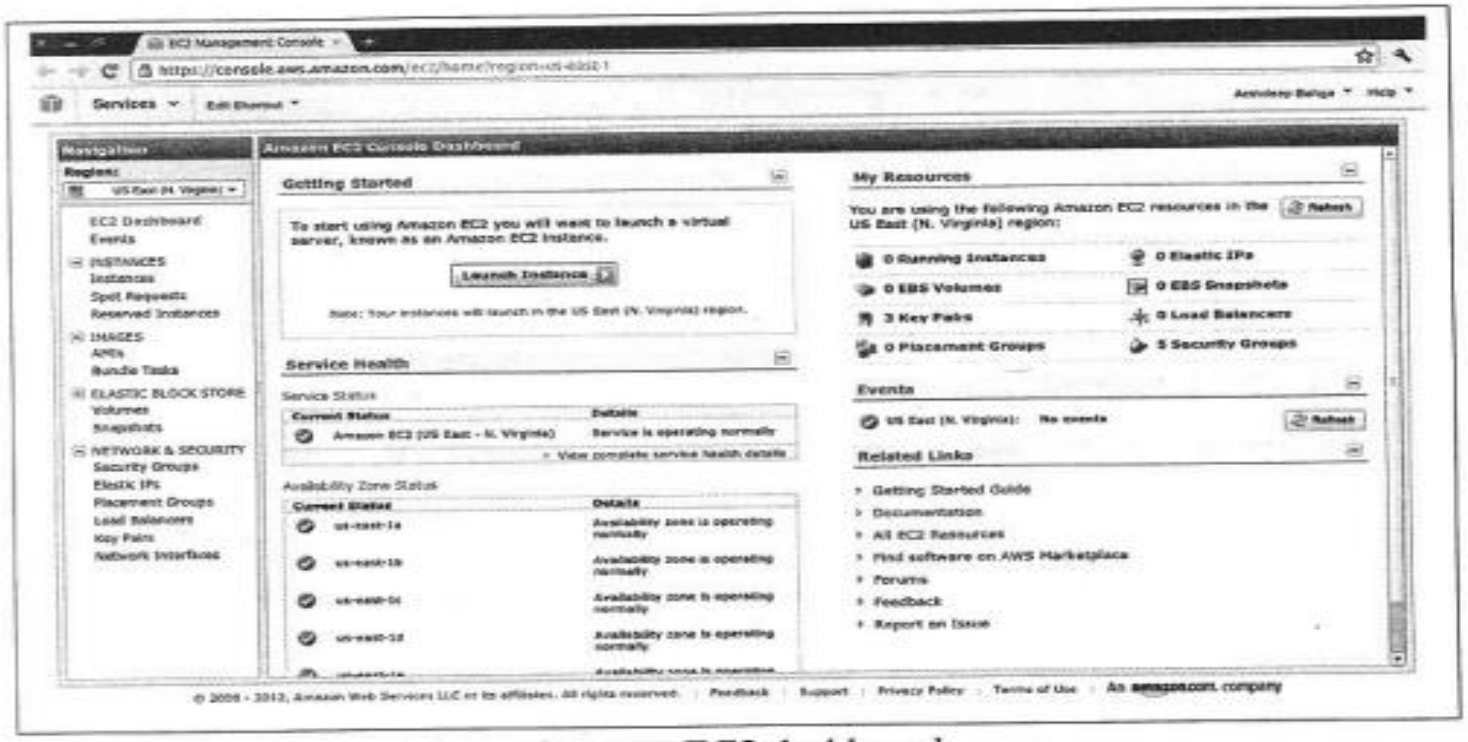
- The **pricing model for EC2 instances is based on Pay-Per Use model**. Users are billed based on the number on instance hours used for on demand instances.

- EC2 also provides **spot instances that allow users to bid on unused Amazon EC2 capacity and run those instances for as long as their bid exceeds the current spot price**.

Cloud Services Examples:
IaaS-Amazon EC2, Google Compute Engine, Azure VMs

Amazon EC2

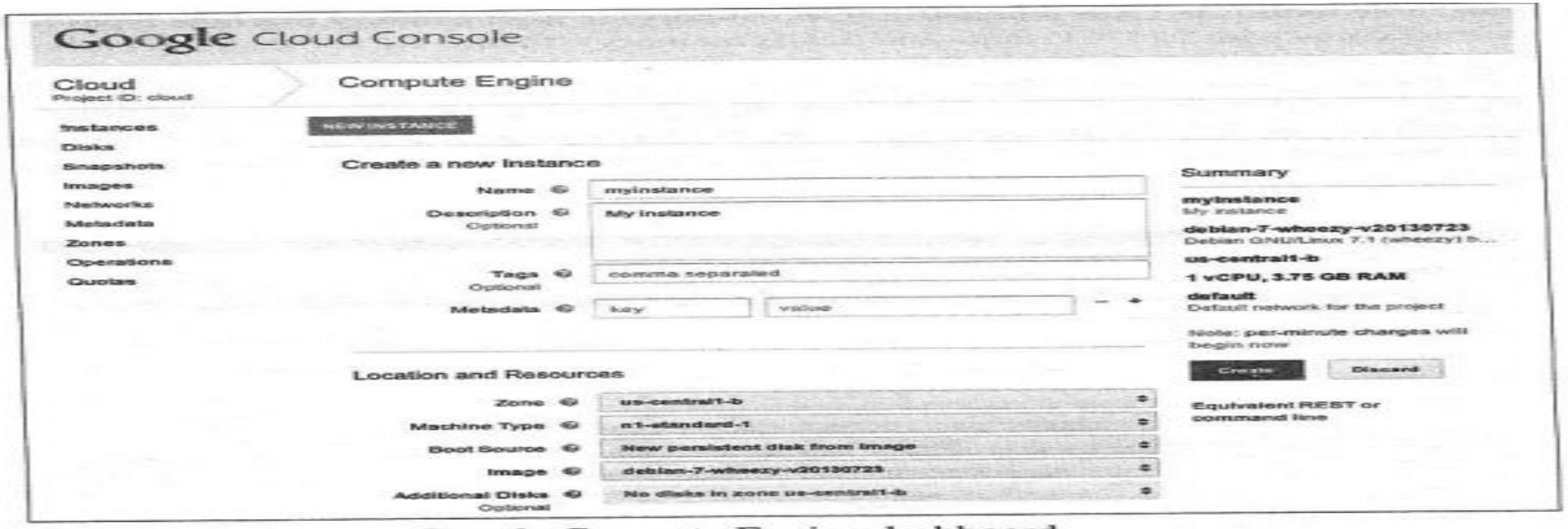
- The below figure shows screenshot of Amazon EC2 dashboard



Amazon EC2 dashboard

Cloud Services Examples:
IaaS-Amazon EC2, Google Compute Engine, Azure VMs
Google Compute Engine

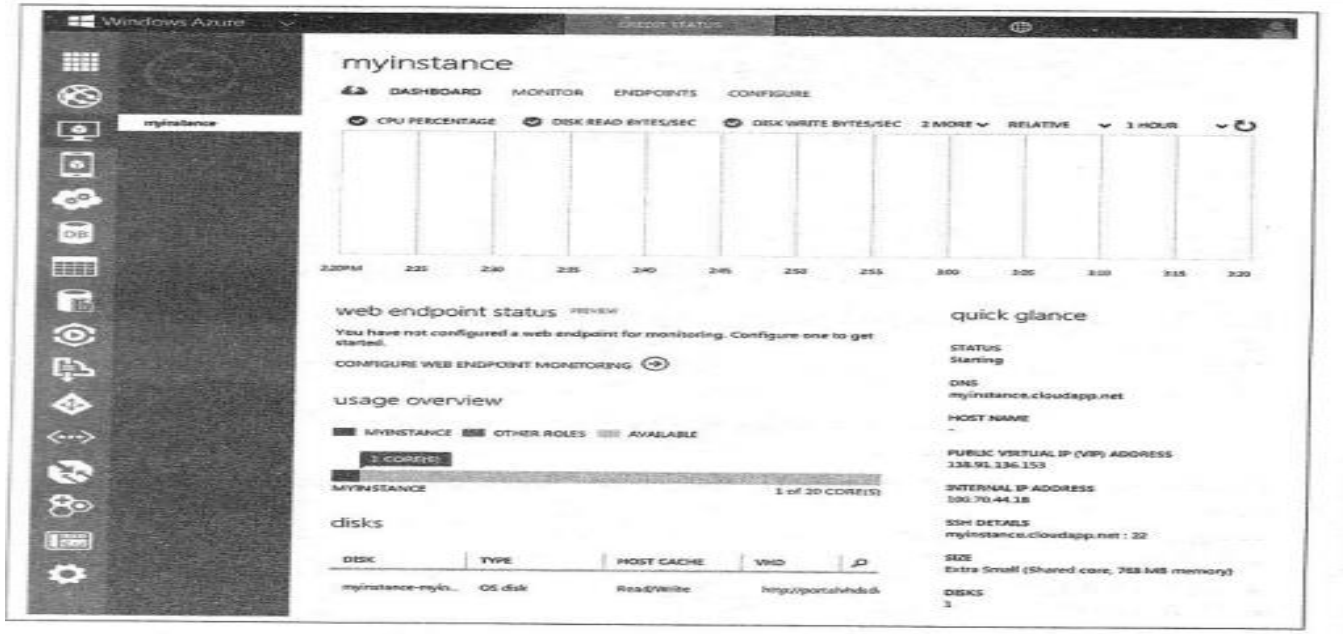
- **Google Compute Engine (GCE) is an IaaS offering from Google**.
- GCE provides **virtual machines of various computing capacities ranging from small instances** (Eg: Virtual core with 1.38 GCE unit and 1.7 GB memory) **to high memory machine types** (8 virtual cores with 22 GCE unit and 52 GB memory.
- The below figure shows screenshot of Google Compute Engine dashboard



Google Compute Engine dashboard

Cloud Services Examples:
IaaS-Amazon EC2, Google Compute Engine, Azure VMs
Windows Azure

- **Windows Azure Virtual Machine is an IaaS offering from Microsoft**.
- Azure VMs provides **virtual machines of various computing capacities ranging from small instances** (1 virtual core with 1.75GB memory) to **memory intensive machine types** (8 virtual cores with 56GB memory).
- The below figure shows screenshot of Google Compute Engine dashboard.



Windows Azure Virtual Machines dashboard

Cloud Services Examples:
PaaS-Google App Engine

- **Google App Engine (GAE) is a Platform as a Service offering from Google**.

- GAE is a cloud based **web service for hosting web applications and storing data**.

- GAE allows users to **build scalable and reliable applications that run on the same systems that power Google's own applications**.

- GAE provides a **Software Development Kit (SDK) for developing web applications software that can be deployed on GAE.**

Cloud Services Examples: PaaS-Google App Engine

- Developers can **develop and test their applications with GAE SDK on a local machine** and **then upload it to GAE** with a simple click of a button

- Applications hosted in GAE are **easy to build, maintain and scale**. Users **don't need to worry about launching additional computing instances when the applications load increases**.

- GAE provides **automatic scaling and load balancing capability**.

- GAE **supports applications written in several programming language**.

- With Java runtime environment developers can **build applications using Java programming language and standard Java technologies such as Java Servlets**. GAE also **provides runtime environment for Python programming languages**.

Cloud Services Examples: PaaS-Google App Engine

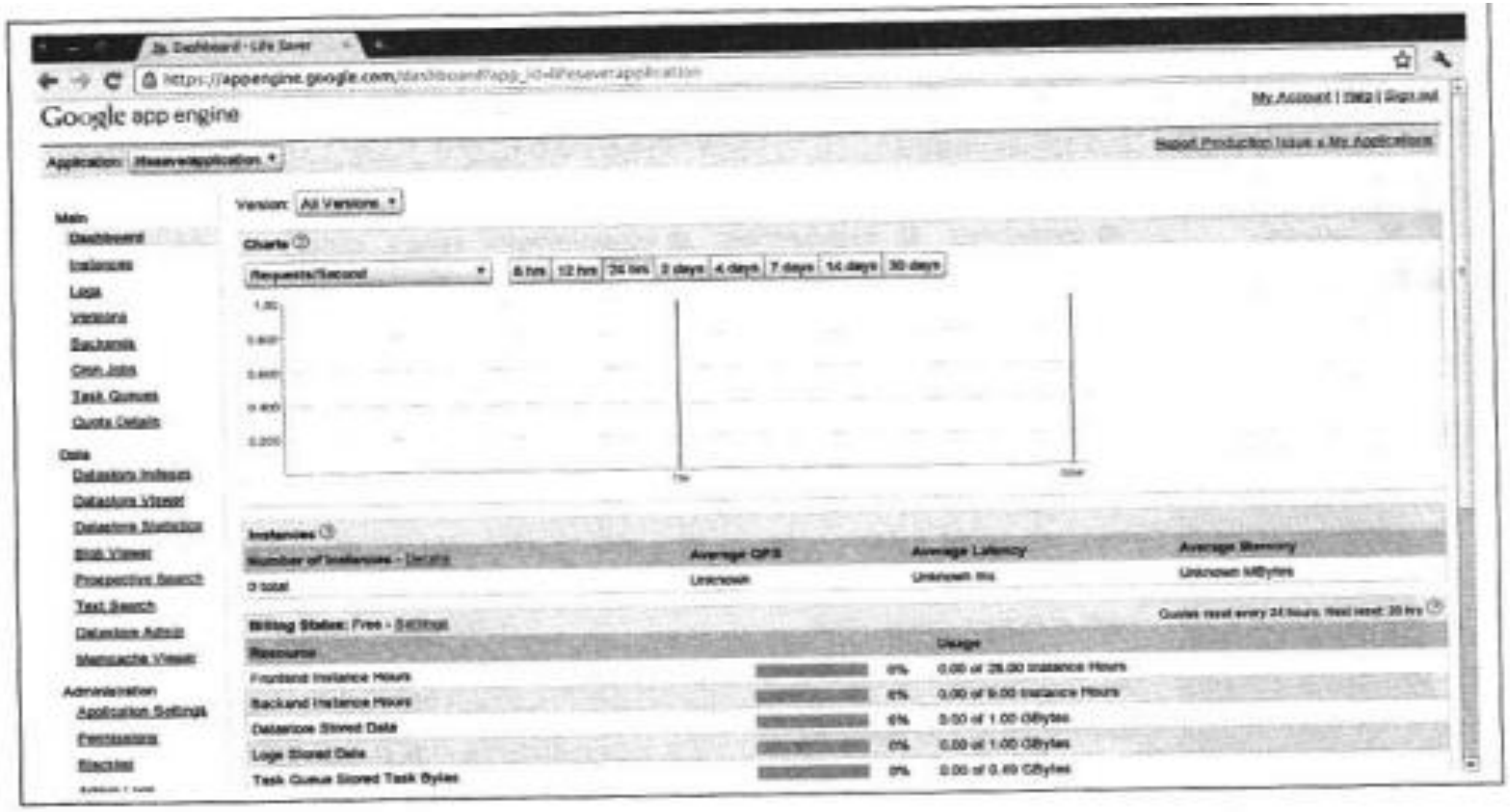
- **Applications hosted in GAE run in secure sandbox** with limited access to the underlying operating system and hardware.

- The **pricing model for GAE is based on the amount of computing resources used**.

- GAE provides **free computing resources for applications up to a certain limit**. **Beyond that limit, users are billed based on the amount of computing resources used such as amount of bandwidth consumed, number of resource instance hours, amount of data stored**.

Cloud Services Examples: PaaS-Google App Engine

- The below figure shows the screenshot of GAE dashboard.



Google App Engine dashboard

Cloud Services Examples: SaaS-Salesforce

Salesforce Sales Cloud

- Salesforce **Sales Cloud** is a cloud based **Customer Relationship Management (CRM) SaaS offering**.

- Users can **access CRM application from anywhere through internet enabled devices such as workstations, laptops, tablets and smartphones.**

- Sales Cloud allows **sales representatives to manage customer profiles**, **track opportunities**, **optimize campaigns from lead to close** and monitor the impact of campaigns. (A lead can be a company or an individual who has expressed interest in a company's product and/or service).

Cloud Services Examples: SaaS-Salesforce

Salesforce Service Cloud

- **Salesforce Service Cloud is a cloud based Customer Service Management SaaS**.

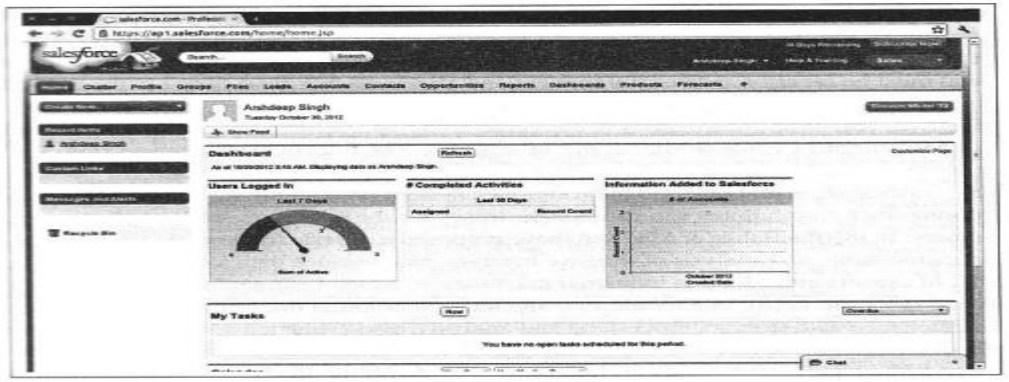
- Service cloud provides companies a call center like view and allows creating, tracking, routing and escalating cases.

- Service cloud includes a **social networking plug-in** that enables social customer service **where comments from social media channels can be used to answer customer questions**.

Cloud Services Examples: SaaS-Salesforce

Salesforce Marketing Cloud

- **Salesforce Marketing Cloud is cloud based social marketing SaaS**.
- Marketing cloud allows companies to **identify sales leads from social media**, discover advocates, identify most trending information on any topic.
- Marketing cloud **allows companies to pro-actively engage with customers**, manage **social advertisement campaigns** and track the performance of social campaigns.
- The below figure shows a screenshot of Salesforce dashboard



Salesforce dashboard

Cloud Services Examples: SaaS-Salesforce

Salesforce Marketing Cloud

- **Some of the tools included in the Salesforce Sales, Service and Marketing Clouds include**
  - **Accounts and Contacts**
  - **Leads**
  - **Opportunities**
  - **Campaigns**
  - **Chatter**
  - **Analytics an Forecasts**

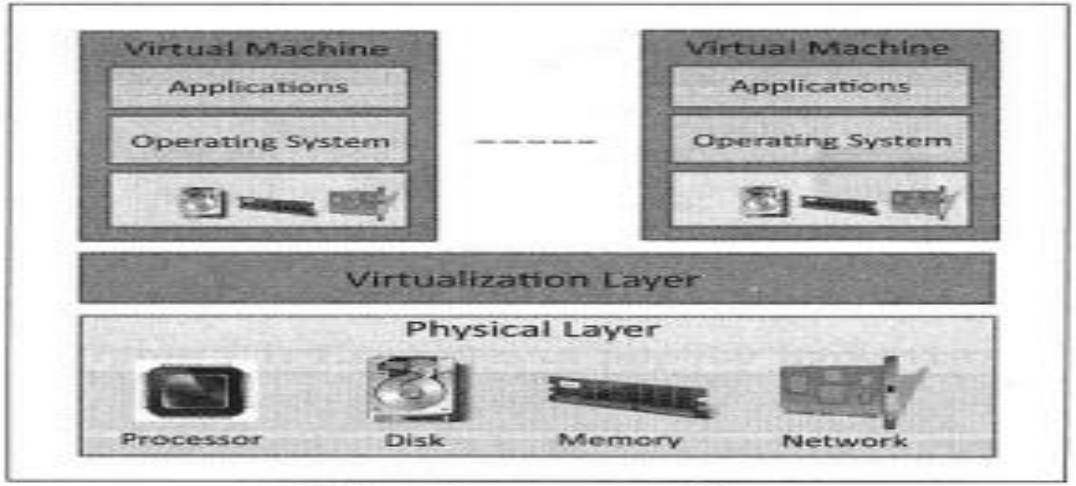
Cloud Concepts and Technologies

- **Virtualization**
- **Load Balancing**
- **Scalability and Elasticity**
- **Deployment**
- **Replication**
- **Monitoring**
- **Software Defined Networking**
- **MapReduce**
- **Identity and Access Management**
- **Service Level Agreements**
- **Billing**

Cloud Concepts and Technologies

Virtualization

- Virtualization refers to **the partitioning the resources of a physical system** (such as computing, Storage, Network and Memory) **into multiple virtual resources**.
- In cloud computing, **resources are pooled to serve multiple users using Multi-Tenancy**.
- Multi-Tenant aspects of the cloud **allow multiple users to be served by the same physical hardware**.
- The below figure shows the architecture of a virtualization technology in cloud computing.
- The physical resources such as **computing, storage, memory and network resources are virtualized**.
- The **virtualization layer partitions the physical resources into multiple virtual machines**.

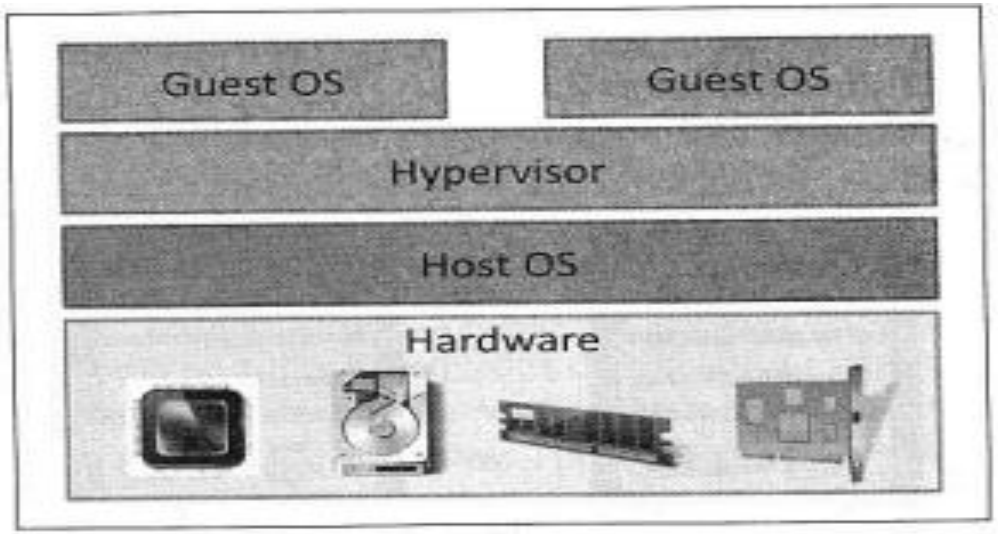


Virtualization architecture

Cloud Concepts and Technologies

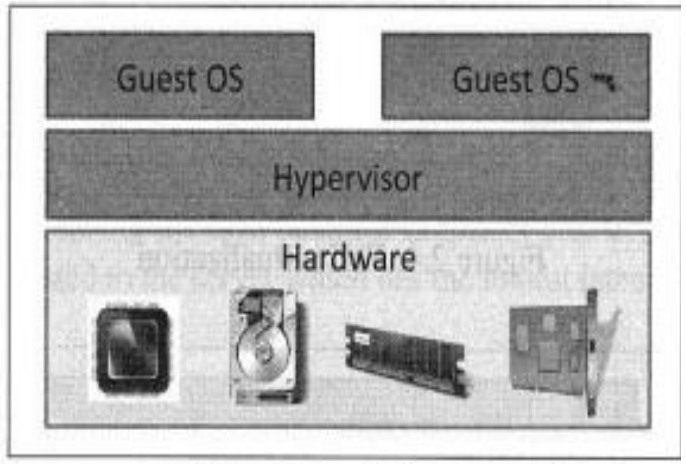
Virtualization: Guest Operating System

- A **guest OS is an operating system that is installed in a virtual machine** in addition to the host OS.
- In virtualization, the **guest OS can be different from the host OS**.

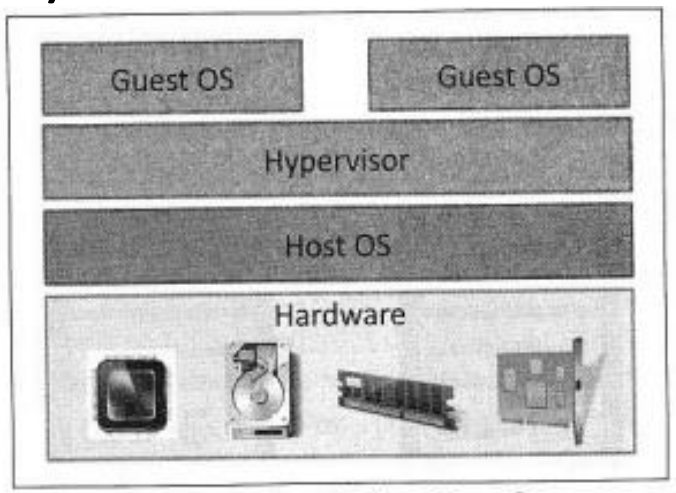


Cloud Concepts and Technologies
Virtualization: Hypervisor

- The **virtualization layer consists of a hypervisor or a Virtual Machine Monitor (VMM)**.
- There are two types of hypervisors
  - **Type-1 Hypervisors or Native Hypervisors**
  - **Type-2 Hypervisors or Hosted Hypervisors**



Hypervisor design: Type-1



Hypervisor design: Type-2

Type-1 Hypervisors or Native Hypervisors

- **Type-1 Hypervisors or Native Hypervisors run directly on the host hardware** and **control the hardware** and **monitor the guest operating system**.

Type 2 Hypervisors or Hosted Hypervisors

- **Type 2 Hypervisors or Hosted Hypervisors run on top of a conventional** (main or Host) **operating system** and **monitor the guest operation systems.**

Cloud Concepts and Technologies

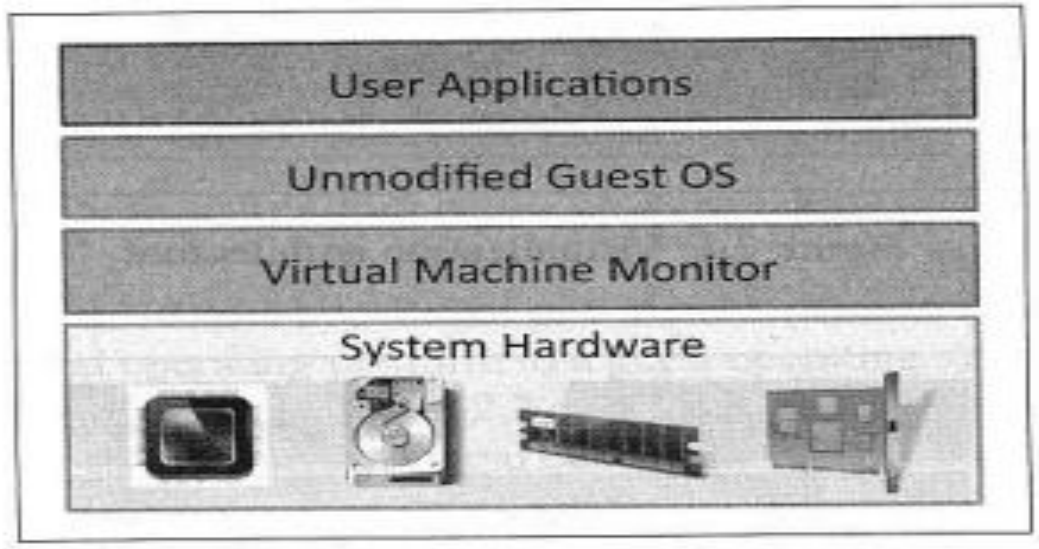
Virtualization: Guest Operating System

- Various forms of virtualization approaches exist:
  - **Full Virtualization**
  - **Para-Virtualization**
  - **Hardware Virtualization**

Cloud Concepts and Technologies

Virtualization: Full Virtualization

- In Full Virtualization, **the guest OS requires no modification** and **is not aware that it is being virtualized**.
- **Full virtualization is enabled by direct execution of user requests** and **binary translation of OS requests**.
- **The below figure shows the Full virtualization approach**



Full virtualization

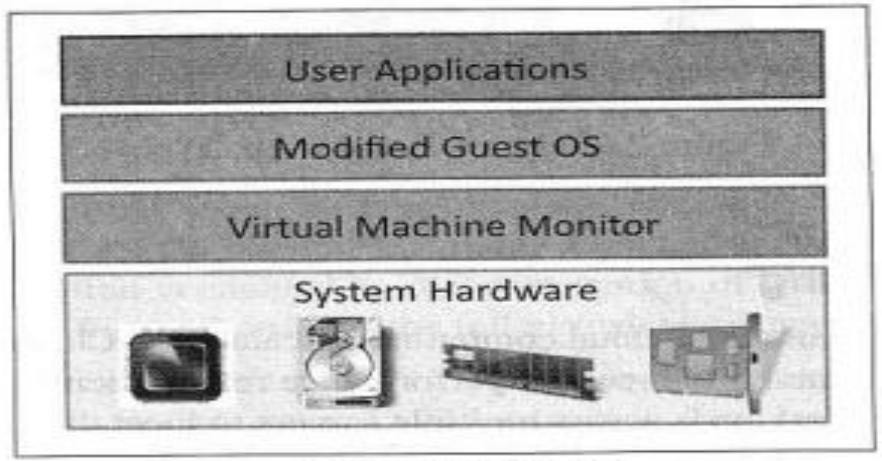
Cloud Concepts and Technologies

Virtualization: Para Virtualization

- In Para virtualization, **the guest OS is modified to enable communication with the hypervisor** to improve performance and efficiency.

