PRATHYUSHA

ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

REGULATION 2017

IV YEAR - VII SEMESTER

CS8791- CLOUD COMPUTING

CS8791 CLOUD COMPUTING L T P C 3 0 0 3

OBJECTIVES:

- To understand the concept of cloud computing.
- To appreciate the evolution of cloud from the existing technologies.
- To have knowledge on the various issues in cloud computing.
- To be familiar with the lead players in cloud.
- To appreciate the emergence of cloud as the next generation computing paradigm

UNIT I INTRODUCTION 9

Introduction to Cloud Computing – Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristics – Elasticity in Cloud – On-demand Provisioning.

UNIT II CLOUD ENABLING TECHNOLOGIES 10

Service Oriented Architecture – REST and Systems of Systems – Web Services – Publish-Subscribe Model – Basics of Virtualization – Types of Virtualization – Implementation Levels of Virtualization – Virtualization Structures – Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices –Virtualization Support and Disaster Recovery.

UNIT III CLOUD ARCHITECTURE, SERVICES AND STORAGE 8

Layered Cloud Architecture Design – NIST Cloud Computing Reference Architecture – Public, Private and Hybrid Clouds - IaaS – PaaS – SaaS – Architectural Design Challenges – Cloud Storage – Storage-as-a-Service – Advantages of Cloud Storage – Cloud Storage Providers – S3.

UNIT IV RESOURCE MANAGEMENT AND SECURITY IN CLOUD 10

Inter Cloud Resource Management – Resource Provisioning and Resource Provisioning Methods – Global Exchange of Cloud Resources – Security Overview – Cloud Security Challenges – Software-as-a-Service Security – Security Governance – Virtual Machine Security – IAM – Security Standards.

UNIT V CLOUD TECHNOLOGIES AND ADVANCEMENTS 8

Hadoop – MapReduce – Virtual Box -- Google App Engine – Programming Environment for Google App Engine —- Open Stack – Federation in the Cloud – Four Levels of Federation – Federated Services and Applications – Future of Federation.

TOTAL: 45 PERIODS

OUTCOMES: On Completion of the course, the students should be able to:

- Articulate the main concepts, key technologies, strengths and limitations of cloud computing.
- Learn the key and enabling technologies that help in the development of cloud.
- Develop the ability to understand and use the architecture of compute and storage cloud, service and delivery models.
- Explain the core issues of cloud computing such as resource management and security.
- Be able to install and use current cloud technologies.
- Evaluate and choose the appropriate technologies, algorithms and approaches for implementation and use of cloud.

TEXT BOOKS: 1. Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012. 2. Rittinghouse, John W., and James F. Ransome, —Cloud Computing: Implementation, Management and Security‖, CRC Press, 2017.

REFERENCES: 1. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, —Mastering Cloud Computing‖, Tata Mcgraw Hill, 2013. 2. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing - A Practical Approach‖, Tata Mcgraw Hill, 2009. 3. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice)‖, O'Reilly,

PRATHYUSHA ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS8791 CLOUD COMPUTING

IV YEAR – VII SEMESTER

LECTURE NOTES - UNIT I

Prepared by

SOUMYA.T.R

CSE Department

UNIT I INTRODUCTION

Introduction to Cloud Computing – Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristics – Elasticity in Cloud – On-demand Provisioning.

INTRODUCTION

Introduction to Cloud Computing

Cloud computing is a virtualization-based technology that allows us to create, configure, and customize applications via an internet connection. The cloud technologyincludes a development platform, hard disk, software application, and database.

What is Cloud Computing?

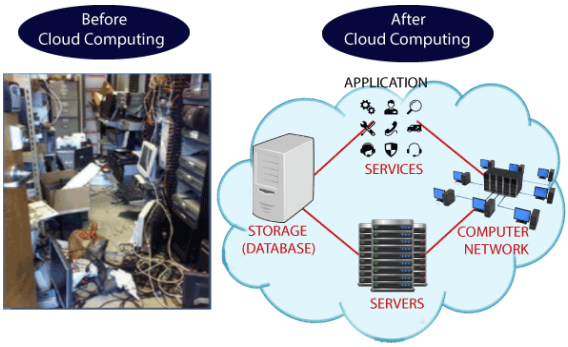
The term cloud refers to a network or the internet. It is a technology that uses remote servers on the internet to store, manage, and access data online rather than local drives. The data can be anything such as files, images, documents, audio, video, and more.

There are the following operations that we can do using cloud computing:

- Developing new applications and services
- Storage, back up, and recovery of data
- Hosting blogs and websites
- Delivery of software on demand
- Analysis of data
- Streaming videos and audios

Why Cloud Computing?

Small as well as large IT companies, follow the traditional methods to provide the IT infrastructure. That means **for any IT company, we need a Server Room that is the basic need of IT companies**.



Advantages of cloud computing

1) Back-up and restore data

Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.

2) Improved collaboration

Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.

3) Excellent accessibility

Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.

4) Low maintenance cost

Cloud computing reduces both hardware and software maintenance costs for organizations.

5) Mobility

Cloud computing allows us to easily access all cloud data via mobile.

6) IServices in the pay-per-use model

Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.

7) Unlimited storage capacity

Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, video, etc. in one place.

8) Data security

Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

Disadvantages of cloud computing

1) Internet Connectivity

As you know, in cloud computing, every data (image, audio, video, etc.) is stored on the cloud, and we access these data through the cloud by using the internet connection. If you do not have good internet connectivity, you cannot access these data. However, we have no any other way to access data from the cloud.

2) Vendor lock-in

Vendor lock-in is the biggest disadvantage of cloud computing. Organizations may face problems when transferring their services from one vendor to another. As different vendors provide different platforms, that can cause difficulty moving from one cloud to another.

3) Limited Control