- The **guest OS kernel is modified to replace non virtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor.**

- The **below figure shows the para virtualization approach**



Para-virtualization

Cloud Concepts and Technologies

Virtualization: Hardware Assisted Virtualization

- Hardware Assisted virtualization **is enabled by hardware features such as Intel's Virtualization technology** (VT-x) and AMD's **AMD-V**. In hardware virtualization, privileged and sensitive calls are set to automatically trap to the hypervisor.

- Thus, **there is no needed for either binary translation or Para virtualization**. Hardware-assisted full virtualization **eliminates the binary translation** and it **directly interrupts with hardware** using the virtualization technology which has been integrated on X86 processors since 2005 (Intel VT-x and AMD-V).

- **Guest OS's instructions might allow a virtual context execute privileged instructions directly on the processor, even though it is virtualized**.

Cloud Concepts and Technologies

Load Balancing Technique

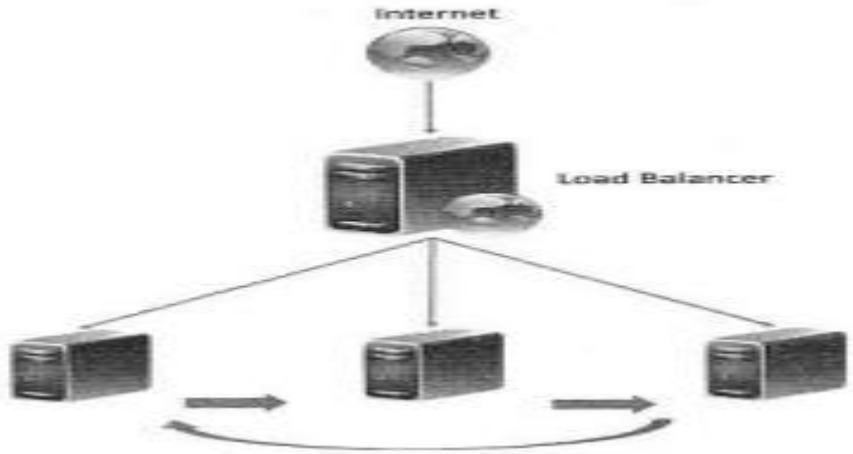
- One of the important features of cloud computing is **scalability**.
- Cloud resources can be **scaled up on demand to meet the performance requirements of applications**.
- **Load balancing distributes workload across multiple servers to meet the application workloads**.
- The **goal of load balancing techniques are to achieve maximum utilization of resources, minimize the response times, maximizing throughput**.
- Since multiple resources under a load balancer are used to serve the user requests, **in the event of failure of one or more of the resources**, the **load balancer can automatically reroute the user traffic to the healthy resources**.

Cloud Concepts and Technologies

Load Balancing Techniques

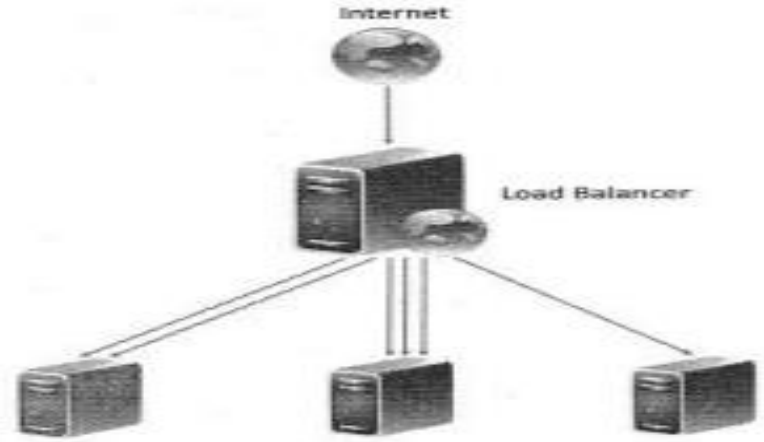
Round Robin

- **In round robin load balancing, the servers are selected one by one in a circular fashion to server the incoming requests from the user.**



Weighted Round Robin

- **In Weighted round robin load balancing, servers are assigned some weights. The incoming requests are proportionally routed to a server based on its weight.**

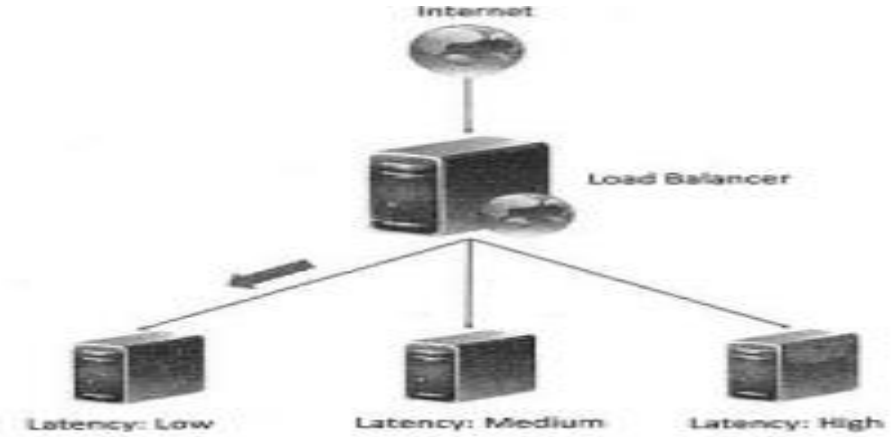


Cloud Concepts and Technologies

Load Balancing techniques

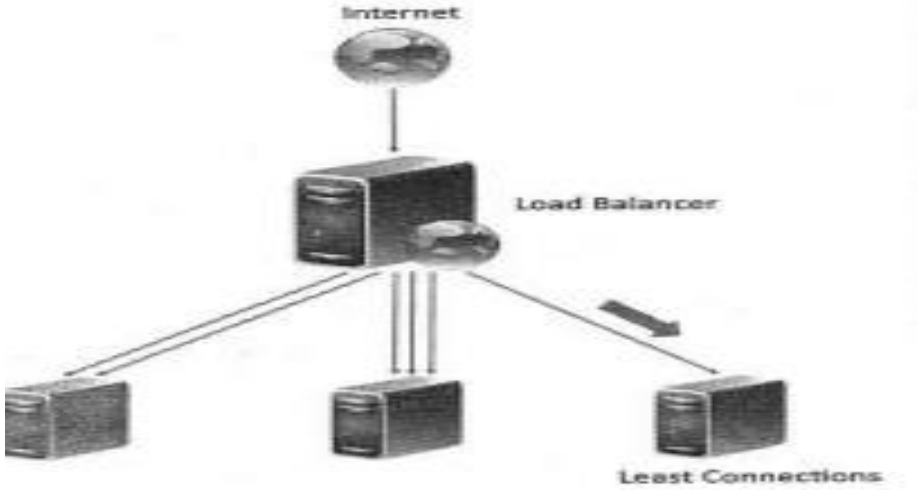
Low Latency

- **In low latency load balancing, the load balancer monitors the latency of each server and request is routed to a server which has lowest latency.**



Least Connections

- **In least connection load balancing, the incoming requests are routed to the server with least number of connections.**

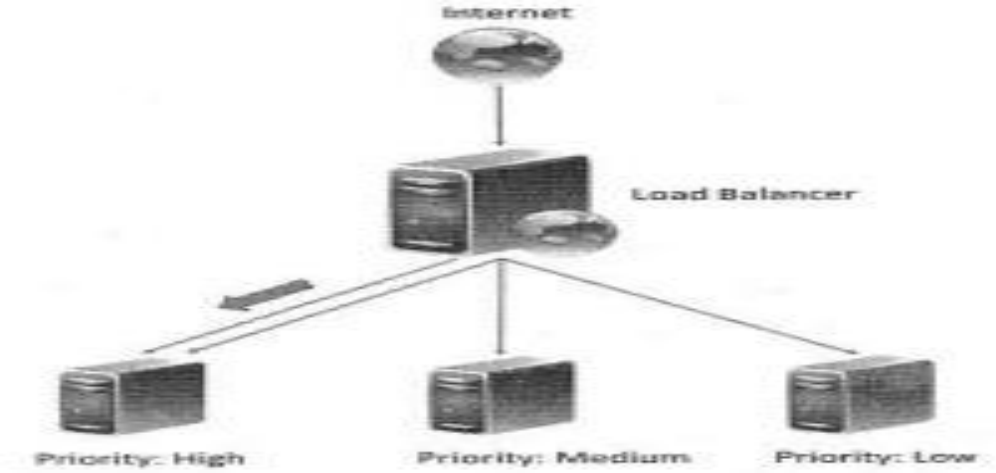


Cloud Concepts and Technologies

Load Balancing techniques

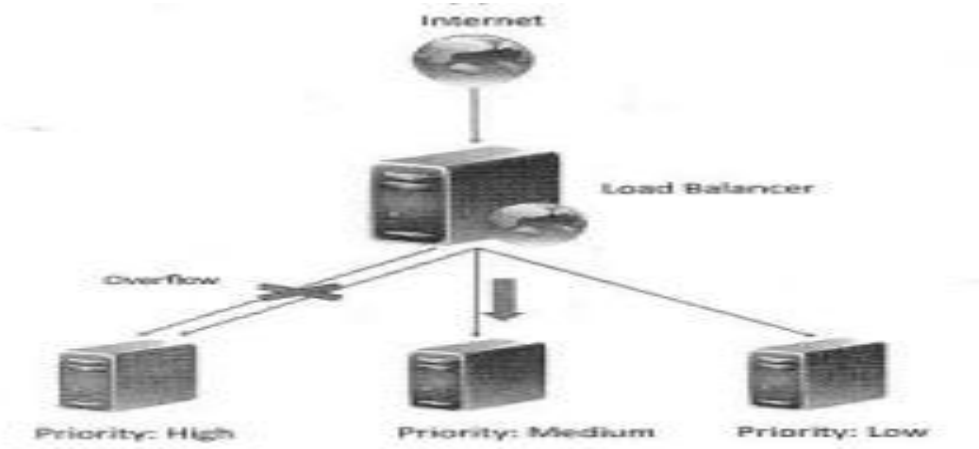
Priority

- **In priority load balancing, each server is assigned a priority. The incoming traffic is routed to the highest priority servers as long as the server is available. When the highest priority server fails, the incoming traffic is routed to a server with a lower priority.**



Overflow

- **Overflow load balancing is similar to priority load balancing. When the incoming request to high priority servers overflow, the requests are routed to a lower priority server.**



Cloud Concepts and Technologies

Load Balancing Techniques

- **A session is defined as a series of related browser requests that come from the same client during a certain time period.** Session tracking ties together a series of browser requests—think of these requests as pages—that may have some meaning as a whole, **such as a shopping cart application**.

- **A web session is a series of contiguous actions by a visitor on an individual website within a given time frame.** This could include your **search engine searches**, **filling out a form to receive content**, **scrolling on a website page**, **adding items to a shopping cart**, researching airfare, or which pages you viewed on a single website. **Any interaction that you have with a single website is recorded as a web session** to that website property.

- **To track sessions, a web session ID is stored in a visitor's browser**. This **session ID is passed along with any HTTP requests that the visitor makes while on the site** (e.g., clicking a link).

- ("Session" is the term used to refer to a visitor's time browsing a web site. It's meant to represent the **time between a visitor's first arrival at a page on the site and the time they stop using the site**.

- A **cookie is a small piece of data from a web site that is stored on a visitor's browser to help the website track the visitor's activity on the web site**.)

Cloud Concepts and Technologies

Load Balancing techniques

- For the **session based applications, an important issue to handle during load balancing is the persistence of multiple requests from a particular user session.** (Because you may go from current page to next page and back to previous page)
- **Since load balancing can route successive requests from a user session to different servers, maintain the state or the information of the session is important**. Four persistence approaches are:
  - **Sticky Sessions**
  - **Session Database**
  - **Browser Cookies**
  - **URL Re-Writing**

Cloud Concepts and Technologies

Load Balancing Techniques

Sticky Sessions

- In this approach, **all the requests belonging to a user session are routed to the same server**.
- **Theses sessions are called Sticky Sessions**.
- **The benefit of this approach is that it makes session management simple**.
- However, **a drawback of this approach is that if a server fails all the sessions belonging to that server are lost**, since there is **no automatic failover possible**.

Session Database

- In this approach, **the session information is stored externally in a separate session database**, which is often replicated **to avoid a single point of failure**.
- **Though, this approach involves additional overhead of storing the session information**, however unlike the sticky session approach, **this approach allows automatic failover**.

Cloud Concepts and Technologies

Load Balancing Techniques

Browser Cookies

- In this approach, **the session information is stored on the client side in the form of browser cookies**.
- **The benefit of this approach is that it makes the session management easy** and has the **least amount of overhead for the load balancer.**

URL Re-Writing

- In this approach, a **URL re-write engine stores the session information by modifying the URL's on the client side**.
- Though this approach avoids overhead on the load balancer, **a drawback is that the amount of session information that can be stored is limited.** (Modifying each URL→ Shortening URL→ storing in engine→ degrades performance)
- **For applications that require larger amounts of session information, this approach does not work.**

(Changing a URL to the required format. URL rewriting allows URLs to be more easily remembered by the user. When the URL is entered into the Web server, the URL rewrite engine modifies the syntax behind the scenes to enable the appropriate Web page or database item to be retrieved. For example, to look up the definition for "path," a user friendly URL might look like computerlanguage.com/path. The rewrite engine could turn it into the following syntax: computerlanguage.com/results.php?definition=path. So, the URL rewrite function simply puts a layer on top of the original address and turns it into something easy to find and that makes sense. Thus, turning https://wiredelta.com/?page_id=16825 into wiredelta.com/url-rewrite, for example. From a user perspective, when a URL rewrite occurs the URL of the website remains the same in the browser and they are none the wiser. But behind the scenes, the browser rewrites the URL back into that complicated mess and sends a query to the servers.)

Cloud Concepts and Technologies

Load Balancing Techniques

- **Load balancing can be implemented in software or hardware.**
- **Software based load balancers run on standard operating systems** and like other cloud resources.
- **Hardware load balancers implement load balancing algorithms in Application Specific Integrated Circuits** (ASICs).

Cloud Concepts and Technologies

Scalability and Elasticity

- **Muti-tier applications such as e-Commerce**, **Social networking**, **business-to-business** etc. **can experience rapid changes in their traffic.**

- **Each website has a different traffic pattern** which is determined by a number of factors that are **generally hard to predict beforehand.**

- **Capacity planning involves determining the right sizing of each tier of the deployment of an application in terms of number of resources and capacity of each resource**.

- **Capacity planning may be for computing, storage, memory or network resources**.

- Traditional approaches for capacity planning are based on **predicted demands for applications and account for worst case peak load as of applications.**

Cloud Concepts and Technologies

Scalability and Elasticity

- **When the workloads of applications increase, the traditional approaches have been either to scale up or scale out.**

- **Scaling up involves upgrading the hardware resources** (adding computing, memory, storage or network resources. **Scaling out involves addition of more resources of the same type**.

- Traditional scaling up and scaling out approaches are based on **demand forecasts at regular intervals of time**.

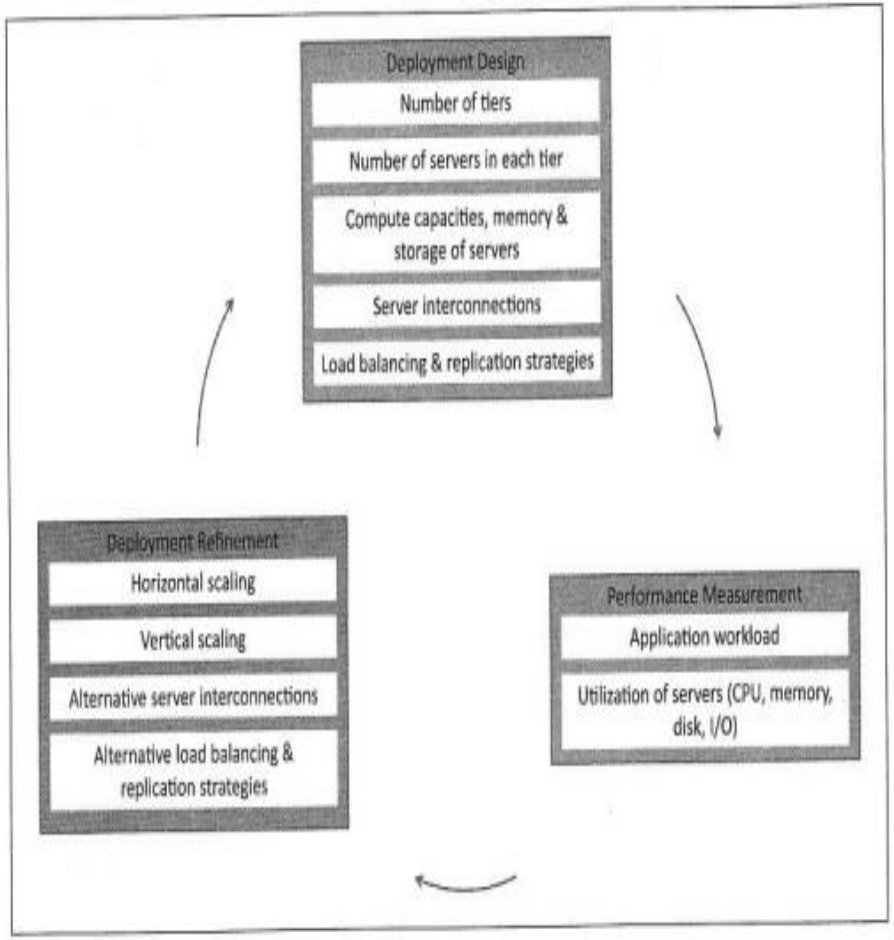
- When variations in workload are rapid, traditional approaches are **unable to keep track with the demand and either overprovisioning or under provisioning of resources**.

- **Over provisioning of resources of resources leads to higher capital expenditure** and **under provisioning of resources leads to traffic overloads, slow response time, low throughput.**

Cloud Concepts and Technologies

Deployment

- Figure shows the **cloud application deployment lifecycle**.
- Deployment prototyping can help in making architecture design choices.
- **By comparing performance of alternative deployment architectures**, deployment prototyping **can help in choosing the best and most effective deployment architecture that can meet the application performance requirements**.
- Deployment design is an iterative process that involves the following steps:
  - **Deployment Design**
  - **Performance Evaluation**
  - **Deployment Refinement**



Cloud application deployment lifecycle

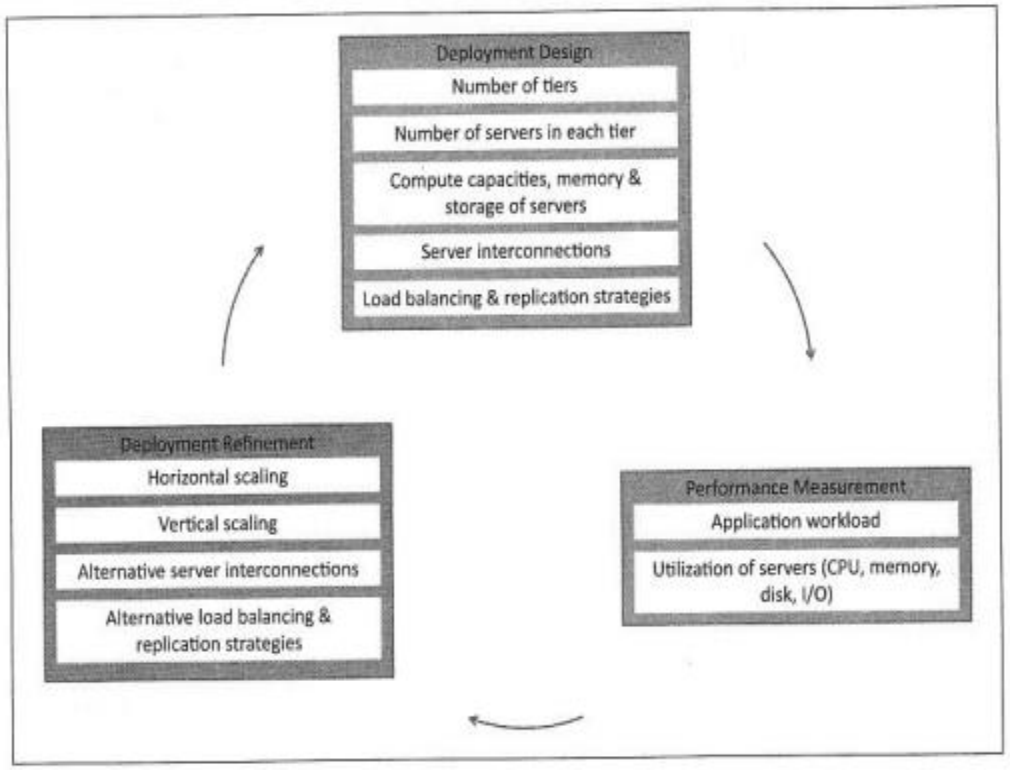
Cloud Concepts and Technologies
Deployment

1) Deployment Design

- In this step, the **application deployment is created with various tiers as specified in the deployment configuration**.

- The variables in this step include the **number of servers in each tier, computing, memory and storage capacities of servers, server interconnection, load balancing and replication strategies**.

- **Deployment is created by provisioning the cloud resources** as specified in the deployment configuration.



Cloud application deployment lifecycle

Cloud Concepts and Technologies

Deployment

2) Performance Evaluation

- **Once the application is deployed in the cloud, the next step in the deployment lifecycle is to verify whether the application meets the performance requirements with the deployment.**

- This step involves **monitoring the workload on the application and measuring various workload parameters such as response time and throughput**. In addition, **the utilization of servers (CPU, memory, disk I/O etc.)** is also monitored.

3) Deployment Refinement

- **After evaluating the performance of the application, deployments are refined so that the application can meet the performance requirements.**

- Various alternatives can exits for **deployment refinement such as vertical scaling, horizontal scaling, alternative server interconnections, alternative load balancing and replication strategies**.

Cloud Concepts and Technologies

Deployment

- The below table lists some popular cloud deployment management tools.

Cloud Deployment Management Tool	Features
RightScale	Design, deploy and manage cloud deployments across multiple public or private clouds.
Scalr	Provides tools to automate the management of servers, monitors servers, replaces servers that fail, provides auto scaling and backups.
Kaavo	Allows deploying applications easily across multiple clouds, managing distributed applications and automating high availability.
CloudStack	Allows simple and cost effective deployment management and configuration of cloud computing environments.

Examples of popular cloud deployment management tools

Cloud Concepts and Technologies

Replication

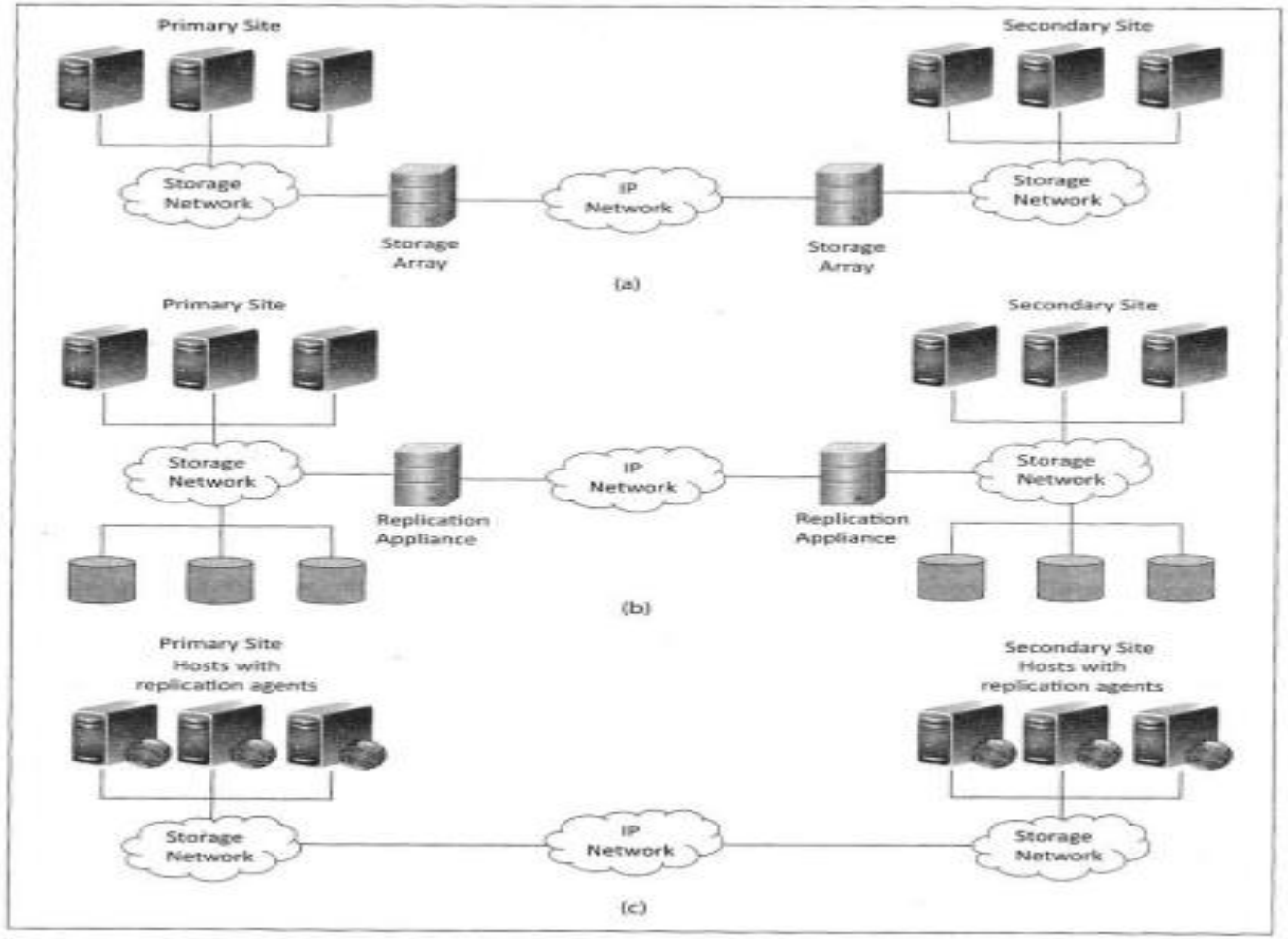
- Replication is used to **create and maintain multiple copies of the data in the cloud**.
- Replication of data is important for practical reasons **such as business continuity and disaster recovery**.
- **In the event of data loss at the primary location, organizations can continue to operate their applications from secondary data sources.**
- Traditional business time objective (RTO).
- continuity and disaster recovery approaches don’t provide efficient, cost effective and automated recovery of data.
- **Cloud based data replication approaches provide replication of data in multiple locations, automated recovery, low recovery point objective (RPO)and low recovery time**.
- With **cloud based data replication**, **organizations can plan for disaster recovery without making any capital expenditures on purchasing, configuring and managing secondary site locations**.

Cloud Concepts and Technologies

Replication

- **There are three types of replication approaches as shown in below figure**

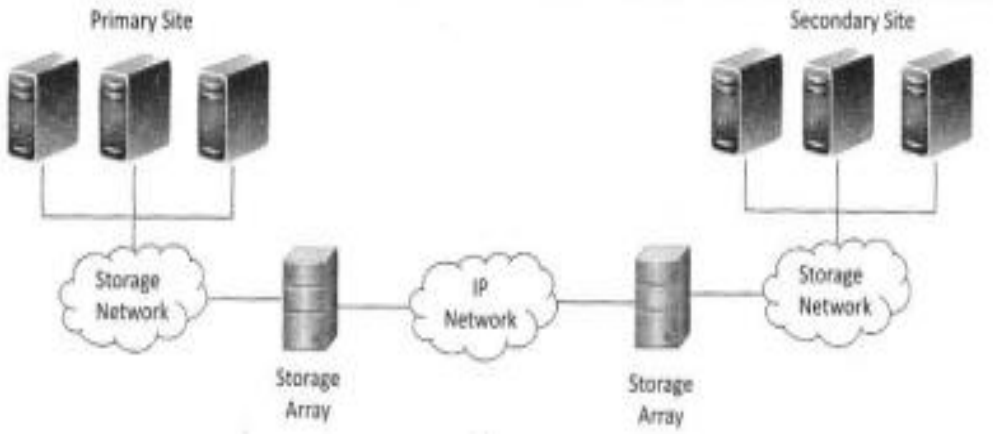
  - **Array Based Replication**
  - **Network Based Replication**
  - **Host Based Replication**



Replication approaches: (a) Array-based replication, (b) Network-based replication, (c) Host-based replication

Cloud Concepts and Technologies

Replication



Array Based Replication

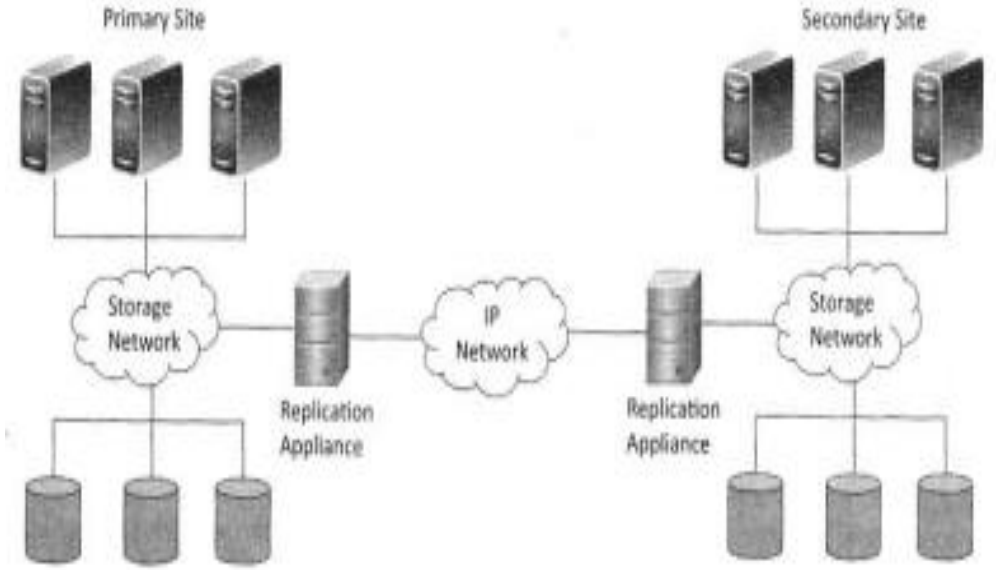
- Array Based Replication uses **compatible storage arrays to automatically copy data from a local storage array to a remote storage array**.
- **Arrays replicate data at the disk sub system level**(in terms of blocks), **therefore the type of hosts accessing the data and the type of data is not important**. **Thus array based replications can work in heterogeneous environments with different operating systems.**
- Array based replication uses **Network Attached Storage (NAS) or Storage Area Network (SAN)** to replicate.
- **A drawback of this array based replication is that it requires similar arrays at local and remote locations.** (Storage methods, access methods, block size, block number, arrays etc must be of same). **Thus the costs for setting up array based replications are higher than the other approaches.**
- (Array-based replication is **an approach to data backup in which compatible storage arrays use built-in software to automatically copy data from one storage array to another**.)

Cloud Concepts and Technologies

Replication

Network Based Replication

- **Network based replication uses an appliance or device that sits on the network** and **intercepts packets that are sent from hosts and storage arrays**. **The intercepted packets are replicated to a secondary locations.**



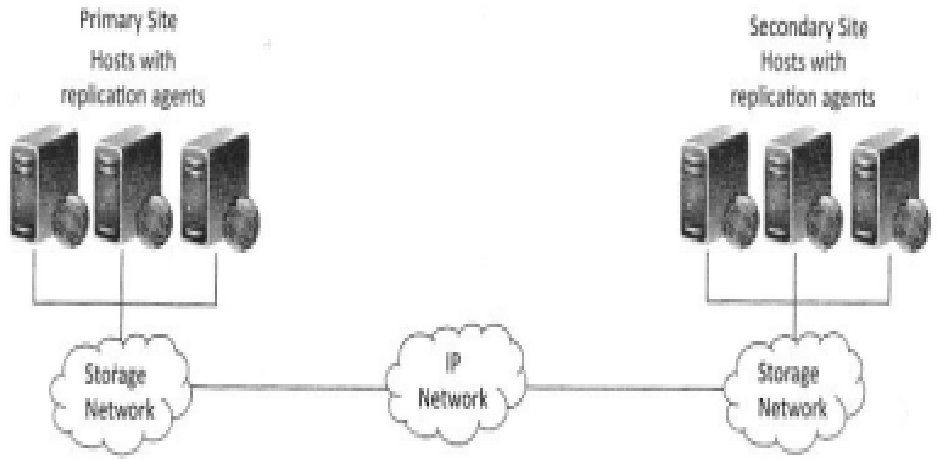
- **By offloading replication from server and storage systems, network-based replication can work across a large number of server platforms and storage systems, making it ideal for highly heterogeneous environments and requires a single point of management.**

- **This approach involves higher initial costs due to replication hardware and software**.

Cloud Concepts and Technologies

Replication

Host Based Replication

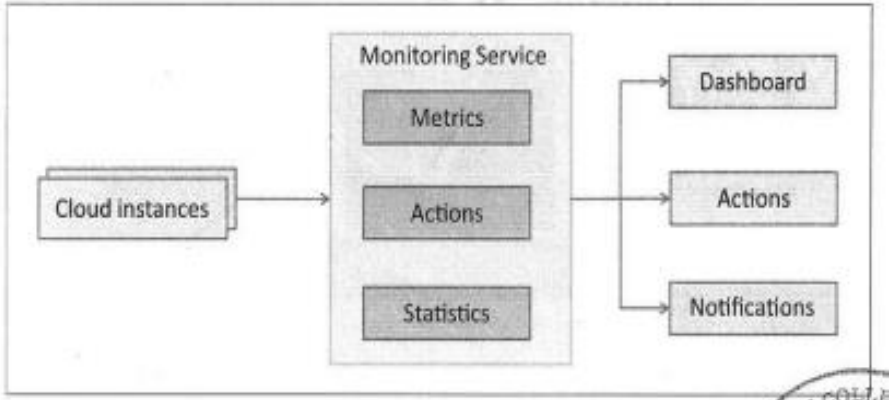


- **Host based replication runs on standard servers and uses software to transfer data from a local to remote location.**
- **An agent is installed on the hosts that communicates with the agents on the other hosts**.
- **Host based replication can either be block based or file based**. Block based replication typically require dedicated volumes of the same size on both the local and remote servers.
- **With host based replication, entire virtual machines can be replicated in real time**.

Cloud Concepts and Technologies

Monitoring

- **Cloud providers provides monitoring service that allows cloud users can monitor their cloud resource usage.**
- A monitoring service at the cloud **collects data on various system and application metrics from cloud computing instances.**



Typical cloud monitoring service architecture

- **Users can define various actions based on the monitoring data**. For eg: **Auto scaling when the CPU usage of the monitored resources becomes high.**
- **Monitoring services also provides various statistics** based on the monitoring data collected as shown in below table.

Type	Metrics
CPU	CPU-Usage, CPU-Idle
Disk	Disk-Usage, Bytes/sec (read/write), Operations/sec
Memory	Memory-Used, Memory-Free, Page-Cache
Interface	Packets/sec (incoming/outgoing), Octets/sec(incoming/outgoing)

Typical monitoring metrics

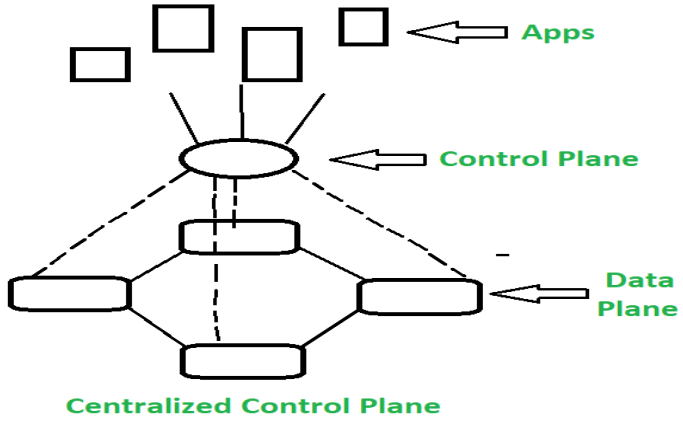
- **Monitoring of cloud resources is important because it allows the users to keep track of the health of applications and services deployed in the cloud.** For eg: an organization which has its website hosted in the cloud can monitor the performance of the website and also the website traffic with which users can make operational decisions such as scaling up or scaling down cloud resources.

Cloud Concepts and Technologies

Software Defined Networking

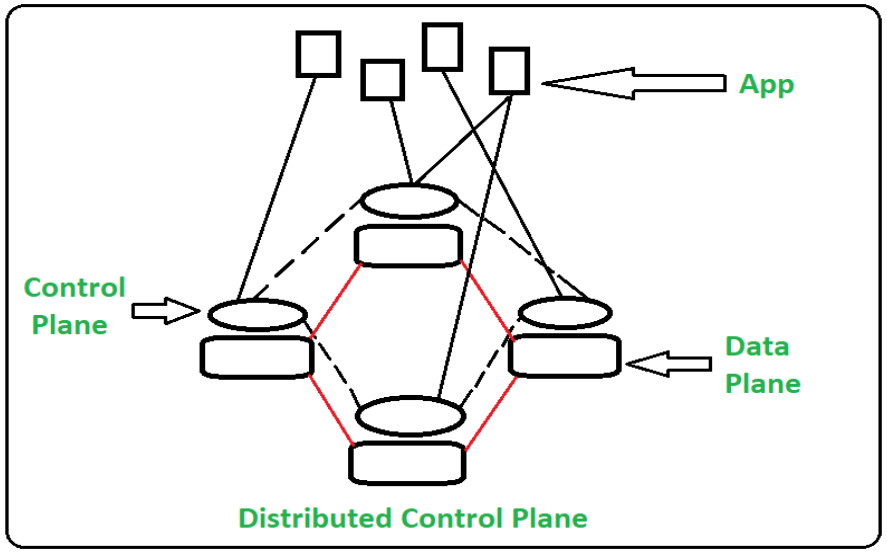
Software Defined Networking

- **SDN decouples the network and control forwarding functions.**
- SDN stands for Software Defined Network which is networking architecture approach. It enables the control and management of network using software applications.
- Through Software Defined Network (SDN) networking **behavior of entire network and its devices are programmed in centrally controlled manner through software applications** using open APIs.



Traditional Network

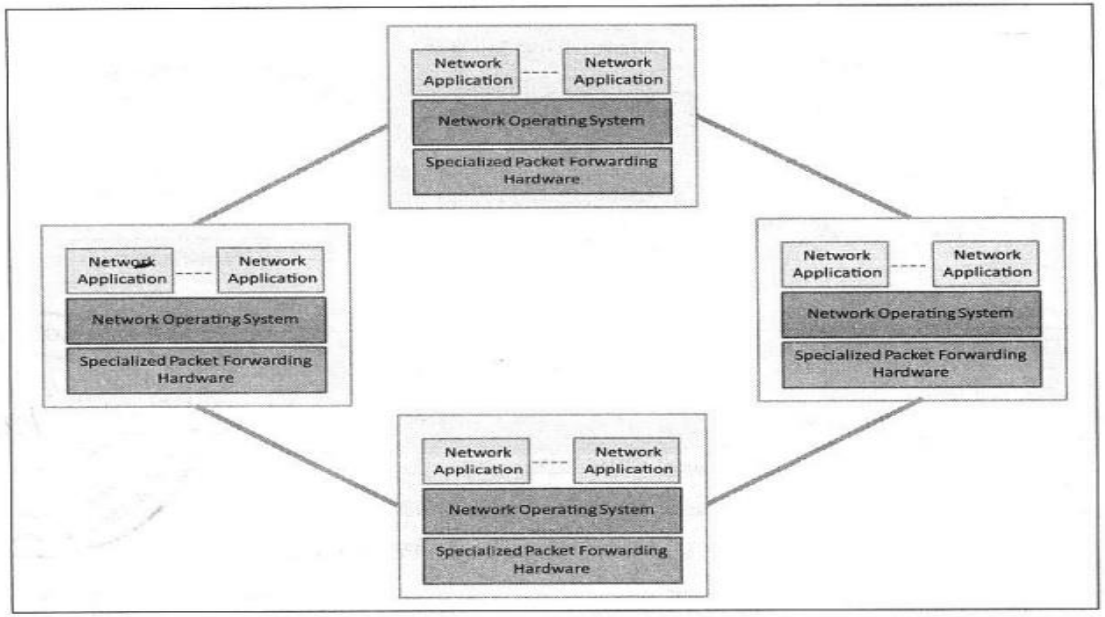
- **Traditional network refers to the old conventional way of networking which uses fixed and dedicated hardware devices such as routers and switches to control network traffic.**
- **Inability to scale and network security and performance are the major concern** now a days in the current growing business situation so that SDN is taking control to traditional network.
- **Traditional network is static** and based on hardware network appliances



Cloud Concepts and Technologies

Software Defined Networking

- **SDN decouples the network and control forwarding functions.** It separates the **control plane** (making traffic decision) from the **data plane** (packet forwarding).
- The below figure shows the conventional network architecture built with specialized hardware (Switches, routers etc).

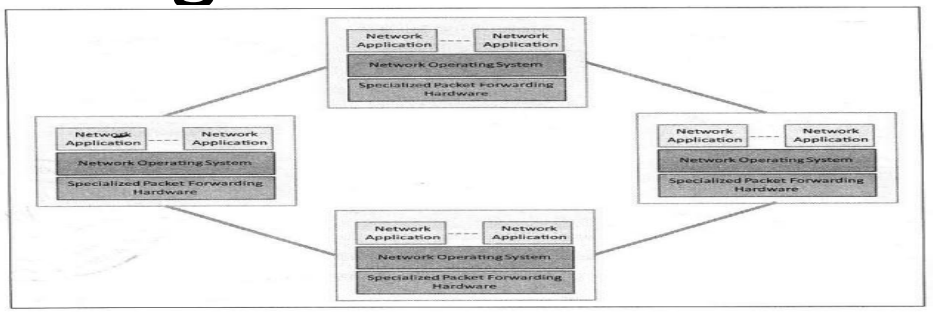


Conventional network architecture

- **In the conventional network architecture, the control plane and data plane are coupled**. **Control plane is the part of the network that carries the signaling and routing message traffic** while the **data plane is the part of the network that carries the payload data traffic.**

Cloud Concepts and Technologies

Software Defined Networking



Conventional network architecture

- **Conventional network architecture has following limitations**

➢ Complex Network Devices

Conventional networks are getting increasingly complex with more and more protocols being implemented to improve link speed and reliability. **The conventional networks are well suited for static traffic patterns**. **Due to the complexity of conventional network devices, making changes in the networks to meet the dynamic traffic patterns has increasingly difficult**.

➢ Management Overhead

Network mangers find it increasingly difficult to manage multiple network devices. **Upgradation of network requires configurations changes in multiple devices** (**switches, routers, firewalls** etc.)

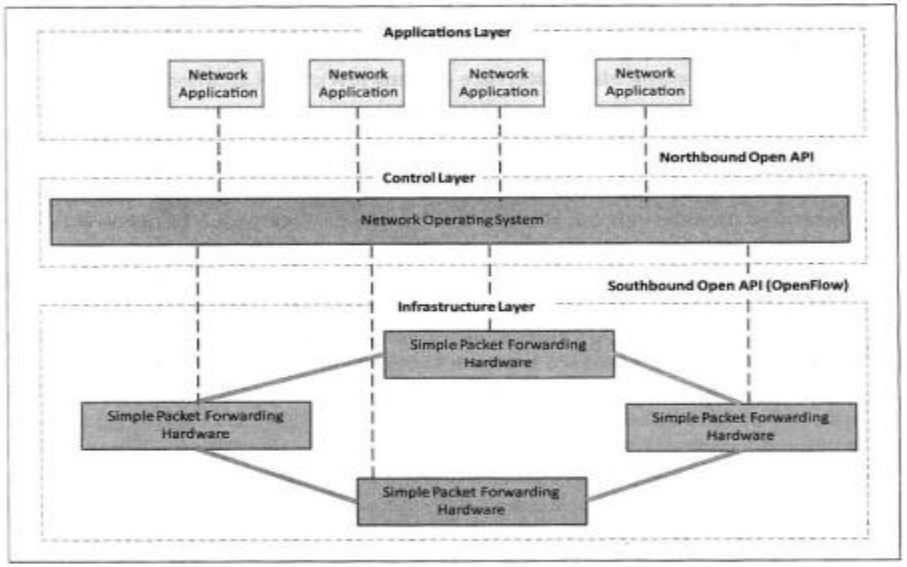
➢ Limited scalability

Big data applications run distributed algorithms on a large number of virtual machines. Such computing environments require highly scalable with minimum manual configurations which is difficult with conventional networks. (running each module of the applications at different places with multiple copies of data at different locations, need to scaling of the network)

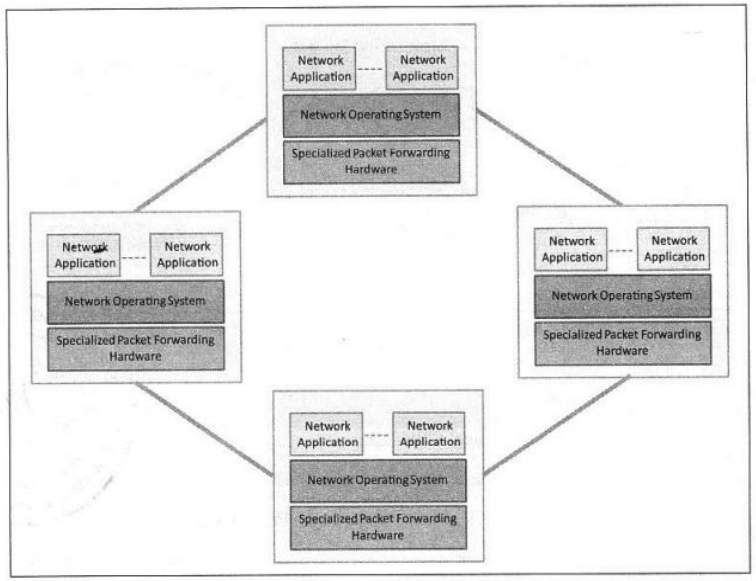
Cloud Concepts and Technologies

Software Defined Networking

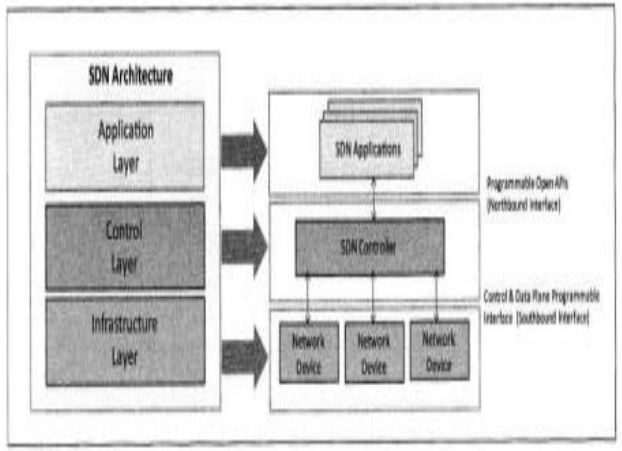
- **SDN attempts to create network architecture that are simpler, inexpensive, scalable, agile and easy to manage.**
- In SDN architecture, **the control plane and data planes are decoupled and network controller is centralized.**
- The **underlying infrastructure in SDN uses simple packet forwarding hardware as opposed to specialized hardware in conventional networks.**
- **Network devices become simple with SDN as they do not require implementation of large number of protocols.**
- **Network devices receive instructions from the SDN controller on how to forward the packets.**



SDN architecture



Conventional network architecture



SDN layers

Cloud Concepts and Technologies

Network Virtualization Function

Cloud Concepts and Technologies

Map Reduce

- **Map Reduce is a parallel data processing technique for processing and analyzing large scale data**. (data intensive problem). (The framework takes care of scheduling tasks, monitoring them and re-executing any failed tasks.)

- **The Map Reduce model includes two important phases: Map and Reduce functions.**

- **Map function takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).** Secondly, **reduce function, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples**.

MAPREDUCE PROGRAMMING MODEL: Example taken from net for your reference

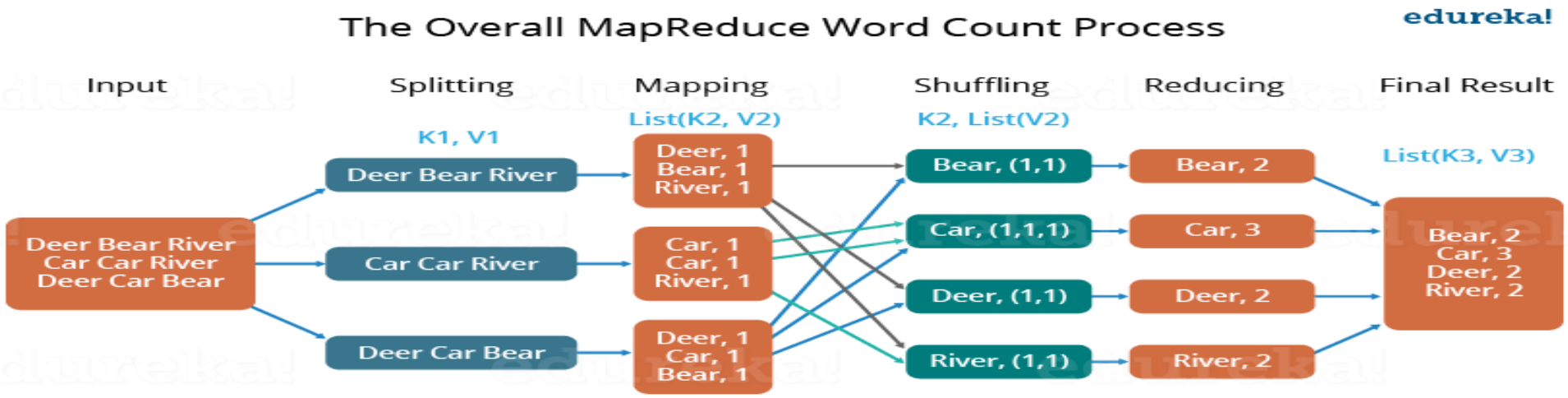
A Word Count Example of MapReduce

Let us understand, how a MapReduce works by taking an example where there is a text file called **example.txt** whose contents are as follows:

Dear, Bear, River, Car, Car, River, Deer, Car and Bear

Now, suppose, we have to perform a word count on the **example.txt** using MapReduce to find the unique words and the number of occurrences of those unique words.

Example taken from net for your reference



- First, we divide the input in three splits as shown in the figure. This will distribute the work among all the map nodes. (Divides into file into 3 subfiles and each file given to each Map node.)

- Then, Map function within each Map node tokenize the words in the file and give a hardcoded value (1) to each of the tokens or words.

- Now, a list of key-value pair will be created where the key is nothing but the individual words and value is one. So, for the first line **(Dear Bear River)** we have 3 key-value pairs – **(Dear, 1); (Bear, 1); (River, 1)**. The mapping process remains the same on all the nodes.

- After mapper phase, a partition process takes place where sorting and shuffling happens so that all the tuples with the same key are sent to the corresponding reducer.

- So, after the sorting and shuffling phase, each reducer will have a unique key and a list of values corresponding to that key. For example, Bear, [1,1]; Car, [1,1,1].., etc. (Input to Reduce Node)

- Now, each Reducer counts the values which are present in that list of values. As shown in the figure, reducer gets a list of values which is [1,1] for the key Bear. Then, it counts the number of ones in the list and gives the final output as – Bear, 2.

- Finally, all the output key/value pairs are then collected and written in the output file.

Cloud Concepts and Technologies

Map Reduce (for your reference)

- Among the cluster of nodes, one acts like a "**master**" and the rest are "**workers**."
- The **master is responsible for scheduling** (assigns the map and reduce tasks to the worker) and **monitoring** (monitors the task progress and the worker health). (Master instructs which nodes should act like a mapper and which nodes should act like a reducer. Also responsible for monitoring health condition of each worker node)
- When map tasks arise, the **master assigns** the task to an **idle worker**, taking into account the data locality.
- A **worker reads** the **content of the corresponding input split** and emits a **key/value pairs**. (On what operation performing and corresponding analysed value)
- The **intermediate key/value pairs** produced by the Map function are first **buffered in memory and then periodically written to a local disk**, **sorts** the intermediate keys so that **all occurrences of the same key are grouped together into R sets by the partitioning function.**
- The master passes the location of these stored pairs to the **reduce worker**, which **reads** the buffered data using remote procedure calls (**RPC**).
- For each key, the worker passes the corresponding intermediate value for its entire occurrence to the **Reduce function**. (worker node gives intermediate set to reduce function, which generates output file)
- Finally, the output is available in **R output files** (one per reduce task).

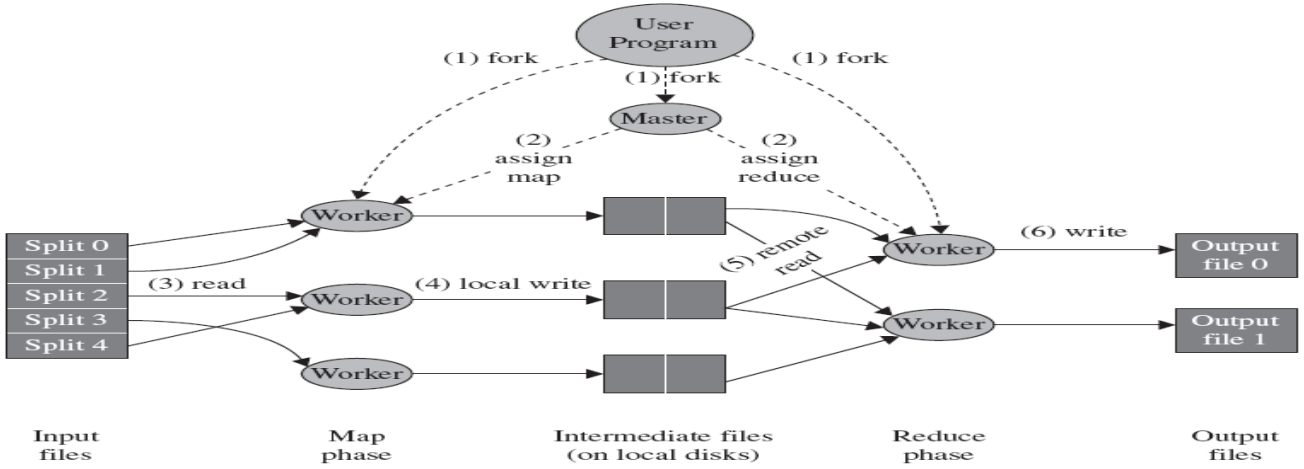


FIGURE 14.3. MapReduce execution overview [4].

Cloud Concepts and Technologies

Map Reduce

- **The below figure shows the workflow of MapReduce.**
- **In the Map phase, data is read from a distributed file system, partitioned among a set of computing nodes.**
- The **Map tasks process the input records independently of each other** and **produce intermediate results as Key-Value pairs.**
- **When all the Map tasks are completed**, the **Reduce phase begins** in which the **intermediate data with the same key is aggregated.**
- An **optional combine task can be used to perform data aggregation** on the intermediate data of the same key **for the output from the Mapper before transferring to the Reduce task.**

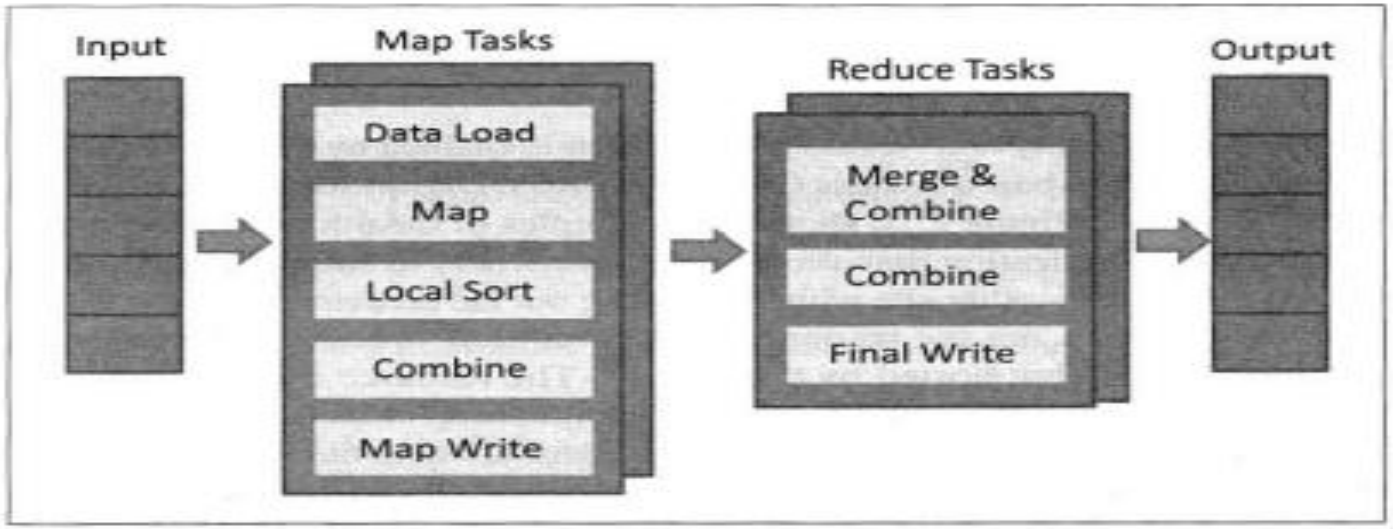


Figure : MapReduce workflow

Cloud Concepts and Technologies

Map Reduce

- **The below figure shows the flow of data for a MapReduce** job.

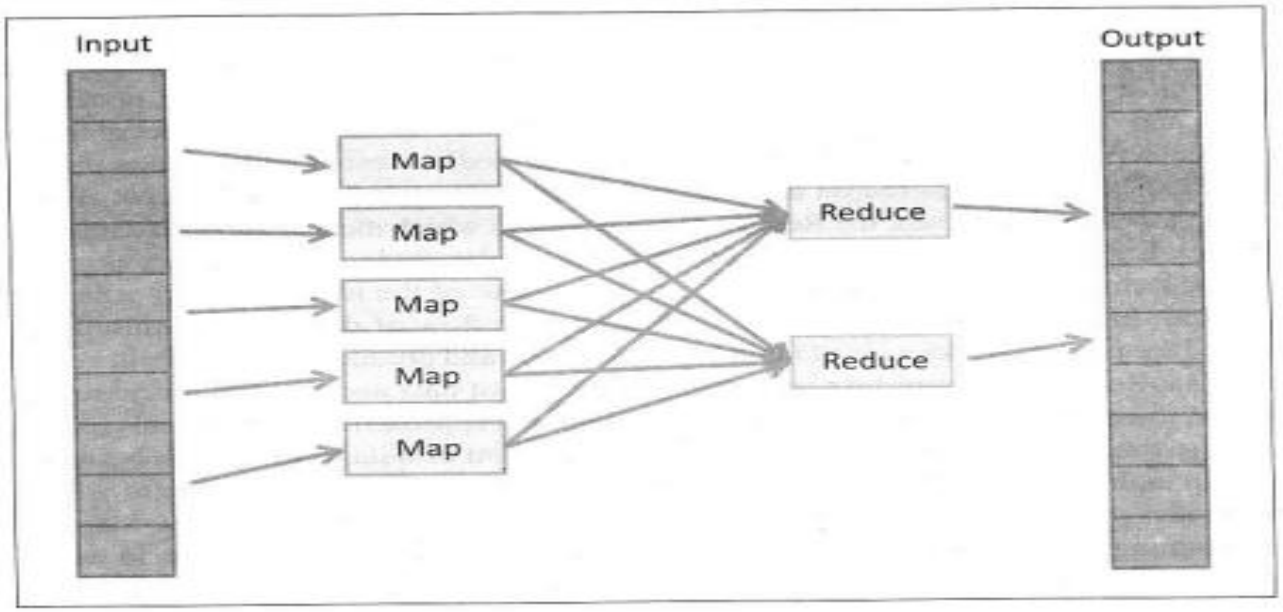


Figure Data flow in MapReduce

- **MapReduce model take advantage of locality of data** and the data processing takes place on the nodes where the data resides.
- **In traditional approach for data analysis, data is moved to the compute nodes** which results in significant of **data transmission between the nodes** whereas **MapReduce model moves the computation to where the data resides thus decreasing the transmission of data and improving efficiency**.

Cloud Concepts and Technologies

Identity and Access Management

- **Identity and Access Management (IDAM)** for cloud describes the **authentication and authorization of users to provide secure access to cloud resources.**

- Organization with multiple users can use IDAM services provided by the cloud service provider **for management of user identifiers and user permissions.**

- **IDAM services allow organizations to centrally manage users, access permission, security credentials and access keys.**

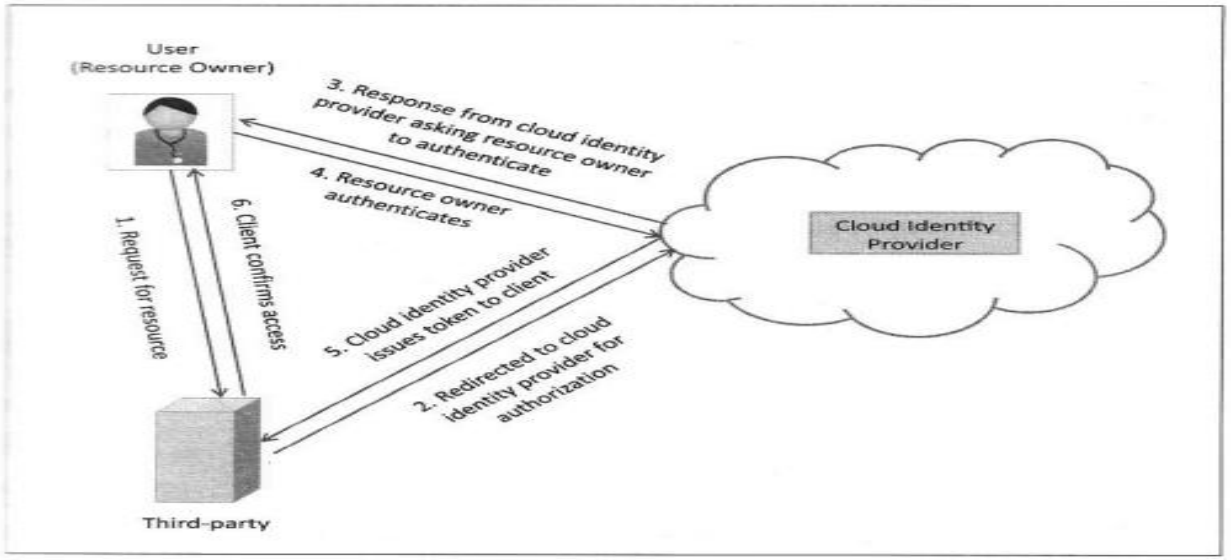
- IDAM services **allow creation of user groups where all the users in a group have the same access permissions.**

- Identity and Management is enabled by a number of technologies such as **OpenAuth, Role-based Access Control(RBAC), Digital Identities, Security Tokens, Identity Providers etc.**

Cloud Concepts and Technologies

Identity and access Management

- **The below figure shows the examples of OAuth identity and Access Management Service provided by the cloud service providers.**



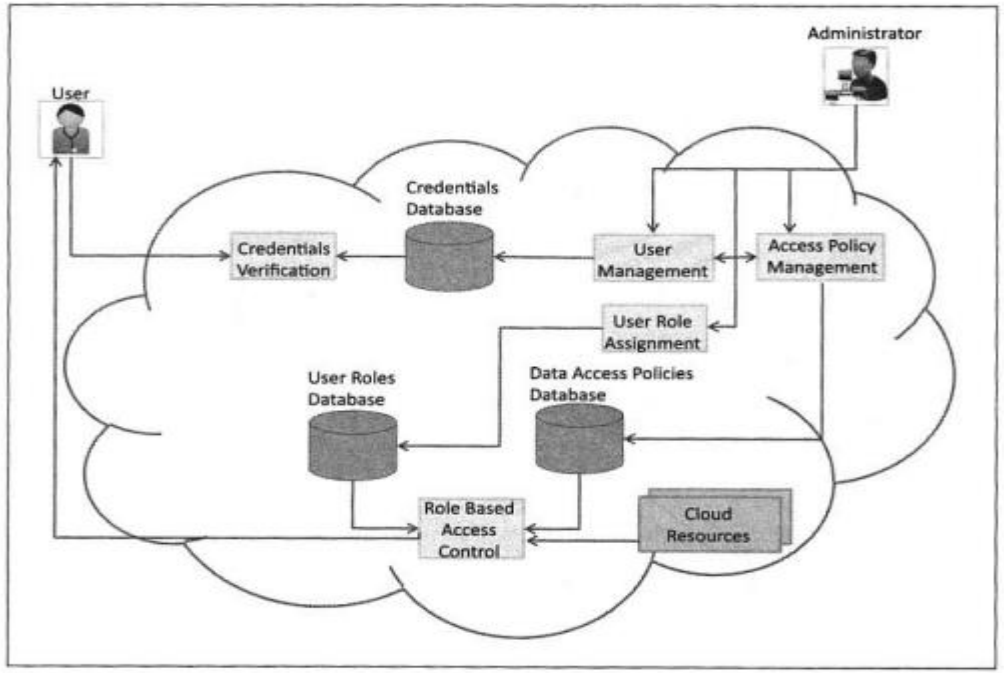
OAuth example

- **Oauth is an open standard for authorization that allows resource owners to share their private resources stored on one site with an other site without handling out the credentials.**
- In the Oauth model, **an application** (which is not the resource owner) **requests access to resources controlled by the resource owner** (but hosted by the server), **the resource owner grants permission to access the resources in the form of a token and matched shared-secret.**
- **Tokens can be issued with a restricted scope and limited lifetime and revoked independently.**

Cloud Concepts and Technologies

Identity and access Management

- **RBAC is an approach for restricting access to authorized users.**
- **The below figure shows an example of a typical RBAC framework.**
- **A user who wants to access cloud resources is required to send his/her data to the system administrator who assigns permissions and access control policies which are stored in the User Roles and Data Access Policies databases respectively.**



Role-based Access Control example

Cloud Concepts and Technologies

Service Level Agreement

- **A Service Level Agreement (SLA)** for cloud **specifies the level of service that is formally defined as a part of the service contract with the cloud service provider.**

- **SLAs provide a level of service for each service** which is **specified in the form of minimum level of service guaranteed and a target level.**

- **SLAs contain a number of performance metrics and the corresponding service objectives.**

- **The below table lis**

Criteria	Details
Availability	Percentage of time the service is guaranteed to be available
Performance	Response time, Throughput
Disaster Recovery	Mean time to recover
Problem resolution	Process to identify problems, support options, resolution expectations
Security and privacy of data	Mechanisms for security of data in storage and transmission

Table : List of criteria for cloud SLAs

Cloud Concepts and Technologies

Billing

- **Cloud Service providers offer a number of billing models described as follows:**
  - **Elastic Pricing**
  - **Fixed Pricing**
  - **Spot Pricing**

Elastic Pricing

- **In Elastic pricing** or **Pay-as-you-use pricing model,** the **customers are charged based on the usage of cloud resources.**
- **Cloud computing provides the benefit of provision resources on demand.**
- **On-demand provisioning and elastic pricing models bring cost savings for the customers.**
- Elastic pricing model is **suited for customers who consume cloud resources for the short durations and who cannot predict the usage beforehand.**

Cloud Concepts and Technologies

Billing

Fixed Pricing

- **In fixed pricing models, customers are charged a fixed amount per month for the cloud resources.**
- For eg: Fixed amount can be **charged per month for the running a virtual machine instance, irrespective of the actual usage.**
- Fixed pricing model is **suited for customers who want to use cloud resources for longer durations and want more control over the cloud expenses.**

Spot Pricing

- **Spot pricing models offer variables pricing for the cloud resources which is driven by market demand.**
- **When the demand for the cloud resources is high, the prices increase** and **when the demand is lower, the prices decrease.**
- **The below table lists the billable resources for cloud including virtual machines, network., storage, data services, security services, support, application services, deployment and management services.**

Resource	Details
Virtual machines	CPU, memory, storage, disk I/O, network I/O
Network	Network I/O, load balancers, DNS, firewall, VPN
Storage	Cloud storage, storage volumes, storage gateway
Data services	Data import/export services, data encryption, data compression, data backup, data redundancy, content delivery
Security services	Identity and access management, isolation, compliance
Support	Level of support, SLA, fault tolerance
Application services	Queuing service, notification service, workflow service, payment service
Deployment and management services	Monitoring service, deployment service

Table : List of billable resources for cloud

Cloud Based Services and Applications

- **Some examples on cloud based services and applications are:**

Cloud Computing for Healthcare

- The figure shows the application of cloud computing environments to the healthcare ecosystem.
- Hospitals and their affiliated providers can securely access patient data stored in the cloud and share the data with the other hospitals and physicians.
- Patients can access their own health information from all of their care providers and store it in a personal health record (PHR)

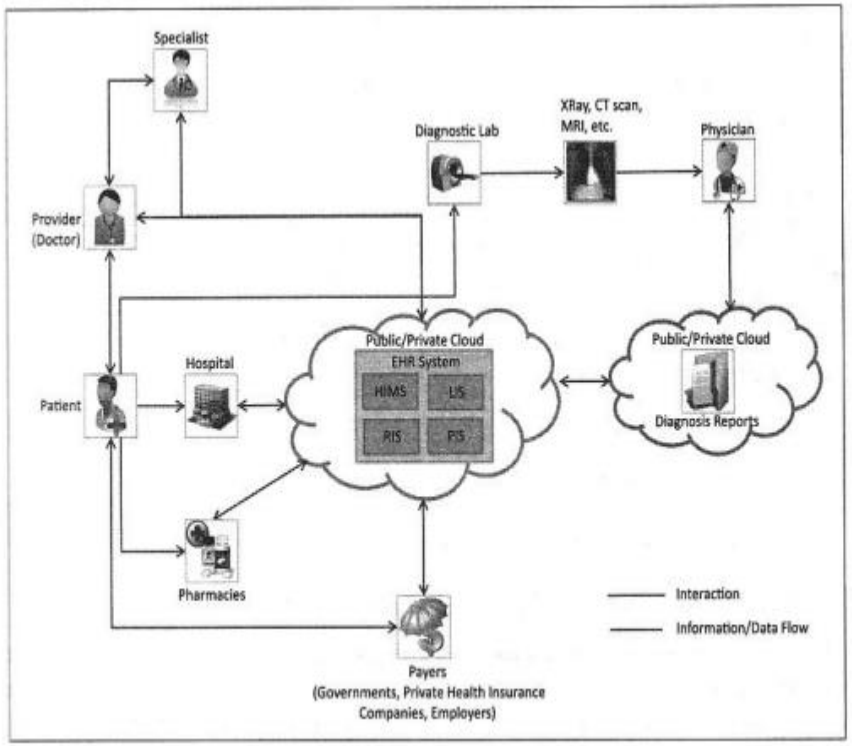


Figure : Cloud computing for healthcare

- The PHR can be a vehicle for e-prescribing, a technique known to
- reduce medication dispensing errors and to facilitate medication reconciliation.
- History and information stored in the cloud can be streamline the admissions, care and discharege process by eliminating redundant data collection and entry.
- Health payers can increase the effectiveness and lower the cost of their care management programs by providing value added services and giving access to health information to members.

Cloud Based Services and Applications

Cloud Computing for Energy Systems

- Energy systems such as Smart Grids, Power Plants, Wind Turbine etc. have thousands of sensors that gather real time maintenance data continuously for condition monitoring and failure prediction purposes.
- The energy systems have a large number of critical components that must function correctly so that the systems can perform their operations correctly.
- For eg: a wind turbine has number of critical components like bearings, turning gears etc. that must be monitored carefully as wear and rear in such critical components or sudden change in operating conditions of the machines can result in failures. In systems such as power grids, real time information is collected using specialized electrical sensors called Phasor Measurement Units (PMU) at the sub-stations. The information received from PMUs must be monitored in real time for estimating the state of the system and for predicting failures.
- There is a generic framework “CloudView” for storage, processing and analysis of massive machine maintenance data collected from a large number of sensors embedded in industrial machines in a cloud computing environment.
- The below figure shows a generic use case of cloud for energy systems.

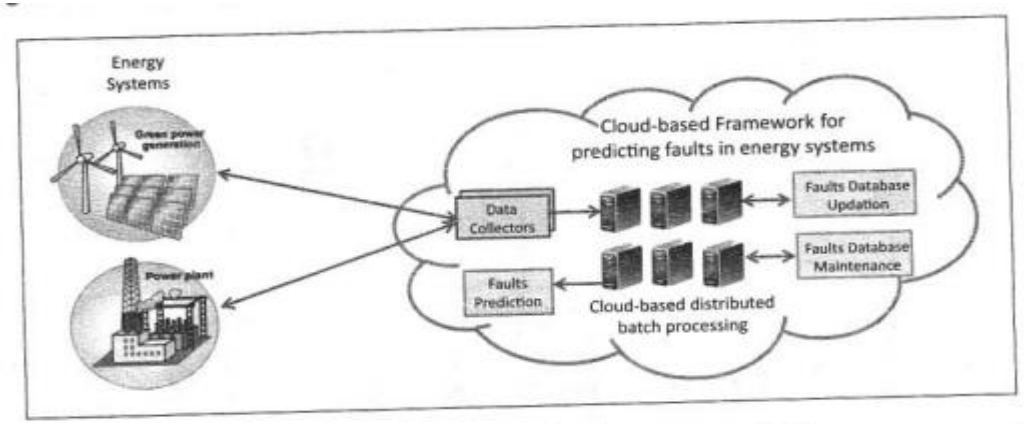


Figure : Cloud computing for energy systems

Cloud Based Services and Applications

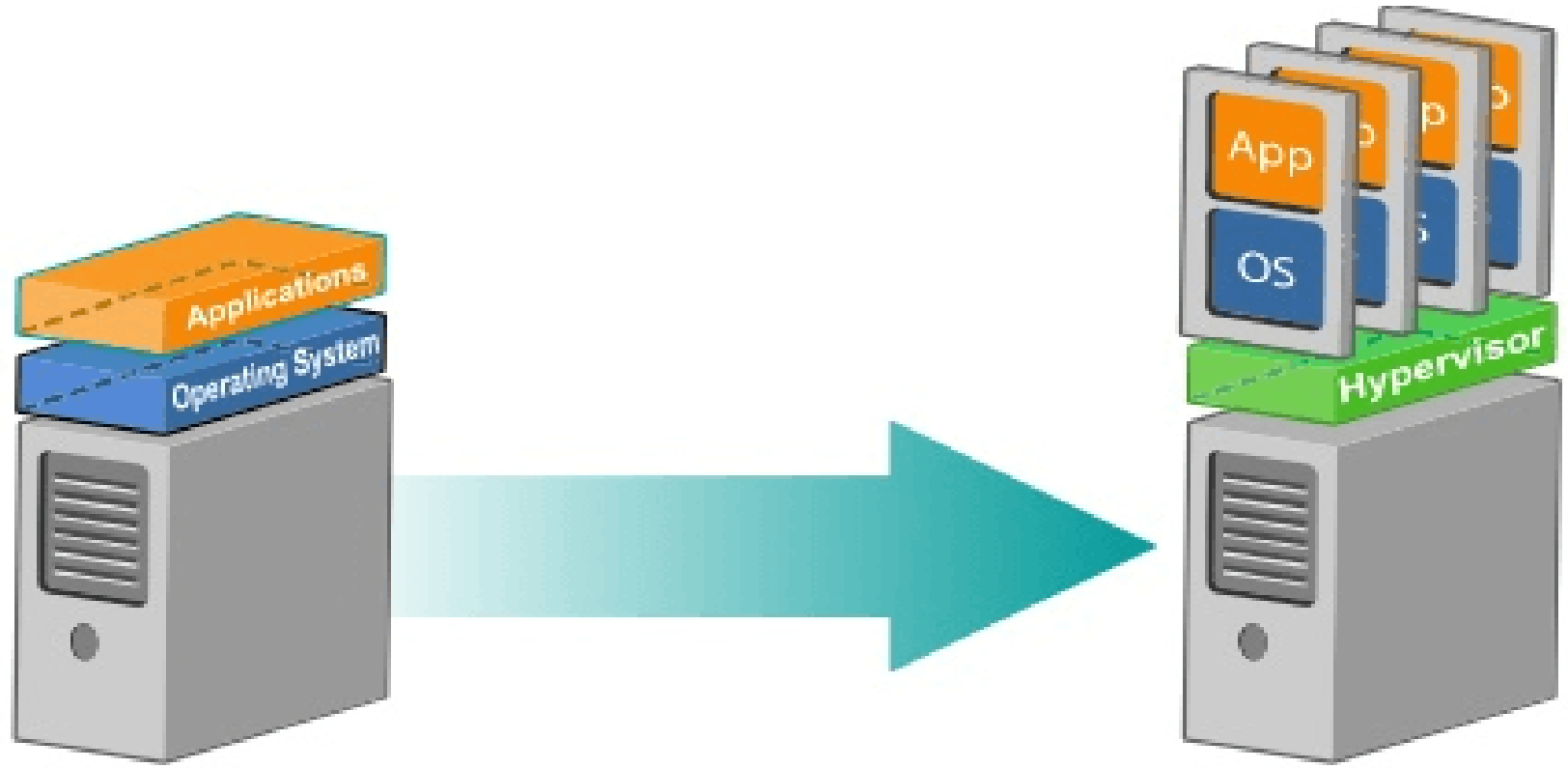
- **Cloud Computing for Transportation System**

Unit-2
VIRTUALIZATION FOR CLOUD

Virtualization for Cloud

- **Virtualization is a technology** that makes it possible to **run multiple applications and various operating systems on the same server at the same time**.

- It increases **hardware utilization, saves energy and costs**.



NEED FOR VIRTUALIZATION

- **Virtualization provides various benefits** including **saving time** and **energy**, **decreasing costs** and **minimizing overall risk.** (Saves Time: instead of maintaining 3 or 4 servers, maintain single server, configuring single server, saves power energy, costs, managing single server→ reduces overall risks)

  - **Provides ability to manage resources effectively**. (Admin can concentrate on managing few number of servers)

  - **Increases productivity, as it provides secure remote access**. (More number of users can access their own VM from a single server in more secured way. Complete VM is given, can access within organization, while roaming, or at home in secured way)

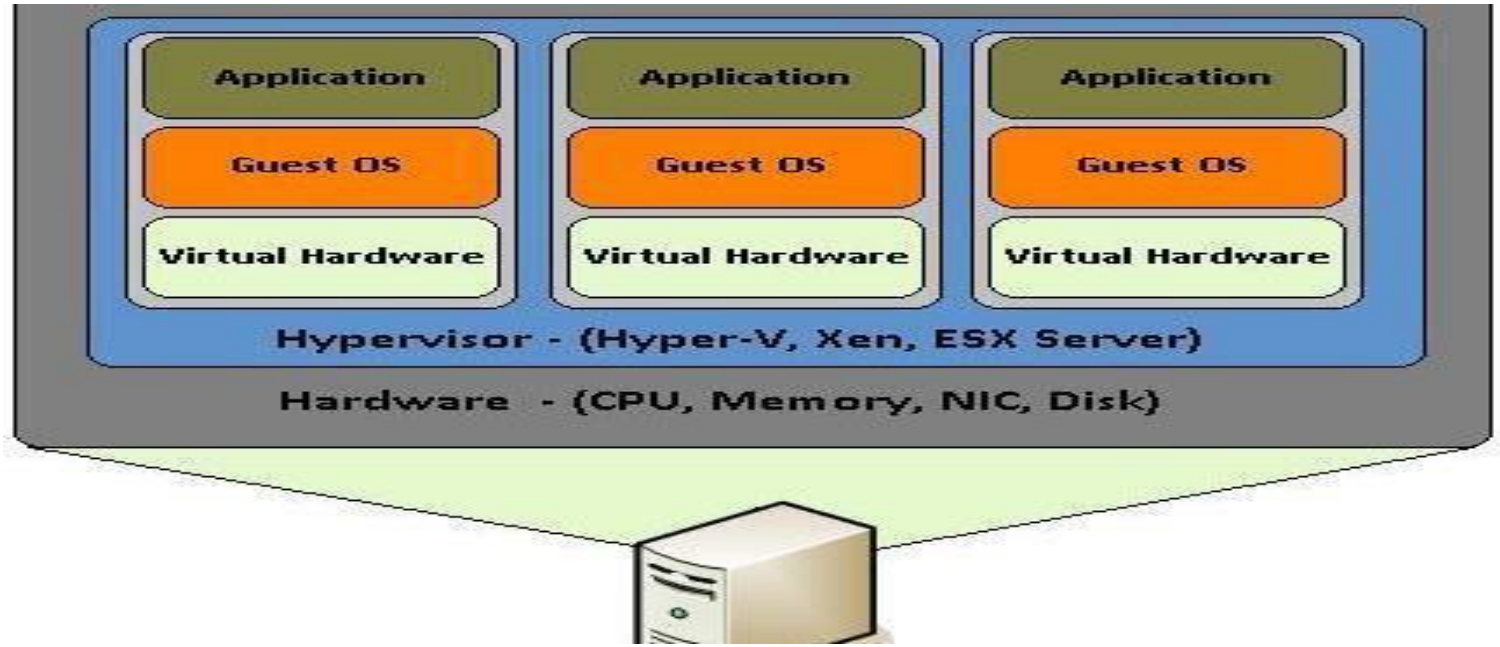
  - **Provides for data loss prevention**. (No data leakage between the users, Complete control on individual VM and complete VM is given in a secured way)

NEED FOR VIRTUALIZATION

What makes virtualization possible?

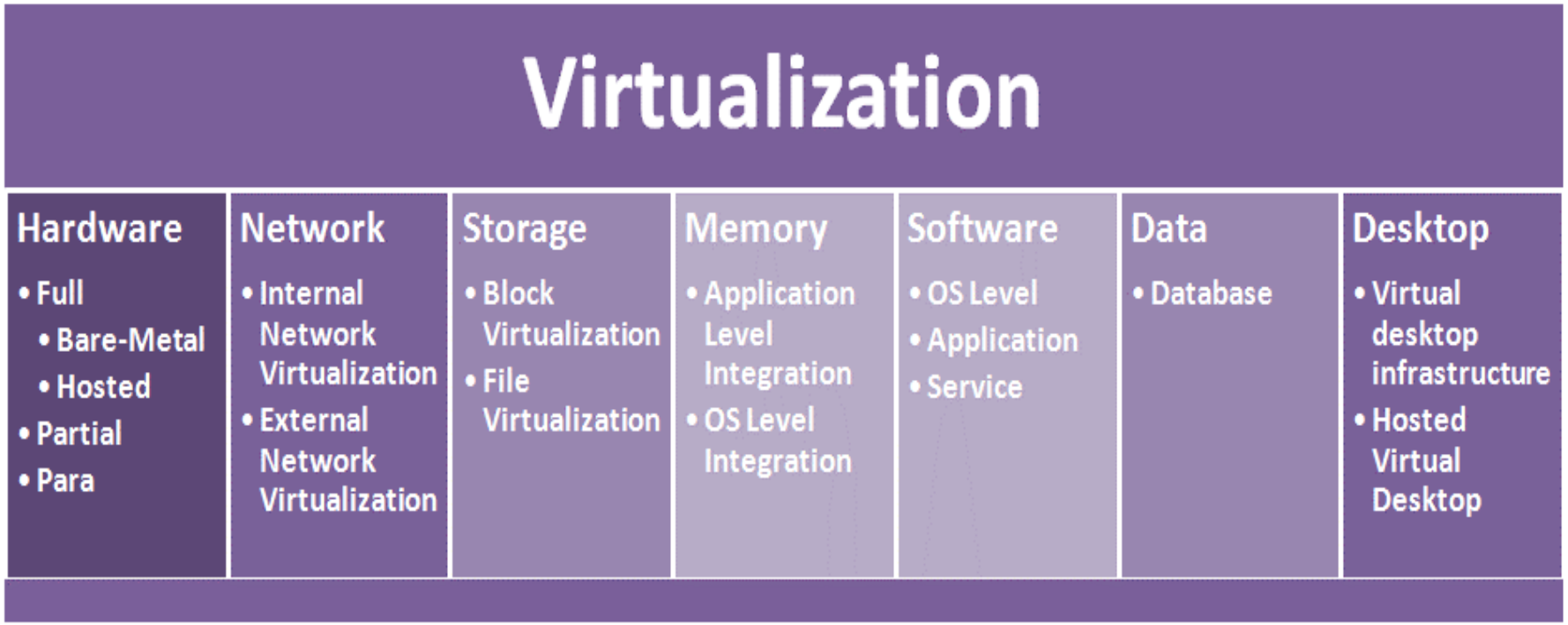
- The software that makes virtualization possible is known as a **Hypervisor**, also known as a **Virtualization Manager**.

- **Hypervisor** sits between the **hardware and the operating system**, and **assigns the amount of access that the applications** and **operating systems have with the processor** and **other hardware resources**. (Hypervisor assigns the CPU and other hardware resources needed to run OS & app)



TYPES OF VIRTUALIZATION

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



TYPES OF VIRTUALIZATION

Hardware/Server Virtualization

- **Server virtualization is a virtualization technique** that involves **partitioning a physical server into a number of small, virtual servers with the help of virtualization software**, so that the processor can be used more effectively and efficiently .

- The **hypervisor** controls the processor, memory, and other components by allowing different OS to run on the same machine. (This hypervisor s/w is responsible for controlling and allocating required percentage of CPU, memory and other hardware resources to individual VM)

- Hardware virtualization is further subdivided into the following types:

➢ **Full Virtualization**

➢ **Para Virtualization**

➢ **Partial Virtualization**

TYPES OF VIRTUALIZATION

Full Virtualization

- **In Full Virtualization**, the complete simulation of the actual hardware takes place. That is, the **hypervisor provides complete abstraction**. (The h/w resources is completely emulated to each individual VM's. for eg: VM1,VM2,VM3 runs as individual machines)

- **Each guest operating system is unaware that it is in a virtualized environment**, and therefore hardware is virtualized by the host operating system so that the **guest can issue commands to what it thinks is actual hardware**, **but really are just simulated hardware devices created by the host**. (guest OS is not knowing that it is running on virtualized environment. It just issue command for read or write operation to memory as like original OS, which is handled by host OS)

- **Guest operating systems are unaware of each other.**

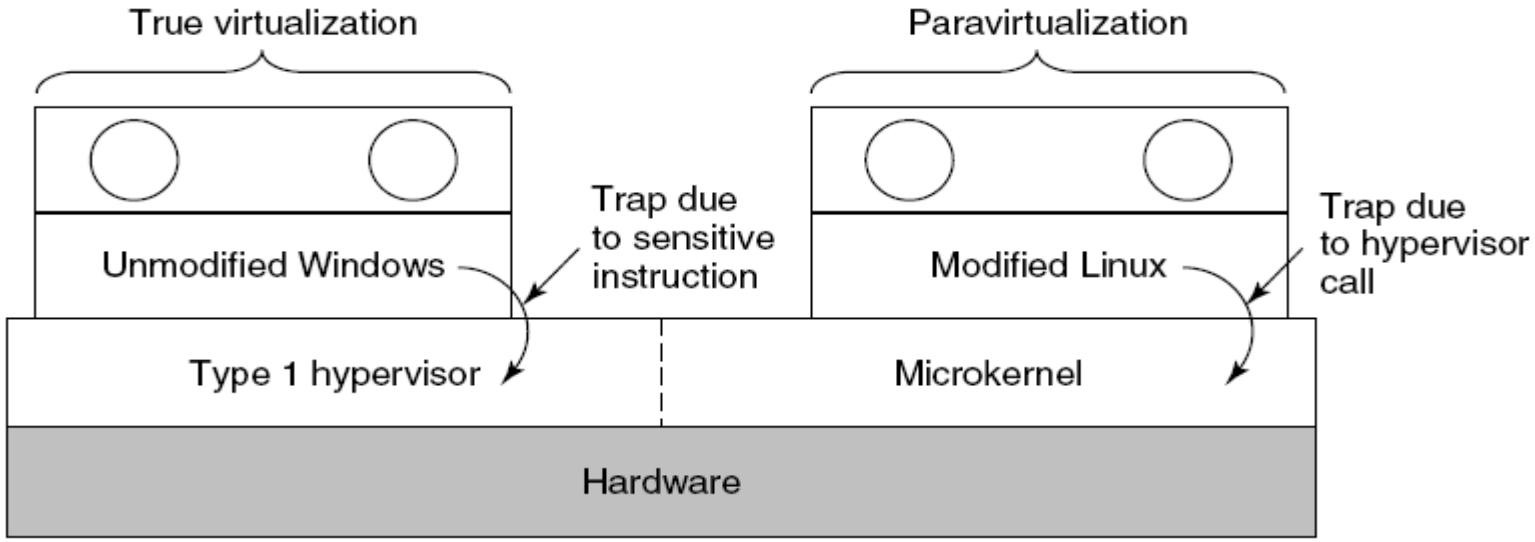
- **Each VM and its guest OS works just as if it was running alone on independent computers**, and **no special modifications or adaptations are needed in the guest operating system**. (no extra patch work needed for guest OS)

TYPES OF VIRTUALIZATION

Para virtualization

- **Para virtualization** is virtualization in which the **guest operating system** (the one being virtualized) **is aware that it is a guest.** (aware that is guest not original OS. Each system call in guest OS is replaced by hyper call. The h/w resources are sharing among VM's) .

- **In Para virtualization modify OS kernel to replace all sensitive instructions with hyper calls – OS behaves like a user program making system calls** – Hypervisor executes the privileged operation invoked by hyper call. (In para virtualization, all system call in guest OS is replaced by hyper call which call hypervisor which in turn makes system call to access h/w resources)



TYPES OF VIRTUALIZATION

2) Network Virtualization

- **It refers to the management and monitoring of a computer network as a single managerial entity** from a single software-based administrator's console. (NV can combine multiple physical networks to one virtual network, or it can divide one physical network into separate, independent virtual networks.Creates virtual networks within physical network. management & monitoring networks functionality is combined into single entity, runs within the VM so that data transfer takes place through that virtual n/w in secured way without sharing with other VM's)
- It is intended to allow **network optimization of data transfer rates, scalability, reliability, flexibility, and security.**
- **Network virtualization is specifically useful for networks experiencing a huge, rapid, and unpredictable increase of usage.**
- The intended result of **network virtualization provides improved network productivity and efficiency.** (Can use the network efficiently and sharing same network among multiple users)

(In network virtualization, multiple sub-networks can be created on the same physical network. This enables restriction of file movement across networks and enhances security, and allows better monitoring and identification of data usage which lets the network administrator's scale up the network appropriately. It also increases reliability as a disruption in one network doesn't affect other networks, and the diagnosis is easier.)

TYPES OF VIRTUALIZATION

- **Two categories:**

  - **Internal: Provide network like functionality to a single system.**

  - **External: Combine many networks, or parts of networks into a virtual unit.**

TYPES OF VIRTUALIZATION

3) Software Virtualization

- **It provides the ability to the main computer to create and run one or more virtual environments.** (Software Virtualization involves the creation of an operation of multiple virtual environments on the host machine, running multiple softwares on a single server, running guest OS on main OS)
- It is used to **enable a complete computer system in order to allow a guest OS to run.**
- **For instance letting Linux to run as a guest that is natively running a Microsoft Windows OS** (or vice versa, running Windows as a guest on Linux).

Types:

- **Operating System** (hosting multiple OS on the native OS)
- **Application Virtualization** (hosting individual applications in a virtual environment separate from the native OS)
- **Service Virtualization** (hosting specific processes and services related to a particular application (Web Service where it takes data from the organization and provides to the user, Data Flow service→ Queue service allows to transfer of data between two VMs, Map Reduce service where runs either Mapper or Reducer with the VM etc)

TYPES OF VIRTUALIZATION

4) Storage Virtualization

- **Storage virtualization is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device** that is **managed from a central console.** (Storage Virtualization is the process which helps in the grouping of physical storage from a number of network storage devices. Therefore, it works as a single storage device.)

- **It provides various advantages as follows:**

  - **Improved storage management** in a heterogeneous IT environment

  - **Easy updates, better availability** (by giving dedicated storage device to individual server may lead to wastage of storage space if not utilized 100% or shortage of storage if more storage space is required than available)

  - **Reduced downtime** (Some technologies optimize the performance by migrating data strategically based upon the utilization. For instance, a frequently used file might be stored on a high performance flash storage system while rarely used files will be placed simply on a slower array.)

  - **Better storage utilization**

  - **Automated management**

TYPES OF VIRTUALIZATION

Types of Storage Virtualization

- **Block-Level -** This type of virtualization works before the file system exists. It replaces controllers and takes over at the disk level.

- **File-Level -** The server that uses the storage must have software installed on it in order to enable file-level usage.

TYPES OF VIRTUALIZATION

5) Memory Virtualization

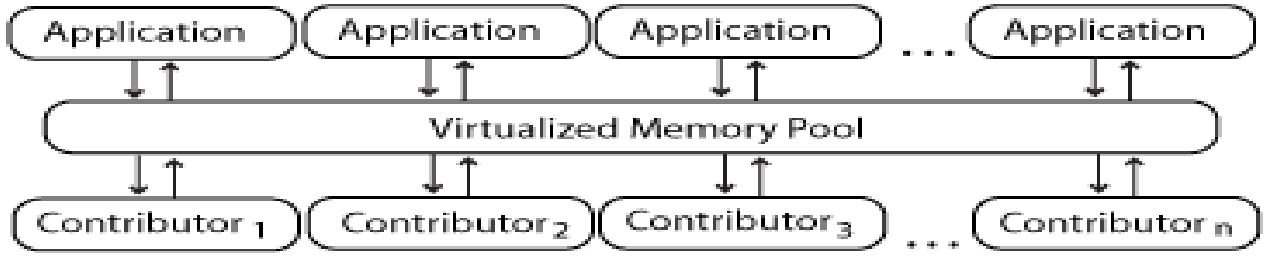
- Memory virtualization is a way to **decouple memory from the server to provide a shared, distributed or networked function.**

(Memory from various servers in the n/w pooled into single memory) (When physical memory around different servers is combined to form a single virtual memory as a pool it is called as Memory Virtualization. With this, you can get the benefit of bigger working memory. Servers are connected in such a way that memory belonging any server can be used and accessed by any other server)

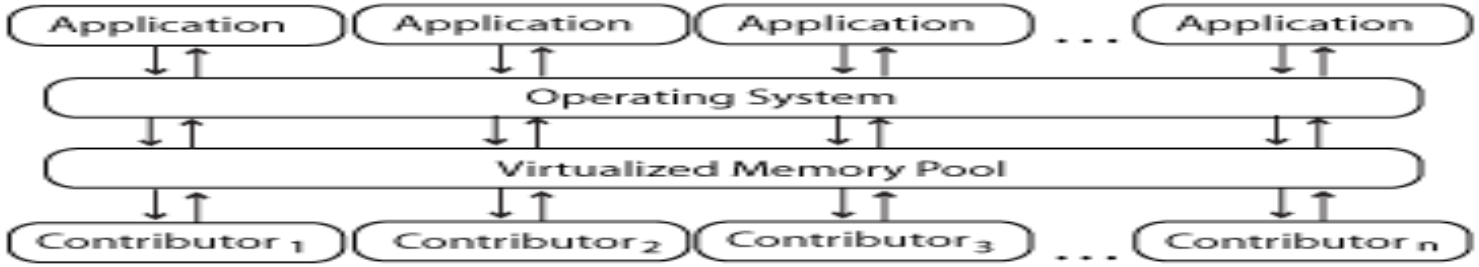
- It enhances performance by providing **greater memory capacity without any addition to the main memory.**

Implementations

- **Application-level integration** – **Applications running on connected computers directly connect to the memory pool through an API** or the file system.



- **Operating System Level Integration** – **The operating system first connects to the memory pool**, and **makes that pooled memory available to applications.**



TYPES OF VIRTUALIZATION

6) Data virtualization

- **Data virtualization is any approach to data management that allows an application to retrieve and manipulate data without requiring technical details about the data, such as how it is formatted at source, or where it is physically located.** (data virtualization is a software layer that hides most of the technical aspects of how and where data is stored for applications. Because of that layer, applications don't need to know where all the data is stored physically, where the database servers run, what the required API is, which database language to use, and so on. To every application using data virtualization, it feels as if it's accessing one large database. It lets you easily manipulate data, as the data is presented as an abstract layer completely independent of data structure and database systems. Decreases data input and formatting errors. Different access method→ Sequential, Direct, Indexed, Storage→ Structures, Unstructured, Object Oriented, Hierarchical)

- It provides a **single customer view** (or single view of any other entity) **of the overall data**.

TYPES OF VIRTUALIZATION

7) Desktop virtualization

- **Desktop virtualization provides a way for users to maintain their individual desktops on a single, central server.** The users may be connected to the central server through a LAN, WAN or over the Internet. (Desktop environment can be virtualized at cloud)
- Desktop virtualization **"virtualizes desktop computers**" and these **virtual desktop environments are "served" to users on the network.**

- **Users interact with a virtual desktop in the same way he/she would use a physical desktop.**
- Another **benefit of desktop virtualization is that it lets users remotely log in to access to their desktop from any location.**
- **It provides a lot of flexibility for employees to work from home or on the go.**
- **It also protects confidential data from being lost or stolen by keeping it safe on central servers.**

PROS AND CONS OF VIRTUALIZATION

Benefits of virtualization offered up by both organizations as well as vendors:

- **Lower overall capital expenditures.** By means of **virtualization we don't need to purchase a server for every single application in our organization. By hosting multiple virtual servers on a single physical machine, dramatically reduces cost overhead.**

- **Automated tasks**. Virtualization automates a number of significant routine IT tasks. (Automatically starting up of VM, if overload automatically migrating to other server, Automatically backup data, Scaling out & scaling in resources based on load on VM).

- **Greater redundancy. Virtualization should improve our uptime.** Virtualization technologies allow **greater safety and security while reducing the points of contact.** (Virtualization allows to maintain redundant copy, if VM crash or fail over automatically starts up redundant VM)

- **Faster deployment. Deploying a virtual machine is overwhelmingly simpler and quick than deploying a physical machine.**

PROS AND CONS OF VIRTUALIZATION

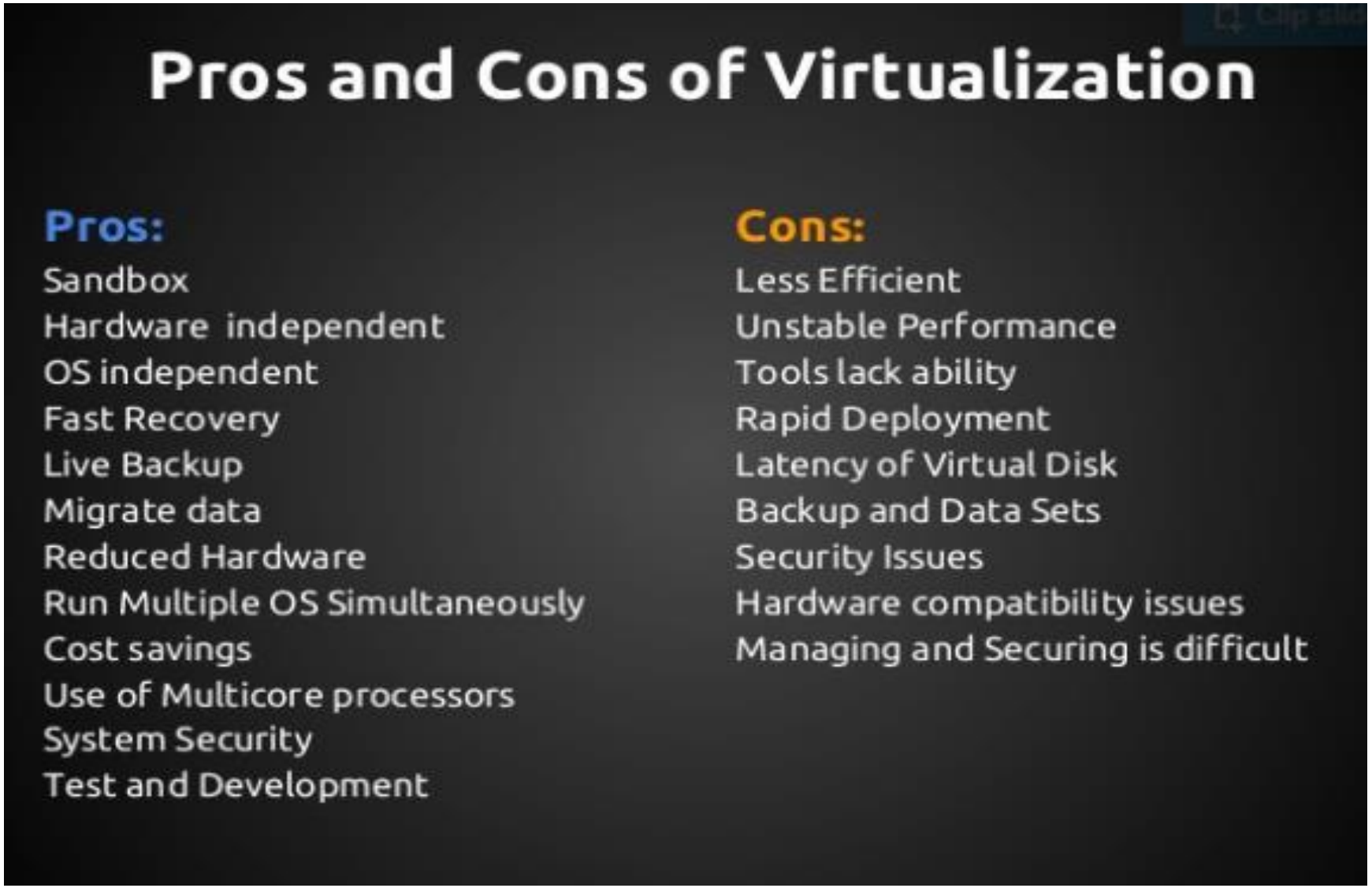
There can be some downsides to virtualization, as well:

- **High upfront expenditures. When we are planning to implementing a virtualization strategy from the ground up, we need to sink more money into hardware for an immediate future.** While will save in the long run, implementation can get pricey. (A 64-bit processor VM Monitor Mode extensions at least 4 GB of RAM, DMA).

- **Not all applications are ready for virtualization. There are still vendors not fully supporting virtualized environments.** (There are still app developing by vendors that cannot run in an virtualized environment)

- **The danger of server sprawl. Because servers are so easy to deploy in a virtualized environment, there's always the danger that new servers will be added even when they're not needed. Instead of the 10 or 20 virtual servers you really need, you might have 30 or 40.** (Simply deploying an VM even when it is not needed may sprawl /slow server)

PROS AND CONS OF VIRTUALIZATION



VIRTUAL MACHINE

- **Virtual Machine** is a completely separate **individual operating system** installation on **main operating system**.

- Virtual machine is a **software implementation** of a physical machine. (VM is just a software pack like folder containing configuration details, OS details, App details and other infm, not signifying any physical structure)

- Virtual machines are divided in two categories based on their use and correspondence to real machine (How it interacts with the physical machine) : **System Virtual Machines and Process Virtual Machines.**

- First category provides a complete system platform that executes complete operating system, second one will run a single program.

- Frequently **multiple virtual machines** with their **own OS's** are used in **server consolidation**, where different services are run in **separate virtual environments**, but on the **same physical machine**. (Server Consolidation→ Applications running on a different servers are consolidated into a single server)

TYPES OF VIRTUAL MACHINE

Virtual machines can be divided into two categories:

1. System Virtual Machines

- **System VM** provides system platform that supports the **sharing of the host computer's physical resources between multiple virtual machines**, **each running with its own copy of the operating system.** The virtualization technique is **provided by a software layer known as a Hypervisor**, which can run **either on bare hardware or on top of an operating system.** (Creates separate multiple VM with OS installed & application running on that on a single physical server, multiple applications runs on its own OS & with its own resources)

2. Process Virtual Machine

- Process VM provides a **platform-independent programming environment that masks the information of the underlying hardware or operating system and allows program execution to take place in the same way on any given platform.**

(provides a platform independent pgmming environment, **VM is created (virtually)** when that process is started and destroyed when it exit. No separate VM with OS installed & application running on that, multiple applications runs on same OS with time sharing and interprets the instruction set to an underlying architecture & OS. Eg: JVM running multiple threads & converting instruction set of app into underlying architecture & OS) (A Process virtual machine, sometimes called an application virtual machine, runs as a normal application inside a host OS and supports a single process. It is created when that process is started and destroyed when it exits. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system, and allows a program to execute in the same way on any platform).

TYPES OF VIRTUAL MACHINE

(just for your reference)

A **Process virtual machine**, sometimes called an application virtual machine, runs as a normal application inside a host OS and supports a single process. It is created when that process is started and destroyed when it exits. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system, and allows a program to execute in the same way on any platform. *For example Wine software in Linux helps to run Windows application* .

A **System virtual machine** provides a complete system platform which supports the execution of a complete operating system (OS),Just like you said *VirtualBox* is one example.

PROCESS VIRTUAL MACHINE

Process VM provides users **Replication**, **Emulation** and **Optimization**

- **Replication:** **Running multiple** copies/**applications** on **single machine**, creating an illusion to app that entire machine is given to it. (Word processing, Printing, Browser, listening to music etc or in a single website, multiple threads one executing scroll message, one executing displaying menus, one executing video→ for each one process is created)

- **Emulation:** **Emulates** (Matching) **instruction set** to an **underlying architecture** & **OS**. (Convert byte code into opcode for an **underlying architecture** & **OS)**

- **Optimization:** **Optimizes** code at Runtime. (for ex: Simple using initialization with in loop→ removes & places before the loop, dead code elimination→ initializes variable but never used)

PROCESS VIRTUAL MACHINE (Cont....)

1) Multiprogramming (Running Multiple applications on a single system)

- The **combination** of the **OS call** interfaces/**statements** and **user instruction set** forms a machine (**process VM**) that executes a user forms. (The combination of your program instructions & system statements together forms a VM→ process VM).

- **OS supports multiple users through multiprogramming**, where **each user process is given an illusion that complete machine is with itself.** (Supporting multiprogramming, running multiple programs at the same time → process VMs (Threads) listening to music, printing, Typing in word)

- **OS shares the hardware devices on time basis** in effect. **OS provides a process VM for each of the concurrently executing applications.** (listening to music, printing, Typing in word→ all 3 threads shares & runs simultaneously in such a way that CPU switches so fast that we even not possible to identify)

- Each **user process** (process VM) is given a **separate address space** and given to access to its file structure.

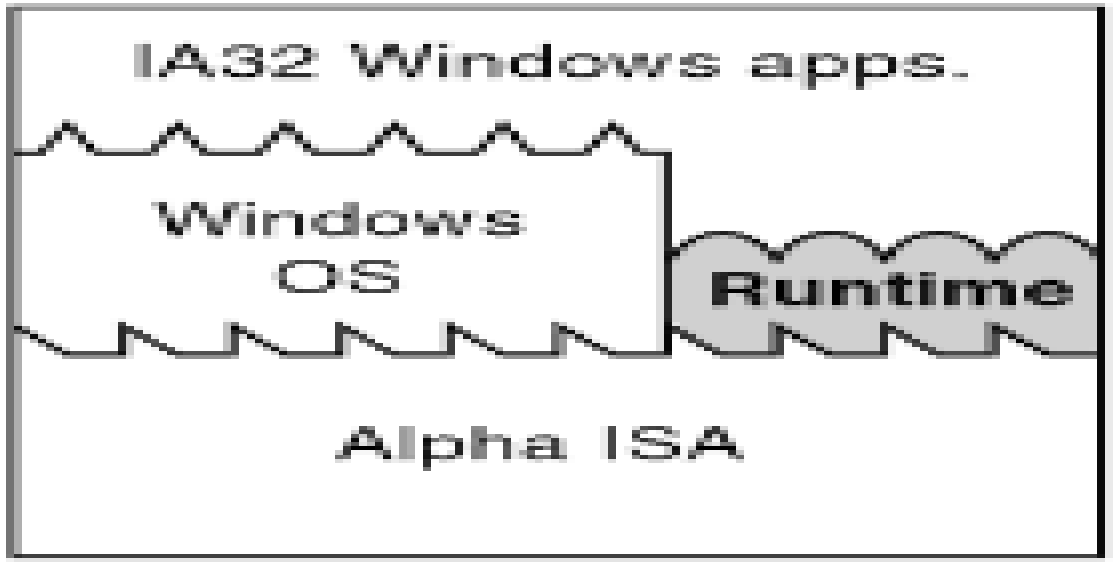
PROCESS VIRTUAL MACHINE (Cont....)

2. Emulate and Dynamic Binary Translators

- **A more challenging problem for process level VM is to support program binaries compiled for one instruction set/architecture to be run by another host's hardware.**

- **Example of emulating process VM is illustrated in below figure.**

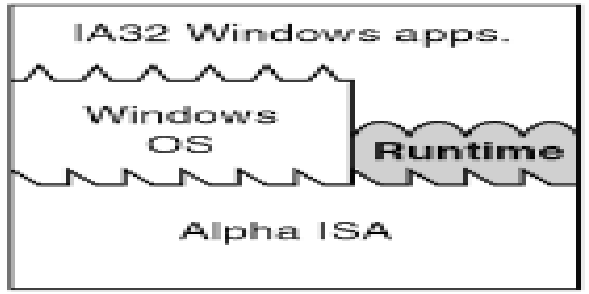
(Appl designed for Intel architecture instruction set executed on Window OS now running on Alpha architecture system)



A Process VM That Emulates Guest Applications. *The Digital FX!32 system allows Windows IA-32 applications to be run on an Alpha Windows platform.*

PROCESS VIRTUAL MACHINE (Cont....)

- As shown in figure, the **OS may be same for both guest process** & **host platform**, although in **other cases the OS may differ as well**.



- The above figure illustrates that **Intel IA-32 application** binaries **compiled for Windows NT executing on Alpha hardware platform running Windows OS.**

- The most **straight forward emulation method is Interpretation. An interpreter fetches, decodes and emulates each and individual instructions.** This can be **relatively slow**. For better performance, a technique called **Binary Translation** is used.

- With Binary translation, **blocks of source instructions are converted into target instructions** that perform equivalent functions. **Once the block of instructions are converted, the translated instructions can be cached and repeatedly executed must faster instead of interpreting each time.** Since Binary translation is most important feature for process VM, they are some times called as **Dynamic Binary Translators.** (Stores binary representation of frequently executed codes in the cache and uses repeated→ Looping instructions) (In computing, binary translation is a form of binary recompilation where sequences of instructions are translated from a source instruction set to the target instruction set. In some cases such as instruction set simulation, the target instruction set may be the same as the source instruction set, providing testing and debugging features such as instruction trace, conditional breakpoints and hot spot detection. The two main types are static and dynamic binary translation. Translation can be done in hardware (for example, by circuits in a CPU) or in software (e.g. run-time engines, static recompiler, emulators).

PROCESS VIRTUAL MACHINE (Cont....)

3) Same-ISA Binary Optimizer

- **Dynamic Binary translators** not only **translates from source to target code**, but also performs some **code optimizations**. This leads naturally to VM where instruction set used by the host and the guest are the same.

- Same ISA-Binary Optimizers are implemented in similar way including **staged optimization** and **Software Caching** of optimized code. (Optimizes the frequently used code and caches in cache memory→ using of initialization of variable within the loop)

- **Dynamic optimizers** are **most effective for the source binaries that are not optimized.** (Binary equivalent of source program that are not optimized takes more & unnecessary execution time. Such code part is replaced with the optimized code reduces execution time)

- A dynamic optimizer collects a profile and use this profile information to optimize the binary code on the fly.

PROCESS VIRTUAL MACHINE (Cont....)

High Level Language Virtual Machines: Platform Independence

- **HLL VM are similar to process VM, but focus on minimizing hardware specific and OS specific.** (In Process VM, application written for one platform need to be converted into another platform instruction by instruction. i.e application written for Intel architecture need to be converted into instructions belonging to Alpha architecture instruction while executing on Alpha system or application written for IBM architecture need to be converted into instructions belonging to Alpha architecture instruction while executing on Alpha system. But in HLL VM, applications are designed in such way that converts into a code (Byte code) in a more generic way that can run on any system by converting into underlying architecture)

- **For the process VM, cross platform portability is very important.**

- For Eg: **Application platform compiled for popular Intel -32 PC architecture need to be converted to less popular Alpha platform.** This **cross platform need to be done case by case** and requires a great deal of **programming effort**.

- Fine, **if one wanted to run application designed for Intel platform to be run on multiple platform SPARC, PowerPC** etc, then need **multiple cross platform**. **The problem would be even more if need to be run on different OS.**

- To overcome this problem, **Full Cross platform** portability can be used.

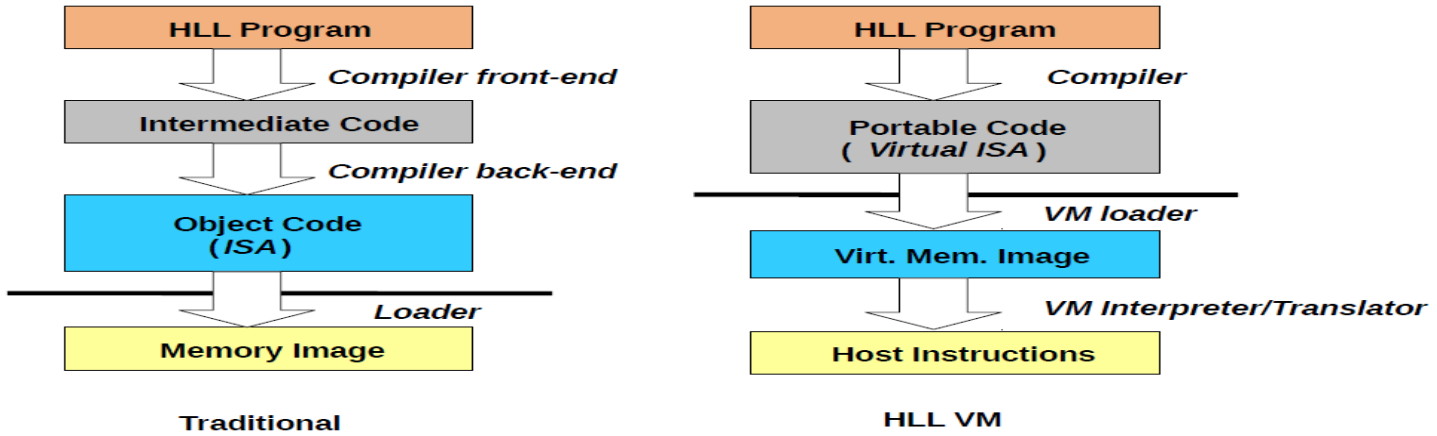
PROCESS VIRTUAL MACHINE (Cont....)

High Level Language Virtual Machines: Platform Independence

- **In Full cross platform, VM/application environment does not corresponds to any real platform, rather instructions converted to more generic (like byte code) features of high level language, minimizing hardware specific and OS specific features.** (appl converted to more generic way not specific to any hardware or any OS, code that reduces number of hardware specific & OS specific instructions→ converts instruction to code that access to hardware & system calls in generic way, can access any architecture and any OS).
- Popular example for HLL VM is **Java VM.**

PROCESS VIRTUAL MACHINE (Cont….)

High Level Language Virtual Machines: Platform Independence

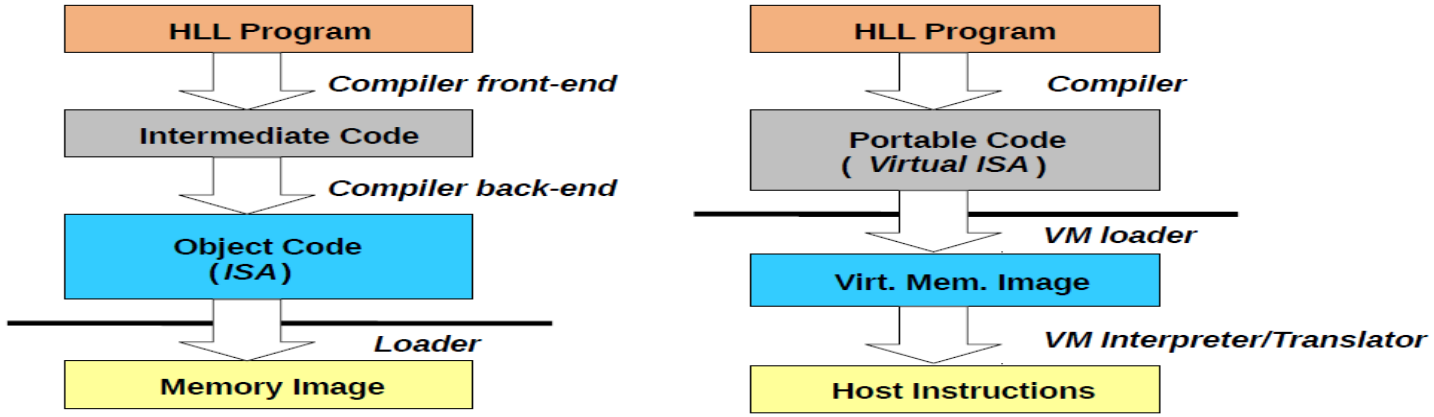


High-Level Language Environments. *(a) A conventional system, where platform-dependent object code is distributed; (b) an HLL VM environment, where portable intermediate code is "executed" by a platform-dependent virtual machine.*

- In a **conventional system** as shown in the figure**, the complier consists of frontend that performs lexical, syntax and semantic analysis to generate intermediate code.**
- **These intermediate code does not contain specific register assignments.**
- Then **code generator takes intermediate code and generates a binary containing machine code for a specific architecture and OS.**
- **This machine code is distributed and executed on platforms that support given architecture and OS.** (Machine code will be loaded to memory and executed on the generated architecture and OS)
- **To execute the program on different platform, it must be recompiled for that platform.**

PROCESS VIRTUAL MACHINE (Cont….)

High Level Language Virtual Machines: Platform Independence



High-Level Language Environments. *(a) A conventional system, where platform-dependent object code is distributed; (b) an HLL VM environment, where portable intermediate code is "executed" by a platform-dependent virtual machine.*

- **In HLL VM, this model is changed.** As shown in the figure, **compiler generates machine code** (called as HLL VM) **similar to intermediate code** (Eg we can say Byte Code in Java). This is generic stack based ISA.

- **VM loader converts that into VM image that minimizes hardware specific & OS specific, and loads.** (A virtual machine image is a single file which contains a virtual disk that has a bootable operating system installed on it). (VM Loader converts the byte code into file wrt underlying architecture and instruction set and loads it into the memory → Translator converts that into machine code (This is done by JVM at the target machine)

- **Finally interpreted & distributed for execution on any platform**.

(Traditional→ Convert HLL program into machine code. Needs to run on different platform again need to compile HLL VM-> example JVM convert into portable code called byte code. With this byte code can run on any machine with different platform) Traditional→ Conventional ISA – guest ISA registers -> host registers is a problem. HLL VM→ – stack-oriented)

HLL VM (Cont…)

- In its simplest form, VM contains an interpreter that takes each instruction, decodes it and then performs the required state transformation. In more sophisticated, higher performance VMs the abstract machine code may be compiled into host machine code for direct execution on the host platform.

An advantage of an HLL VM is that software is easily portable, once the VM is implemented on a target platform. While the VM implementation would take some effort, it is a much simpler task than developing a compiler for each platform and recompiling every application when it is ported. It is also simpler than developing a conventional emulating process VM for a typical real-world ISA.

Virtual Machines

Virtual Machines can be divided into two categories:

1. Process Virtual Machine

- Process VM provides a **platform-independent programming** environment that masks the information of the underlying hardware or operating system and allows program execution to take place in the same way on any given platform. (provides a platform independent pgmming environment, **VM is created (virtually)** when that process is started and destroyed when it exits. Separate VM with OS installed & application running on that is not created, multiple applications runs on same OS with time sharing and interprets the instruction set to an underlying architecture & OS. EG: JVM running multiple threads & converting instruction set of app into underlying architecture & OS)

2. System Virtual Machines

- System VM provides system platform that supports the sharing of the host computer's physical resources between multiple virtual machines, each running with its own copy of the operating system. The virtualization technique is provided by a software layer known as a **Hypervisor**, which can run either on bare hardware or on top of an operating system. (Creates separate multiple VM with OS installed & application running on that, multiple applications each running with its own copy of the operating system on the same physical server)

System Virtual Machines

- **System VM provides system platform that supports the sharing of the host computer's physical resources between multiple virtual machines, each running with its own copy of the operating system.**

- **The virtualization technique is provided by a software layer known as a Hypervisor**, which can run either on **bare hardware** or on **top of an operating system**.

- **A System virtual machine provides a complete system platform which supports the execution of a complete operating system (OS)**, VirtualBox is one example.

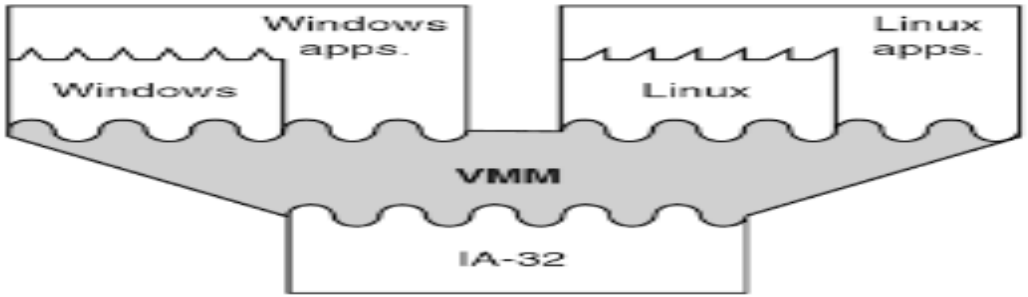
- **System VM provides a complete system environment in which many processes possibly belonging to multiple users can coexists.**

- **By using system VM, a single host hardware platform can support multiple guest OS environment simultaneously.**

- **Software running on one guest system is isolated from software running on other guest OS.** Furthermore**, if security on one guest system is comprised or if the guest OS suffers a failure, the software running on other guest system is not affected.**

System Virtual Machines

- **System VM has a ability to support different OS simultaneously. E.g. Windows and Linux OS can run simultaneously on a single hardware as shown in the figure.** (but not possible in process VM)



A System VM That Supports Multiple OS Environments on the Same Hardware.

- **In System VMs, platform replication is the major feature supported by VMM, but the problem of dividing a single set of hardware resources among multiple guest OS.**

- **The VMM has a access to and manages all the hardware resources. The guest OS and application running on it is under the control of VMM.** (whenever app wants to read from the memory or write to the memory, it must issue command to VMM, VMM then issue command to main OS & access resources)

- When a guest OS performs certain operations. **Such as privileged instructions that involves access to hardware is intercepted and executed by VMM on behalf of guest.** Guest OS is unaware of " **Behind the Scenes**" work performed by the VMM.

Implementation of System Virtual Machines

There are two types of implementation of System VM

- **In the first type of implementation, VMM is first places on bare hardware** and **virtual machines fit on the top of it.** (No main OS)

- **VMM runs in most highly privileged mode, while all the guest operating systems runs in less privileged mode**.

- **VMM in a completely transparent way intercept and performs all OS's actions that interact with hardware resources.** (Because bare metal) (i.e., accessing hardware resources are done by the VMM)

Drawbacks of this type of System VM implementation are:

1. **Installation requires wiping an existing system** and **starting from scratch**, **first installing VMM and then installing guest OS on the top**.

2. **I/O device drivers must be available for installation in the VMM**, **because VMM directly interacts with the I/O devices.**(No host OS→ Bare metal, I/O device drivers must be installed and make available to VMM)

Implementation of System Virtual Machines

There are two types of implementation of System VM

- **In the second type of implementation, VMM is installed on the top of existing host OS**. This type of implementation is called as **Hosted VM.**

- **In this type, installation process is similar to adding a typical application program.** (VMM installation is just like installing app)

- **Here VMM depends on host OS for device drivers and other lower level services.**

Drawbacks of this type of System VM implementation are:

- **There may be chance of loss of efficiency, because more layers of software involved when OS service is required** (if need to write into the memory→application→Guest OS→ VMM→ Host OS→ Memory, But in first implementation bare metal, Application→Guest OS→ VMM→ Memory).

- **This hosted VM implementation approach is taken in Vmware implementation**.

Whole System VMs: Emulation

- **In conventional system VMs, guest OS, host OS and application use same ISA as the underlying hardware.** (i.e, 3 or 4 VMs created with the same environment)

- **But some times host OS and guest OS do not have common ISA.** (Appln written for intel arch and windows OS, but to be run on Apple s/m which is diff arch)

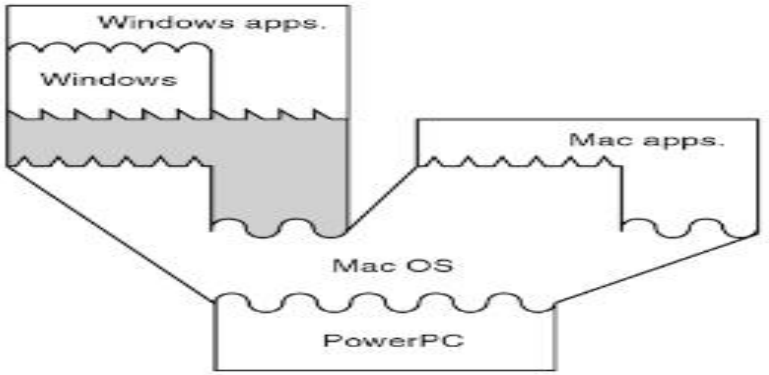
- **For eg: Apple PowerPC based systems and Windows PCs use different ISA and OS also different. Similarly Sun Microsystem servers use different OS and ISA than Windows PCs., because system software's closely tied to its hardware systems** (ISA completely different, On apple PC OS cannot run Windows guest OS & On Sun Microsystems Servers cannot run Windows guest OS) apple PC OS systems calls specifically tied to its apple hardware systems. On Apple s/m, cannot install windows as guest OS)

- **This requires purchase of multiple platform types, which may be complicated.**(Appln designed for Apple ISA to be run on only Apple s/m and appln designed for Intel arch to be run only on Intel s/m, appln designed for Sun Microsystem to be executed only on Sun microsystem)

- **This motivates system VMs to support both guest OS and applications to run on host system that runs different ISA and OS.** This type of system is called as **Whole System VMs.**

Whole System VMs: Emulation

- **Because the ISAs are different, both application and OS code requires emulation via binary translation.** (if linux application to be run on windows guest OS in the Apple system, both app & windows OS need binary translation to its equivalent Apple ISA).
- **The below figure shows Whole system VM built on the top of conventional system with its own OS & application programs.** (Whole system VM→ complete VM including guest app & guest OS need to emulate to its underlying ISA) (Entire app and Guest OS are treated like a single app and converted to be run on underlying architecture.)



A Whole-System VM That Supports a Guest OS and Applications, in Addition to Host Applications.

- **As shown in figure, Windows systems to run on Macintosh platform.** Complete VM software, **guest OS and guest application are just like one very big application implemented on host OS and hardware.** (application call statement, Windows system call statement, VMM instructions completely converted into underlying system call statement)
- **Meantime host OS can also continue to run application compiled for its native ISA; this feature is illustrated in the right hand section of the figure.**

Whole System VMs: Emulation

- **To implement system VM of this type, requires emulation of application instructions, convert guest OS ISA to its equivalent OS call to call host OS.** (Converting app instruction, guest OS system calls statement into its equivalent host OS system call statement→ complete emulation)

Full Virtualization: BINARY TRANSLATION

- Depending on implementation technologies, hardware virtualization can be classified into two categories:
  - **Full virtualization**
  - **Host-based virtualization**.

Full virtualization

- **In Full Virtualization, the complete simulation of the actual hardware takes place.** That is, the hypervisor provides complete abstraction. (The h/w resources is completely emulated to each individual VM's) .
- **Each guest operating system is unaware that it is in a virtualized environment**, and therefore hardware is virtualized by the host operating system so that the guest can issue commands as it thinks is actual hardware, but really are just simulated hardware devices created by the host.
- **It relies on binary translation to trap and to virtualize the execution of certain sensitive, non virtualizable instructions.**
- **The guest OS and their applications consist of noncritical and critical instructions.** (Non-critical→ reading from the memory, storage Critical→ writing to memory, modifying memory contents etc).

Full Virtualization: BINARY TRANSLATION

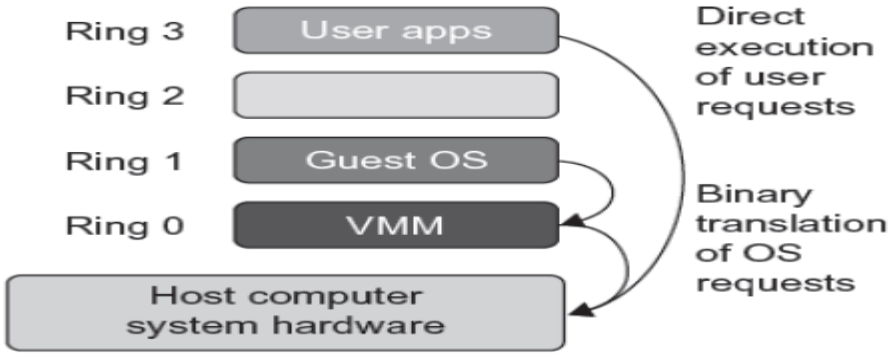
- **With Full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM.**(app can perform read from memory directly while write operation needs issue command to VMM, the to host OS then to memory)

Why are only critical instructions trapped into the VMM?

- **This is because binary translation can incur a large performance overhead. Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do.** Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.

Binary Translation of Guest OS Requests Using a VMM

- This approach was implemented by VMware and many other software companies.



- **As shown in Figure, VMware puts the VMM at Ring 0 and the guest OS at Ring 1.**

Full Virtualization: BINARY TRANSLATION

- **The instructions stream are scanned and identifies the privileged, control- and behaviour-sensitive instructions.**
- **When these instructions are identified, they are trapped into the VMM, which emulates the behaviour of these instructions.** (Converts instruction/system call to VMM in turn Host OS call, in turn access hardware)
- **The method used in this emulation is called Binary Translation.**
- **Thus, Full virtualization combines binary translation and direct execution.**
- **The performance of full virtualization may not be ideal, because it involves binary translation which is rather time-consuming.**
- **Particularly, I/O-intensive applications is a really a big challenge.**
- **Binary translation employs a code cache to store translated hot instructions to improve performance, but it increases the cost of memory usage.**

Host-Based Virtualization: BINARY TRANSLATION

- **An alternative VM architecture is to install a virtualization layer on top of the host OS.** (Full Virtualization can create using virtualization software installed either on bare metal or on top of main OS, but Host based VM created using virtualization software installed on top of main OS)

- **Host OS is still responsible for managing the hardware.**

- **The guest OS are installed and run on top of the virtualization layer.**

- **Dedicated applications may run on the VMs. Certainly, some other applications can also run with the host OS directly**. (Supports both few app can run on VM and few app can run directly on host OS)

Host-Based Virtualization: BINARY TRANSLATION

- **This host-based architecture has some distinct advantages.**

  - **First, the user can install this VM architecture without modifying the host OS. The virtualizing software can rely on the host OS to provide device drivers and other low-level services.** (not like bare metal). This will **simplify the VM design and ease its deployment.**

  - **Second, the host-based approach appeals to many host machine configurations.**

- **The performance of the host-based architecture may also be low. When an application requests hardware access, it involves four layers of mapping which downgrades performance significantly.** (App statement request access to memory→Translates to Guest OS call statement→ Translates to VMM→Translates to Host OS call statement, then access memory)

- **When the ISA of a guest OS is different from the ISA of the underlying hardware, binary translation must be adopted.** Although the host-based architecture has flexibility, the performance is too low to be useful in practice.

Live VM Migration Steps and Performance Effects

- **Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters.**
- **The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks.**
- **In a cluster, built with mixed nodes of host and guest systems, the normal method of operation is to run everything on the physical machine.** (VM cluster created with set of VMs created on multiple hosts say for eg: 1 VM cluster running 10 modules of an application→ 10 VMs created on 10 physical machines with one running guest system acts like a master which takes care of all 10 VMs operation in the cluster. Normally, VMs runs on physical machine. If any VM running on any physical machine fails, then its role could be replaced by another VM on different physical machine such that the target physical machine connected to the same guest OS)
- **When a VM fails, its role could be replaced by another VM on a different node, as long as they both run with the same guest OS.**
- **The advantage is enhanced failover flexibility** (supports failover situation)**.**
- The potential **drawback is that a VM must stop playing its role if its residing host node fails.** However, **this problem can be mitigated with VM live migration.**

Live VM Migration Steps and Performance Effects

- **There are four ways to manage a virtual cluster:**

  - **Guest-Based Manager**
  - **Host-Based Manager**
  - **Independent Cluster Manager**
  - **Integrated Cluster**

Live VM Migration Steps and Performance Effects

Guest-Based Manager

- **Guest-Based Manager, by which the cluster manager resides on a guest system.** (VM cluster consisting of multiple VMs from different physical machine designed and cluster manager runs on the top of VM cluster. Host System→ Physical System, Guest System→ Virtual System)
- In this case, **multiple VMs form a virtual cluster.**
- For example, **openMosix is an open source Linux cluster running different guest systems on top of the Xen hypervisor.**
- Another example is **Sun's cluster Oasis, an experimental Solaris cluster of VMs supported by a VMware VMM.**

Host-Based Manager

- **We can build a cluster manager on the host systems.** (Cluster Manager software runs on a dedicated physical machine)
- The **host-based manager supervises the guest systems** and **can restart the guest system on another physical machine.** (Cluster manager running on dedicated physical system monitors functionality of virtual cluster. If any VM fails, starts a VM on another physical machine)
- **A good example is the VMware HA system that can restart a guest system after failure.**

Live VM Migration Steps and Performance Effects

- **These two cluster management systems are either guest-only or host-only, but they do not mix.**

Independent cluster manager

- **A third way to manage a virtual cluster is to use an independent cluster manager on both the host and guest systems.** (Cluster manager for host→ takes care of its physical machine and Cluster manger for virtual cluster→ takes care of virtual cluster)

- **This will make infrastructure management more complex, however.**

Integrated cluster

- **Finally, we can use an integrated cluster on the guest and host systems.**

- **This means the manager must be designed to distinguish between virtualized resources and physical resources.** (One cluster manager that integrated with host cluster manager and guest cluster manager that takes care of both physical machine as well as virtual cluster)

Live VM Migration Steps and Performance Effects

- **VMs can be live-migrated from one physical machine to another; in case of failure, one VM can be replaced by another VM.**

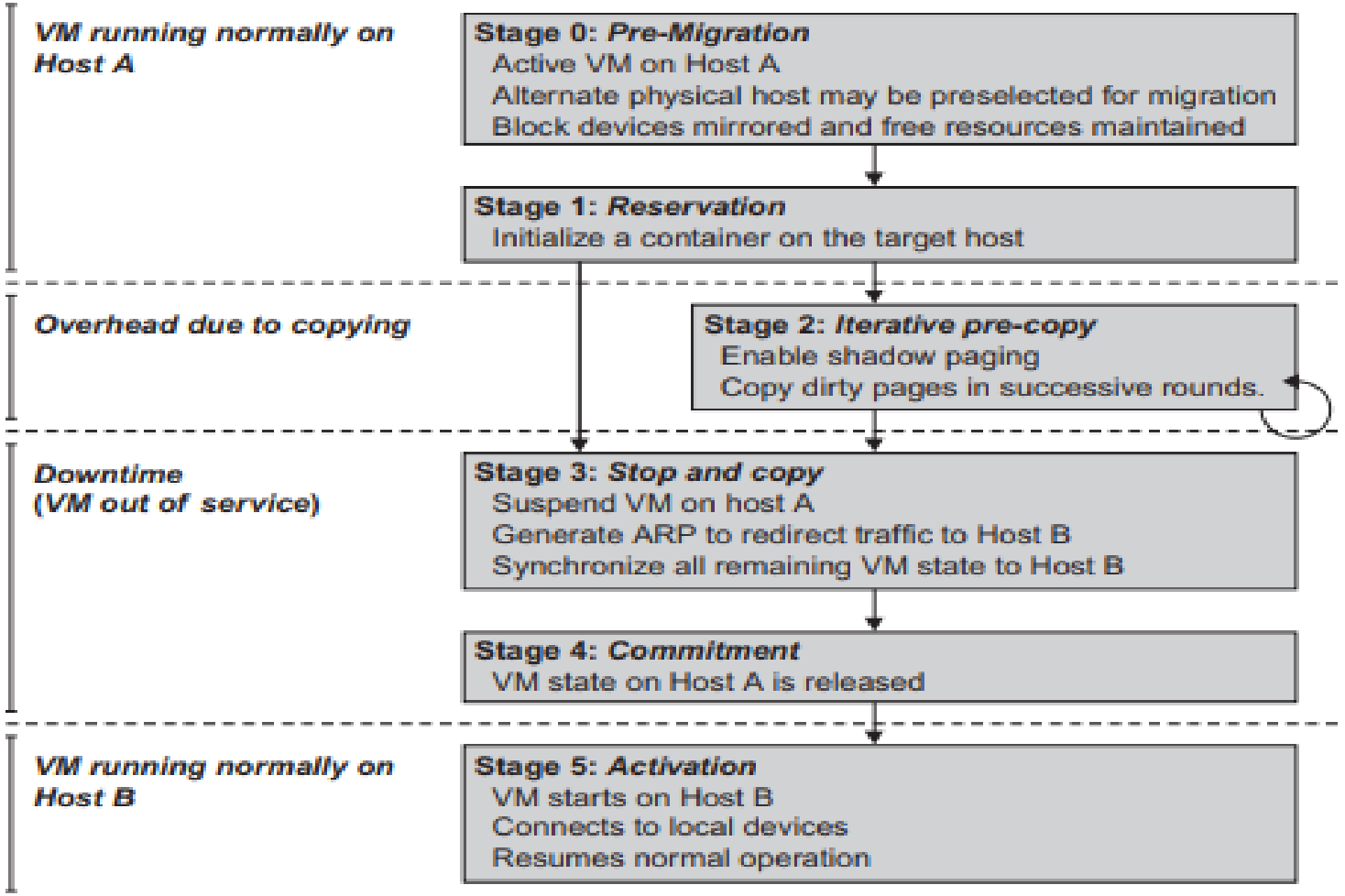
- **Virtual clusters can be applied in computational grids, cloud platforms, and high-performance computing (HPC) systems.**

- **Live migration is the process of transferring a live virtual machine from one physical host to another without disrupting its normal operation.** (without interrupting its operation)

- **Live migration enables the porting(shipping) of virtual machines and is carried out in a systematic manner to ensure minimal operational downtime.**

Live VM Migration Steps and Performance Effects

- **Below figure shows the process of life migration of a VM from host A to host B. The migration copies the VM state file from the storage area to the host machine.**



Live migration process of a VM from one host to another.

Live VM Migration Steps and Performance Effects

- **Live migration of a VM consists of the following six steps:**

➢ Steps 0 and 1: Start migration

- **This step makes preparations for the migration, including determining the migrating VM and the destination host.**

- **Although users could manually make a VM migrate to an appointed host, in most circumstances, the migration is automatically started by strategies such as load balancing and server consolidation.**

➢ Steps 2: Transfer memory

- **Since the whole execution state of the VM is stored in memory, sending the VM's memory to the destination node ensures continuity of the service provided by the VM.**

- **All of the memory data is transferred in the first round, and then the migration controller recopies the memory data which is changed in the last round. These steps keep iterating until the dirty portion of the memory is small enough to handle the final copy.**

- **Although precopying memory is performed iteratively, the execution of programs is not obviously interrupted.**

Live VM Migration Steps and Performance Effects

- **Live migration of a VM consists of the following six steps:**

➢ Step 3: Suspend the VM and copy the last portion of the data

- **The migrating VM's execution is suspended when the last round's memory data is transferred.**
- **Other nonmemory data such as CPU and network states should be sent as well.**
- **During this step, the VM is stopped and its applications will no longer run. This "service unavailable" time is called the "downtime" of migration, which should be as short as possible so that it can be negligible to users.**

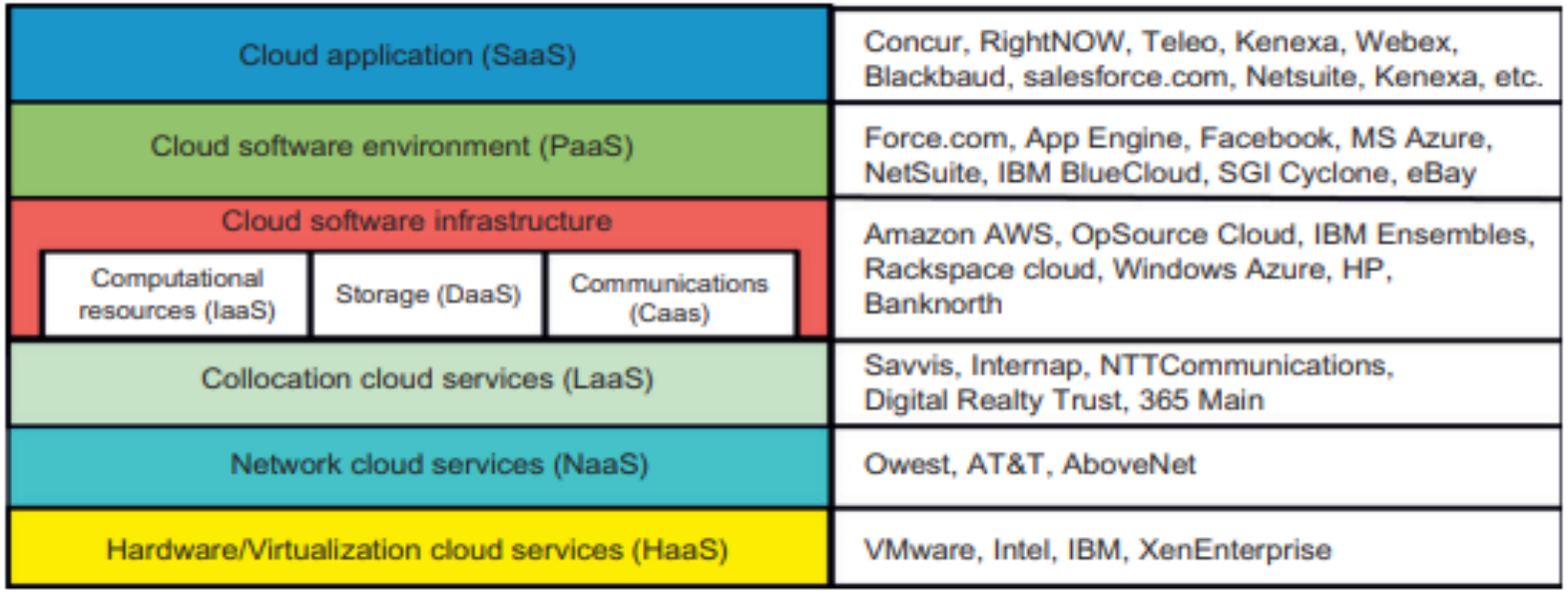
➢ Steps 4 and 5: Commit and activate the new host

- **After all the needed data is copied, on the destination host, the VM reloads the states and recovers the execution of programs in it, and the service provided by this VM continues.**
- **Then the network connection is redirected to the new VM and the dependency to the source host is cleared.**
- **The whole migration process finishes by removing the original VM from the source host.**

UNIT-3

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services (the various cloud service models and their extensions)



A stack of six layers of cloud services and their providers

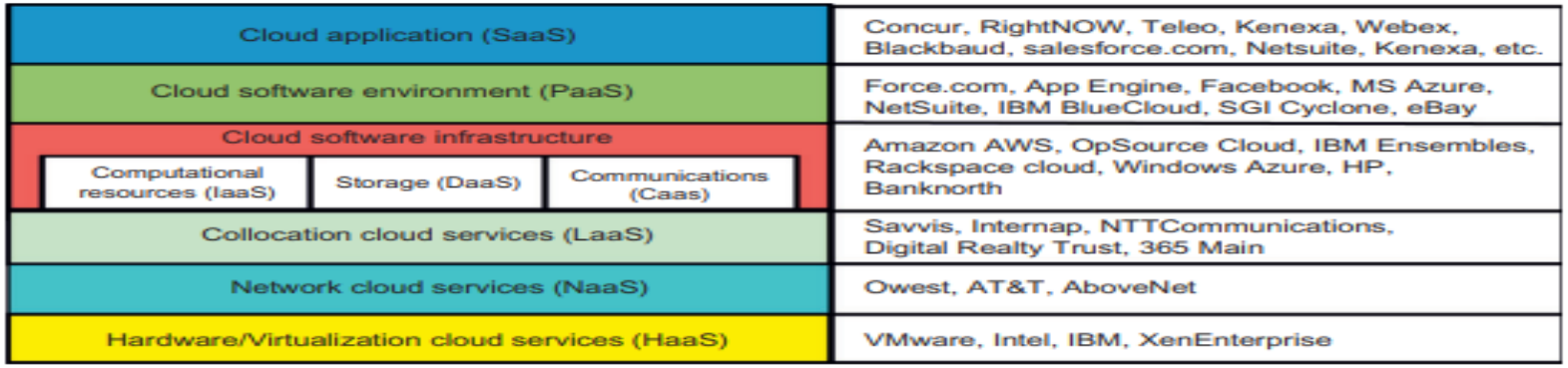
- **Figure shows six layers of cloud services, ranging from hardware, network, and collocation to infrastructure, platform, and software applications.**

- **The top three service layers as SaaS, PaaS, and IaaS.**

- **The cloud platform provides PaaS, which sits on top of the IaaS infrastructure. The top layer offers SaaS. Although the three basic models are dissimilar in usage, they are built one on top of another.**

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services (the various cloud service models and their extensions)



A stack of six layers of cloud services and their providers

- **The bottom three layers are more related to physical requirements. The bottommost layer provides Hardware as a Service (HaaS).**
- **The next layer is for interconnecting all the hardware components, and is simply called Network as a Service (NaaS).** (allowing companies to set up their own networks entirely without hardware). **Virtual LANs fall within the scope of NaaS.**
- **The next layer up offers Location as a Service (LaaS), which provides a collocation service to house, power, and secure all the physical hardware and network resources.** (LaaS is the facility that offers space with the proper power, cooling and security to host businesses' computing hardware and servers).
- **The cloud infrastructure layer can be further subdivided as Data as a Service (DaaS) and Communication as a Service (CaaS) in addition to compute.**

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services (the various cloud service models and their extensions)

- As shown in Table, cloud players are divided into three classes:
  - **Cloud service providers and IT Administrators**
  - **Software developers or Vendors**
  - **End Users or Business Users**

Table Cloud Differences in Perspectives of Providers, Vendors, and Users

Cloud Players	IaaS	PaaS	SaaS
IT administrators/cloud providers	Monitor SLAs	Monitor SLAs and enable service platforms	Monitor SLAs and deploy software
Software developers (vendors)	To deploy and store data	Enabling platforms via configurators and APIs	Develop and deploy software
End users or business users	To deploy and store data	To develop and test web software	Use business software

- **These cloud players vary in their roles under the IaaS, PaaS, and SaaS models.**
- **The table entries distinguish the three cloud models as viewed by different players.** (Table shows how three players view the three cloud models)
- From the **software vendors' perspective**, **application performance** on a given cloud platform is most important.(Designing application with optimized performance wrt time, space, works for all scenario)
- From the **providers' perspective**, cloud **infrastructure performance** is the primary concern.(optimized CPU utilization, Storage utilization)
- From the **end users' perspective**, the **quality of services**, including **security**, is the most important

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

1. Cloud Service Tasks and Trends

- The **top layer in the cloud service is SaaS applications** for business applications.
- For example, **CRM is heavily practiced in business promotion, direct sales, and marketing services**.
- **CRM offered the first SaaS on the cloud successfully.**
- The approach is to **widen market coverage by investigating customer behaviors and revealing opportunities by statistical analysis.**
- **SaaS tools also apply to distributed collaboration** (Google docs), and **financial and human resources management.**
- These cloud services have been growing rapidly in recent years.
- **PaaS is provided by Google, Salesforce.com, and Facebook** (Facebook service), among others.
- **IaaS is provided by Amazon, Windows Azure, and RackRack**, among others.
- **Collocation services require multiple cloud providers to work together to support supply chains in manufacturing.**
- **Network cloud services provide communications** such as those by AT&T, Qwest, and AboveNet.

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

2. Software Stack for Cloud Computing

- **The overall software stacks are built from scratch to meet rigorous goals.**

- **Developers have to consider how to design the system to meet critical requirements such as high throughput, HA, and fault tolerance.** (Developers need to think in designing a software that takes care of all these layers/services at different levels and providing these services to users in such a way the meet required throughput, HA, and fault tolerance).

- **Even the operating system might be modified to meet the special requirement of cloud data processing.** (Even OS that runs at cloud data center need to be modified that takes care of all these services NaaS, LaaS etc.)

- **the overall software stack structure of cloud computing software can be viewed as layers. Each layer has its own purpose and provides the interface for the upper layers just as the traditional software stack does. However, the lower layers are not completely transparent to the upper layers.** (SaaS layer takes care of providing/sharing softwares to customers, PaaS SaaS layer takes care of providing platform to customers, IaaS layer takes care of sharing hardwares to customers)

INTER-CLOUD RESOURCE MANAGEMENT

Extended Cloud Computing Services

3. Runtime Support Services

- As in a cluster environment, there are also some runtime supporting services in the cloud computing environment.
- Cluster monitoring is used to collect the runtime status of the entire cluster.
- The scheduler queues the tasks submitted to the whole cluster and assigns the tasks to the processing nodes according to node availability.
- The distributed scheduler for the cloud application has special characteristics that can support cloud applications, such as scheduling the programs written in MapReduce style.
- The runtime support system keeps the cloud cluster working properly with high efficiency.

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

- **Cloud architecture puts more emphasis on the number of processor cores or VM instances.**

1. Provisioning of Compute Resources (VMs)

- **Providers supply cloud services by signing SLAs with end users.**
- **The SLAs must commit sufficient resources such as CPU, memory, and bandwidth that the user can use for a preset period.**
- **Underprovisioning of resources will lead to broken SLAs and penalties.**
- **Overprovisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider.**
- **Deploying an autonomous system to efficiently provision resources to users is a challenging problem.**
- **The difficulty comes from the unpredictability of consumer demand, software and hardware failures, heterogeneity of services (User may take NaaS,Queue Service, SaaS) , power management (heat dissipation from server), and conflicts in signed SLAs between consumers and service providers.**

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

1. Provisioning of Compute Resources (VMs) (Cont…)

- **Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures.**
- **In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration, and fast recovery from failures.**
- **To deploy VMs, users treat them as physical hosts with customized operating systems for specific applications.**
- **For example, Amazon's EC2** (IaaS service from Amazon) **uses Xen as the virtual machine monitor (VMM). The same VMM is used in IBM's Blue Cloud.**
- **In the EC2 platform, some predefined VM templates are also provided. Users can choose different kinds of VMs from the templates.**
- **IBM's Blue Cloud does not provide any VM templates. In general, any type of VM can run on top of Xen.**
- **Microsoft also applies virtualization in its Azure cloud platform. The provider should offer resource-economic services.**

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

2. Resource Provisioning Methods

- **Demand-Driven method**
- **Event Driven method**
- **Popularity-Driven method**

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

Demand-Driven method

- **This method adds or removes computing instances based on the current utilization level of the allocated resources.**
- **The demand-driven method automatically allocates two Xeon processors for the user application, when the user was using one Xeon processor more than 60 percent of the time for an extended period.**
- **In general, when a resource has surpassed a threshold for a certain amount of time, the scheme increases that resource based on demand. When a resource is below a threshold for a certain amount of time, that resource could be decreased accordingly.** (Defines a range for CPU utilization say for eg: 30% to 70%. if CPU utilization below 30% decreases the CPU capacity. If CPU utilization above 70% increases the CPU capacity)
- **Amazon implements such an auto-scale feature in its EC2 platform.**
- **This method is easy to implement.**
- **Disadvantage: The scheme does not work out right if the workload changes abruptly.**

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

Event Driven method

- This scheme adds or removes machine instances based on a specific time event.
- The scheme works better for seasonal or predicted events such as Christmastime in the West and the Lunar New Year in the East.
- During these events, the number of users grows before the event period and then decreases during the event period.
- This scheme anticipates peak traffic before it happens.
- The method results in a minimal loss of QoS, if the event is predicted correctly. Otherwise, wasted resources are even greater due to events that do not follow a fixed pattern.

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

Popularity-Driven method

- **In this method, the Internet searches for popularity of certain applications and creates the instances by popularity demand.** **(Currently popular applications→ Facebook, Instagram, Twitter)**

- **The scheme anticipates increased traffic with popularity.**

- **Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct. Resources may be wasted if traffic does not occur as expected.**

INTER-CLOUD RESOURCE MANAGEMENT

Resource Provisioning and Platform Deployment

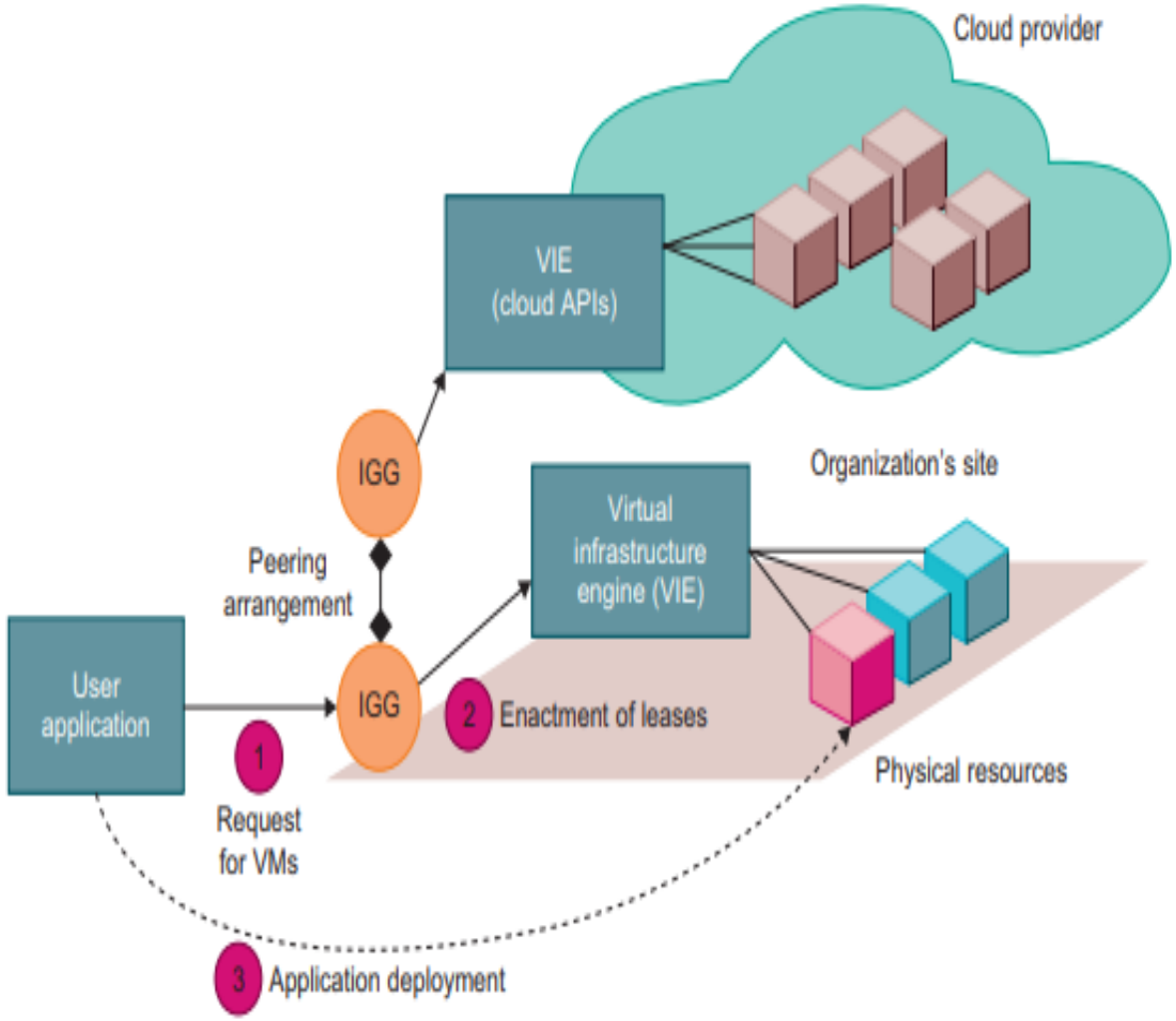
Dynamic Resource Deployment

- (Grid is a distributed high performance computing paradigm that offers various types of resources (like computing, storage, communication) to resource-intensive user tasks.) Grid→ a site which provides some set of resources for the user applications)

- **The cloud uses VMs as building blocks to create an execution environment across multiple resource sites.**

- **The InterGrid-managed infrastructure was developed by a Melbourne University group.**

- **The InterGrid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources.** (Framework/Software designed by Melbourne University group that runs on each grid(organization site/cloud site) that allows users to create VMs on the top of all participating grid, where each grid maintains the set of resources)



Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

INTER-CLOUD RESOURCE MANAGEMENT

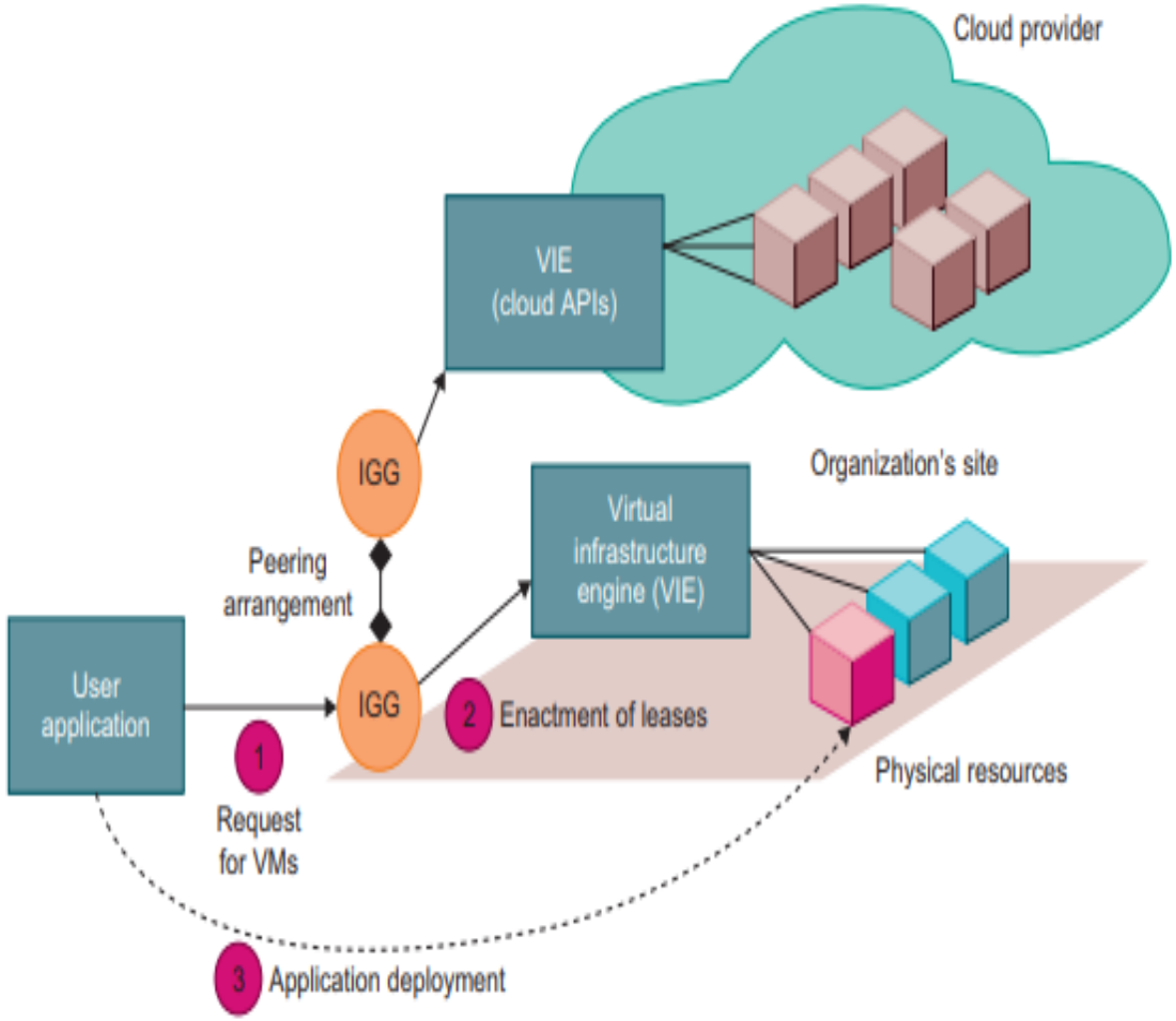
Resource Provisioning and Platform Deployment

Dynamic Resource Deployment

- **Peering arrangements established between gateways enable the allocation of resources from multiple grids to establish the execution environment.**

- **In Figure, a scenario is illustrated by which an intergrid gateway (IGG) allocates resources from a local cluster to deploy applications in three steps: (1) requesting the VMs, (2) enacting the leases (sanctioning), and (3) deploying the VMs as requested.**

- **Under peak demand, this IGG interacts with another IGG that can allocate resources from a cloud computing provider.**

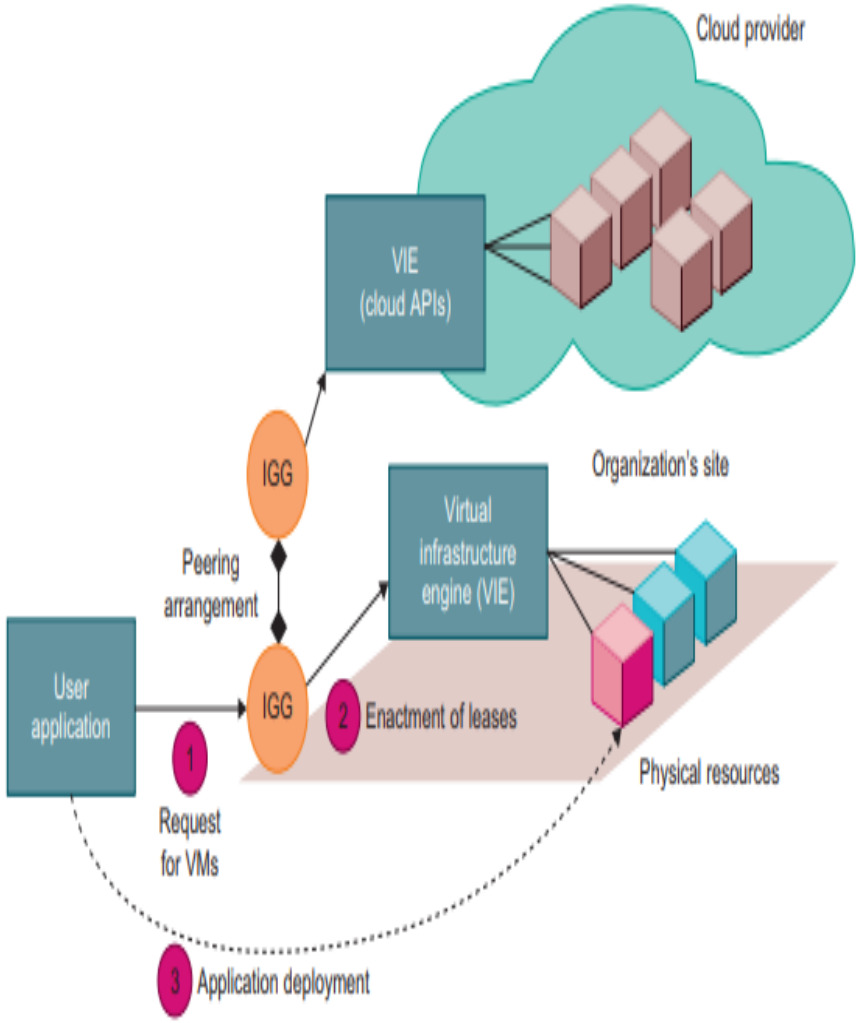


Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

Resource Provisioning and Platform Deployment

Dynamic Resource Deployment

- **A grid has predefined peering arrangements with other grids, which the IGG manages.**
- **Through multiple IGGs, the system coordinates the use of InterGrid resources.**
- **An IGG is aware of the peering terms with other grids, selects suitable grids that can provide the required resources, and replies to requests from other IGGs.**
- **An IGG can also allocate resources from a cloud provider.**
- **The cloud system creates a virtual environment to help users deploy their applications. These applications use the distributed grid resources.**
- **The InterGrid** allocates and provides a **distributed virtual environment (DVE). This is a virtual cluster of VMs that runs isolated from other virtual clusters.**
- **A component called the DVE manager performs resource allocation and management on behalf of specific user applications.**



Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

INTER-CLOUD RESOURCE MANAGEMENT

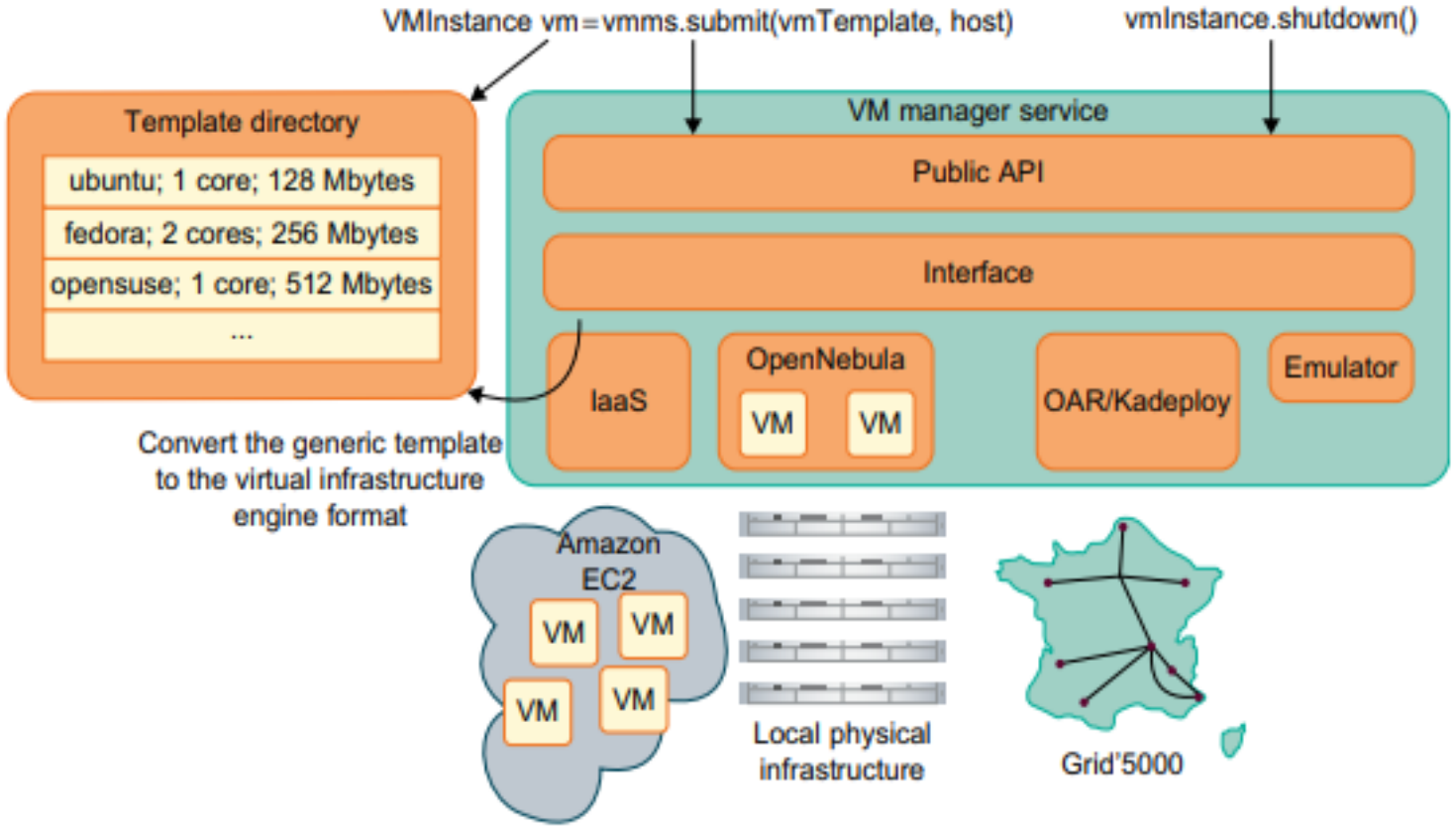
Resource Provisioning and Platform Deployment

Provisioning of Storage Resources

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

- Figure shows the interactions among VM managers for VM creation and management. The managers provide a public API for users to submit and control the VMs.



INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

Independent Service Management

- **Independent service request facilities to execute many unrelated tasks.**

- **Commonly, the APIs provided are some web services that the developer can use conveniently.**

- **In Amazon cloud computing infrastructure, SQS is constructed for providing a reliable communication service between different providers. Even the endpoint does not run while another entity has posted a message in SQS.**

- **By using independent service providers, the cloud applications can run different services at the same time.** (providing data, compute or storage services).

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

Running Third-Party Applications

- **Cloud platforms have to support for building applications by providing applications that are constructed by third-party application providers or programmers.**

- **As current web applications are often provided by using Web 2.0 forms (interactive applications with Ajax), the programming interfaces are different from the traditional programming interfaces such as functions in runtime libraries.**

- **The APIs are often in the form of services.**

- **Web service application engines are often used by programmers for building applications.**

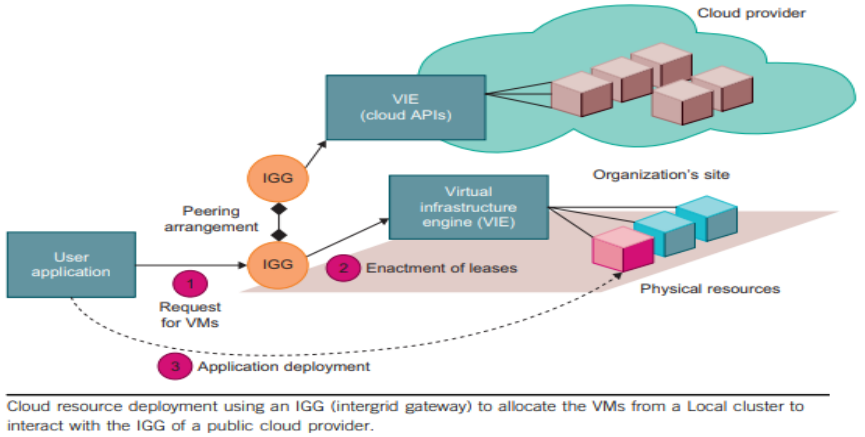
- **As examples, GAE and Microsoft Azure apply their own cloud APIs to get special cloud services.**

- **The WebSphere application engine is deployed by IBM for Blue Cloud. It can be used to develop any kind of web application written in Java.**

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

Virtual Machine Manager



Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

- **The VM manager is the link between the gateway and resources.**
- **The gateway doesn't share physical resources directly, but relies on virtualization technology. Hence, the actual resources it uses are VMs. (VIE runs at each cloud, VMM connects with different VIE.**
- **The manager manage VMs deployed on a set of physical resources.**
- **The VM manager implementation is generic so that it can connect with different VIEs. Typically, VIEs can create and stop VMs on a physical cluster.** (If OpenNebula platform running at one cloud to create VMs, then VIE can communicate with VIE of other cloud running Amazon EC2 platform to create VMs)
- **The Melbourne group has developed managers for OpenNebula, Amazon EC2, and French Grid'5000.**
- **To deploy a VM, the manager needs to use its template**

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

Virtual Machine Templates

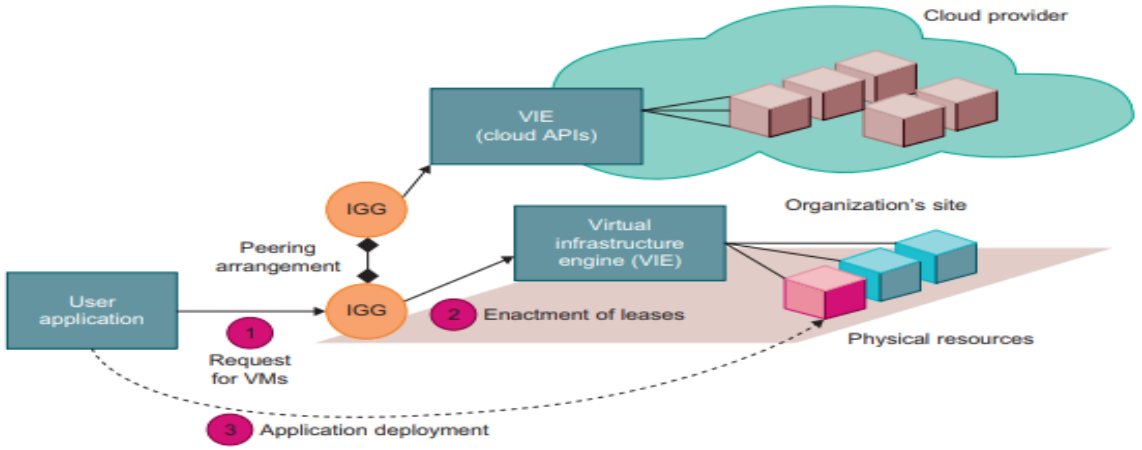
- A VM template is analogous to a computer's configuration and contains a description for a VM with the following static information:
  - **The number of cores or processors to be assigned to the VM**
  - **The amount of memory the VM requires**
  - **The kernel used to boot the VM's operating system**
  - **The disk image containing the VM's file system (Files)**
  - **The price per hour of using a VM**

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

Virtual Machine Templates

- **The gateway administrator provides the VM template information when the infrastructure is set up. The administrator can update, add, and delete templates at any time.**
- **In addition, each gateway in the InterGrid network must agree on the templates to provide the same configuration on each site.**
- **To deploy an instance of a given VM, the VMM generates a descriptor from the template.**
- **This descriptor contains the same fields as the template and additional information related to a specific VM instance.**
- **Typically the additional information includes:**
  - **The disk image that contains the VM's file system**
  - **The address of the physical machine hosting the VM**
  - **The VM's network configuration**



Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

INTER-CLOUD RESOURCE MANAGEMENT

Virtual Machine Creation and Management

Distributed VM Management

INTER-CLOUD RESOURCE MANAGEMENT

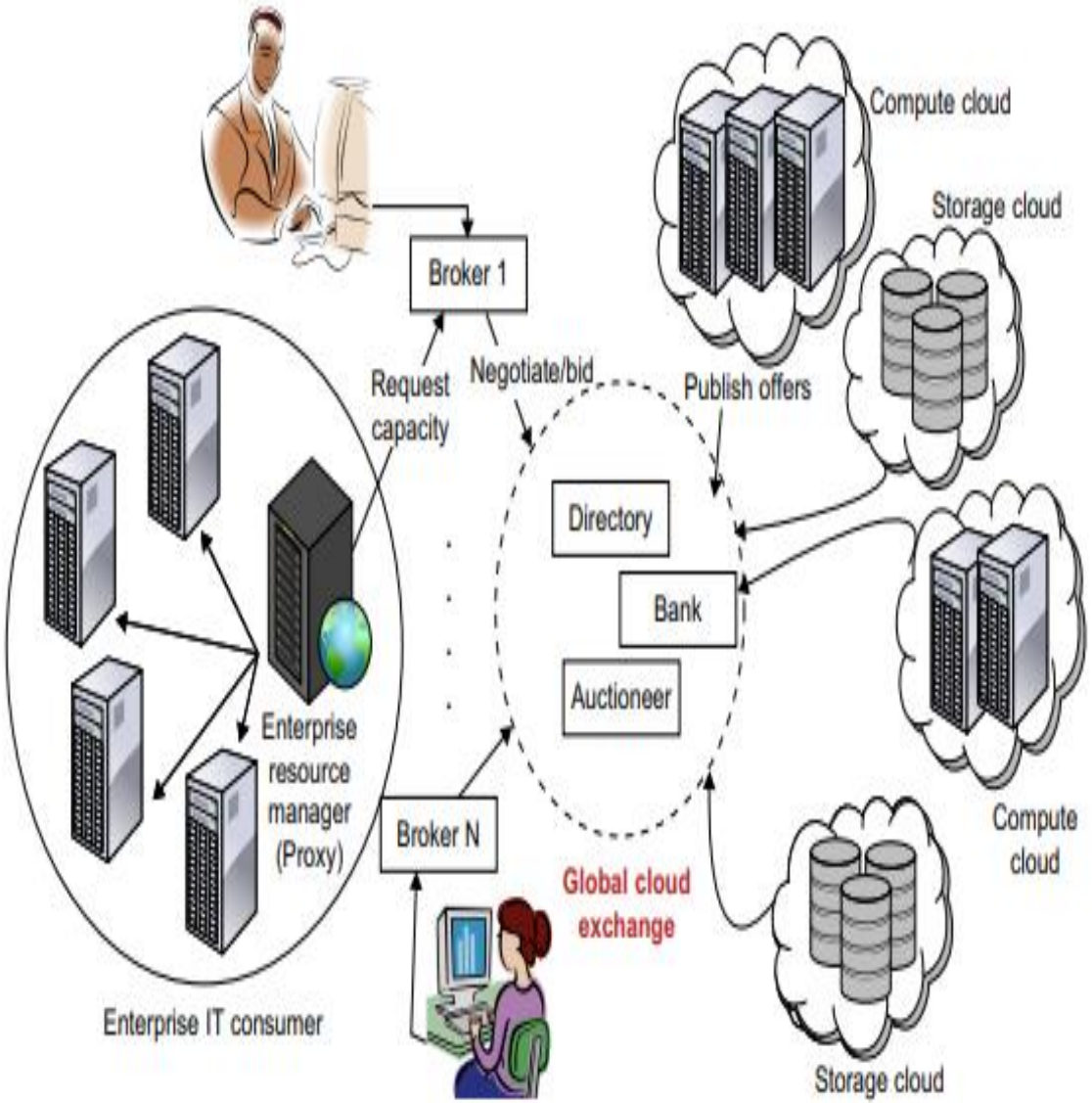
Global Exchange of Cloud Resources

- **In order to support a large number of consumers from around the world, cloud infrastructure providers have established data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures.**
- **For example, Amazon has data centers in the United States (e.g., one on the East Coast and another on the West Coast) and Europe.**
- **However, it is difficult for cloud customers to determine in advance the best location for hosting their services as they may not know the origin of consumers of their services.**
- **Also, SaaS providers may not be able to meet the QoS expectations of their service consumers originating from multiple geographical locations.**
- **This necessitates building mechanisms for seamless federation of data centers of a cloud provider or providers supporting dynamic scaling of applications across multiple domains in order to meet QoS targets of cloud customers.** (Creating of VMs at multiple data centers at multiple places all over the world that satisfies customer QoS),

INTER-CLOUD RESOURCE MANAGEMENT

Global Exchange of Cloud Resources

- Figure shows the high-level components of the Melbourne group's proposed InterCloud architecture.
- In addition, no single cloud infrastructure provider will be able to establish its data centers at all possible locations throughout the world.
- As a result, cloud providers will have difficulty in meeting QoS expectations for all their consumers.
- Hence, they would like to make use of services of multiple cloud infrastructure service providers who can provide better support for their specific consumer needs.
- This necessitates federation of cloud infrastructure service providers for seamless provisioning of services across different cloud providers.



Inter-cloud exchange of cloud resources through brokering.

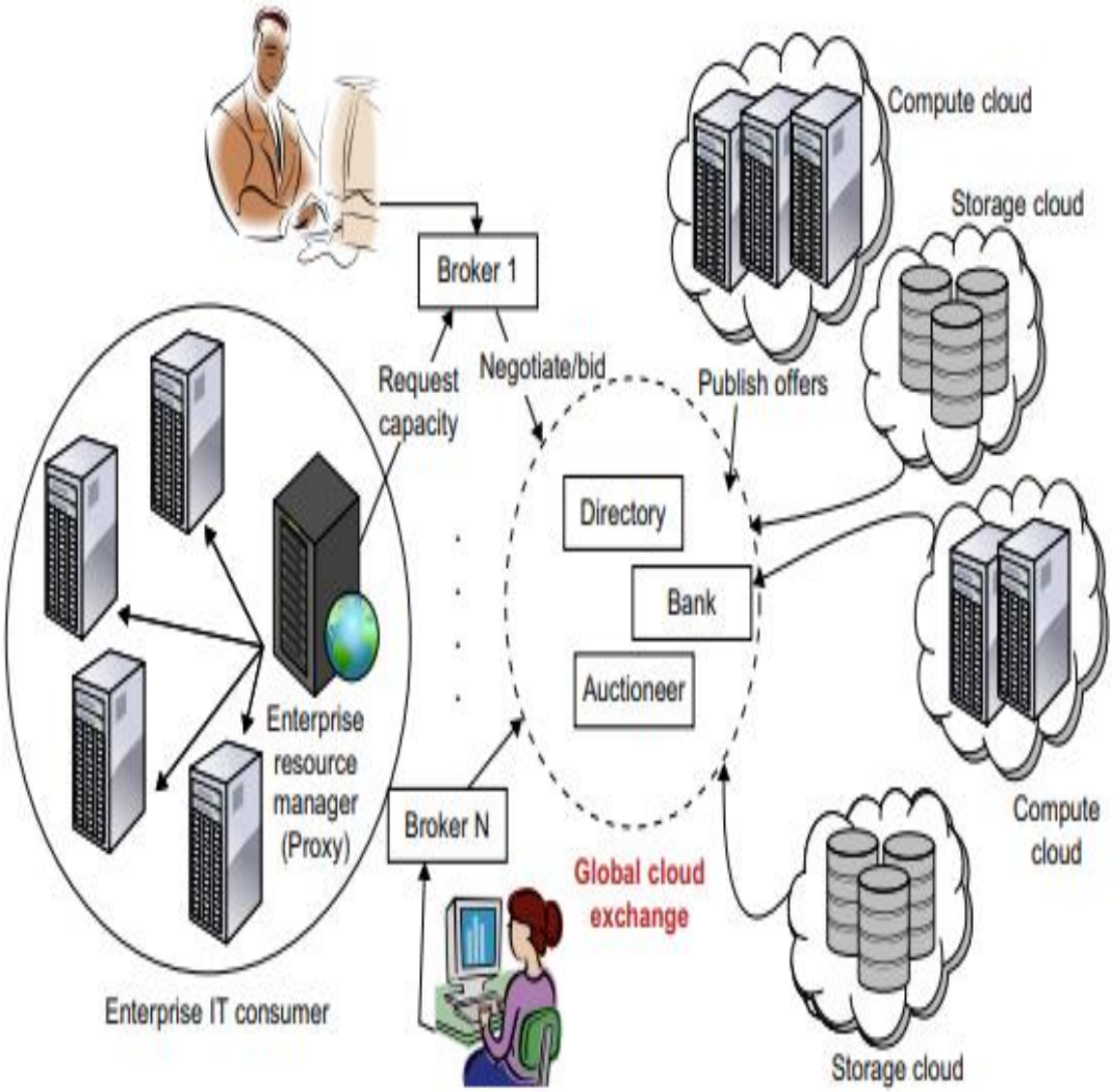
INTER-CLOUD RESOURCE MANAGEMENT

Global Exchange of Cloud Resources

- To realize this, the University of Melbourne has proposed InterCloud architecture supporting brokering and exchange of cloud resources for scaling applications across multiple clouds.

- Cloud providers will be able to dynamically expand or resize their provisioning capability based on sudden spikes in workload demands by leasing available computational and storage capabilities from other cloud service providers; operate as part of a market-driven resource leasing federation.

- They consist of client brokering and coordinator services that support utility-driven federation of clouds: application scheduling, resource allocation, and migration of workloads.



Inter-cloud exchange of cloud resources through brokering.

INTER-CLOUD RESOURCE MANAGEMENT

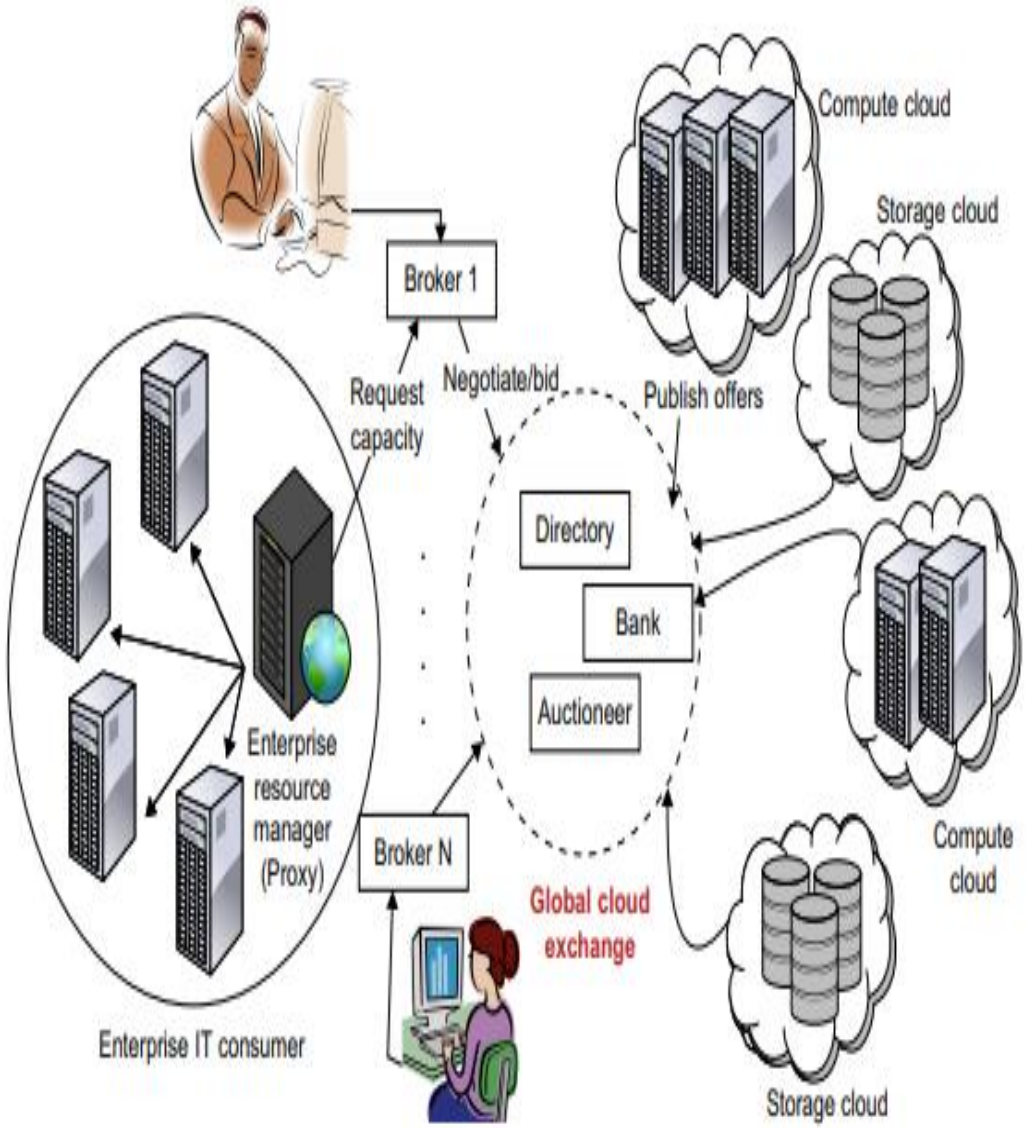
Global Exchange of Cloud Resources

- The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers. It aggregates the infrastructure demands from application brokers and evaluates them against the available supply currently published by the cloud coordinators.

- It supports trading of cloud services based on competitive economic models such as commodity markets and auctions.

- An SLA specifies the details of the service to be provided in terms of metrics agreed upon by all parties, and incentives and penalties for meeting and violating the expectations, respectively.

- The availability of a banking system within the market ensures that financial transactions pertaining to SLAs between participants are carried out in a secure and dependable environment.



Inter-cloud exchange of cloud resources through brokering.

CLOUD SECURITY AND TRUST MANAGEMENT

- **Lacking trust between service providers and cloud users has hindered the universal acceptance of cloud computing as a service on demand.** (Because of lack of trust between providers and users, not universally accepted by users in taking cloud services)

- **For web and cloud services, trust and security become even more demanding, because leaving user applications completely to the cloud providers has faced strong resistance by most users.**

- Cloud platforms become worrisome to **some users for lack of privacy protection, security assurance, and copyright protection.** (Some users worry in taking cloud services because they worry on providers confidentially maintain their data, complete security given to their data with authentication, authorization, accessing method, storing data so that no other users access the data)

- **Trust is a social problem, not a pure technical issue. However, the social problem can be solved with a technical approach.**

Cloud Security Defense Strategies (protection strategies)

- **A healthy** (well defined) **cloud ecosystem is desired to free users from abuses, violence, cheating, hacking, viruses, rumors, pornography** (spoiling), **spam** (moving data to some undermined location), **and privacy** (confidentiality) **and copyright violations.**
- **The security demands at three cloud service models, IaaS, PaaS, and SaaS.**

1. **Basic Cloud Security**
2. **Security Challenges in VMs**
3. **Cloud Defense Methods**
4. **Defense with Virtualization**
5. **Privacy and Copyright Protection**

Cloud Security Defense Strategies

1. Basic Cloud Security

- **Three basic cloud security enforcements are expected.**

➢ **First,** facility **security in data centers demands on-site** security year round. **Biometric readers, CCTV (close-circuit TV), motion detection, and man traps are often deployed.** (on-site security at data center→ Biometric, CCTV, motion detection, man traps)

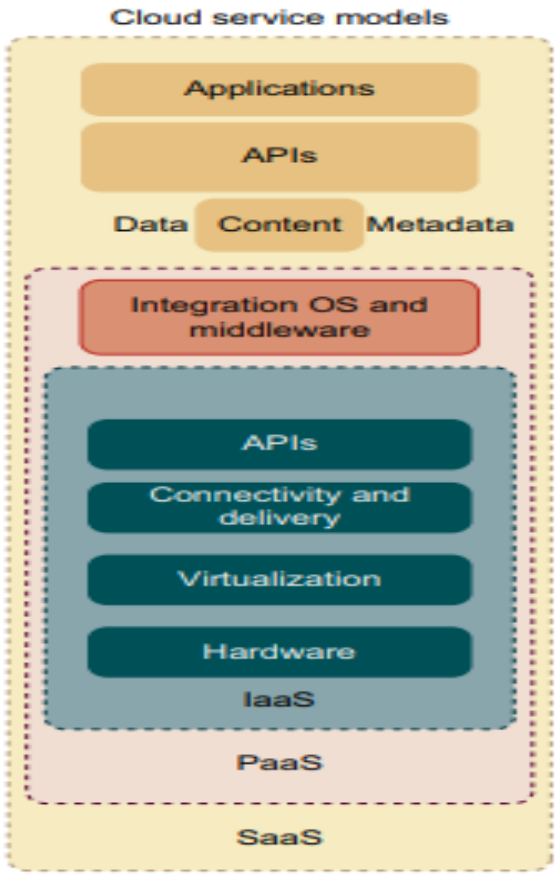
➢ **Second, Network security demands** fault-tolerant external **firewalls, intrusion detection systems** (IDSes), and **third-party vulnerability assessment**.

➢ **Finally, platform security demands SSL and data decryption**, **strict password policies**, and **system trust certification** (providing trust certificate in using platform). (SSL (Secure Sockets Layer) is a security technology used to secure transactions b/w server and browser)

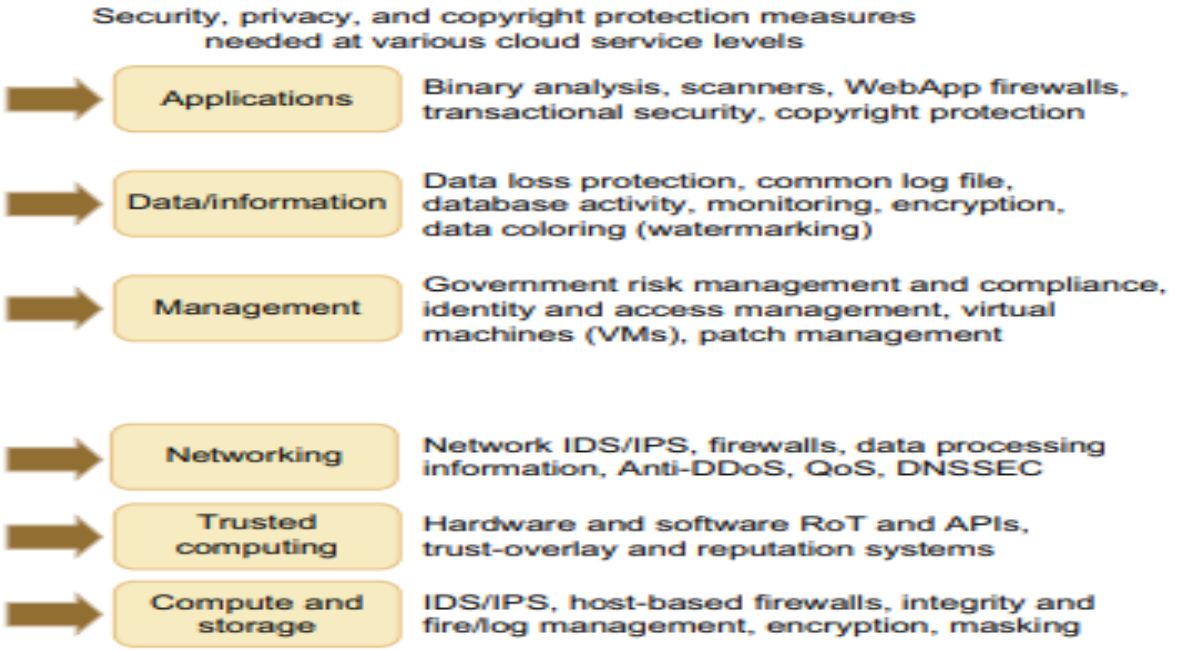
Cloud Security Defense Strategies

Basic Cloud Security

- Below figure shows the mapping of cloud models and special security measures deployed at various cloud operating levels.



(a) Cloud service models



Acronyms:	
IPS:	Intrusion-prevention system
RoT:	Root of trust
DDoS:	Distributed denial of service
DNSSEC:	Domain Name System Security Extensions
QoS:	Quality of service

(b) Security, privacy, and copyright protection measures

Cloud service models on the left (a) and corresponding security measures on the right (b); the IaaS is at the innermost level, PaaS is at the middle level, and SaaS is at the outermost level, including all hardware, software, datasets, and networking resources.

Cloud Security Defense Strategies

Basic Cloud Security

- **Servers in the cloud can be physical machines or VMs.**
- **User interfaces are applied to request services.**
- **A security-aware cloud architecture demands security enforcement.**
- **Malware-based attacks such as network worms, viruses, and DDoS attacks provide intruders unauthorized access to critical information.** (Virus at network level, virus at system level makes intruders unauthorized access to critical information)
- **Thus, security defenses are needed to protect all cluster servers and data centers.**
- **Here are some cloud components that demand special security protection:**
  - **Protection of servers from malicious software attacks such as worms, viruses, and malware.**
  - **Protection of hypervisors or VM monitors from software-based attacks and vulnerabilities.**
  - **Protection of VMs and monitors from service disruption and DoS attacks**
  - **Protection of data and information from theft, corruption, and natural disasters**
  - **Providing authenticated and authorized access to critical data and services**

Cloud Security Defense Strategies

2. Security Challenges in VMs

- **Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms.**

(**Buffer overflow attack**→ attackers feed buffer with some input that overwrites some part of the executable code with attackers code. A **Denial-of-Service** (DoS) attack → is an attack where intruders makes to shut down a machine or network and stop providing service. **Spyware**→ software that enables a user to obtain secret information about another's computer activities by transmitting data covertly from their hard drive. **Malware**→ software that damage, or gain unauthorized access to components of a computer system. A **Rootkit** → software, once installed, it is easy to mask its presence and makes unauthorized privileged access to a computer)

- **In a cloud environment, newer attacks may result from hypervisor malware or VM rootkits.**
- **Another type of attack is the man-in-the-middle attack for VM migrations.**
- In general, **passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.** (kernel data structures→ a data structure created by kernel to store the process information and thread information like current status of process and threads)
- **Program shepherding** (guiding) **can be applied to control and verify code execution.**
- **Other defense technologies include using the RIO dynamic optimization infrastructure** (Runtime Introspection (Self examination) and Optimization), or **VMware's vSafe** and **vShield tools**, and **Intel vPro technology.**
- Others apply a **hardened OS environment** or use **isolated execution** and **sandboxing.** A sandbox is **a tightly controlled environment where programs can be run**. (For example, your web browser essentially runs web pages you visit in a sandbox→ Sandbox makes web browser to run on a tight environment so as to display web pages without accessing any of our system components)).

Cloud Security Defense Strategies

3. Cloud Defense Methods

- **With virtualization, a single physical machine can be divided or partitioned into multiple VMs.**
- **This provides each VM with better security isolation and each partition is protected from DoS attacks by other partitions.**
- **Security attacks in one VM are isolated and contained from affecting the other VMs.**
- **VM failures do not propagate to other VMs.**
- **Malicious** (cruel) **intrusions may destroy valuable hosts, networks, and storage resources. Internet anomalies found in routers, gateways, and distributed hosts may stop cloud services.** (Some malfunctioning at the internet level→ routers, gateways may stop cloud services)
- **Trust negotiation is often done at the SLA level.**
- **Public Key Infrastructure (PKI) services could be augmented with data-center reputation systems.** (a public key and a private key. … By using a two-key encryption system, PKI secures sensitive electronic information as it is passed back and forth between two parties, and provides each party with a key to encrypt and decrypt the digital data.)
- **It is harder to establish security in the cloud because all data and software are shared by default.**

Cloud Security Defense Strategies

3. Cloud Defense Methods

Table lists eight protection schemes to secure public clouds and data centers.

Physical and Cyber Security Protection at Cloud/Data Centers	
Protection Schemes	**Brief Description and Deployment Suggestions**
Secure data centers and computer buildings	Choose hazard-free location, enforce building safety. Avoid windows, keep buffer zone around the site, bomb detection, camera surveillance, earthquake-proof, etc.
Use redundant utilities at multiple sites	Multiple power and supplies, alternate network connections, multiple databases at separate sites, data consistency, data watermarking, user authentication, etc.
Trust delegation and negotiation	Cross certificates to delegate trust across PKI domains for various data centers, trust negotiation among certificate authorities (CAs) to resolve policy conflicts
Worm containment and DDoS defense	Internet worm containment and distributed defense against DDoS attacks to secure all data centers and cloud platforms
Reputation system for data centers	Reputation system could be built with P2P technology; one can build a hierarchy of reputation systems from data centers to distributed file systems
Fine-grained file access control	Fine-grained access control at the file or object level; this adds to security protection beyond firewalls and IDSes
Copyright protection and piracy prevention	Piracy prevention achieved with peer collusion prevention, filtering of poisoned content, nondestructive read, alteration detection, etc.
Privacy protection	Uses double authentication, biometric identification, intrusion detection and disaster recovery, privacy enforcement by data watermarking, data classification, etc.

Cloud Security Defense Strategies

4. Defense with Virtualization

- **The VM is decoupled from the physical hardware.**
- **The entire VM can be represented as a software component and can be regarded as binary or digital data.**
- **The VM can be saved, cloned, encrypted, moved, or restored with ease.**
- **VMs enable HA and faster disaster recovery.**
- **Multiple IDS VMs can be deployed at various resource sites including data centers.** (distributed IDS (dIDS) consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other and monitors intrusion at network)
- **Security policy conflicts in designing distributed intrusion detection systems (DIDSes) must be resolved at design time and updated periodically.** (Security policy conflicts while communication among IDS VMs)

Cloud Security Defense Strategies

5. Privacy and Copyright Protection

- With shared files and data sets, privacy, security, and copyright data could be compromised in a cloud computing environment.
- Users desire to work in a software environment that provides many useful tools to build cloud applications over large data sets.
- Google’s platform essentially applies in-house software to protect resources.
- The Amazon EC2 applies HMEC and X.509 certificates in securing resources.

Cloud Security Defense Strategies

5. Privacy and Copyright Protection (Cont....)

- **Here are several security features desired in a secure cloud:**
  - **Dynamic web services with full support from secure web technologies**
  - **Established trust between users and providers through SLAs and reputation systems**
  - **Effective user identity management and data-access management**
  - **Single sign-on and single sign-off to reduce security enforcement overhead** (Single sign-on (SSO) is a user authentication service that permits a user to use one set of login credentials to access multiple applications. For Eg: Gmail account with which can access Google Docs, Spread Sheets, Youtube, Swayam portal)
  - **Auditing and copyright compliance through proactive enforcement** (Auditing periodically)
  - **Shifting of control of data operations from the client environment to cloud providers**
  - **Protection of sensitive and regulated information in a shared environment**

Distributed Intrusion/Anomaly Detection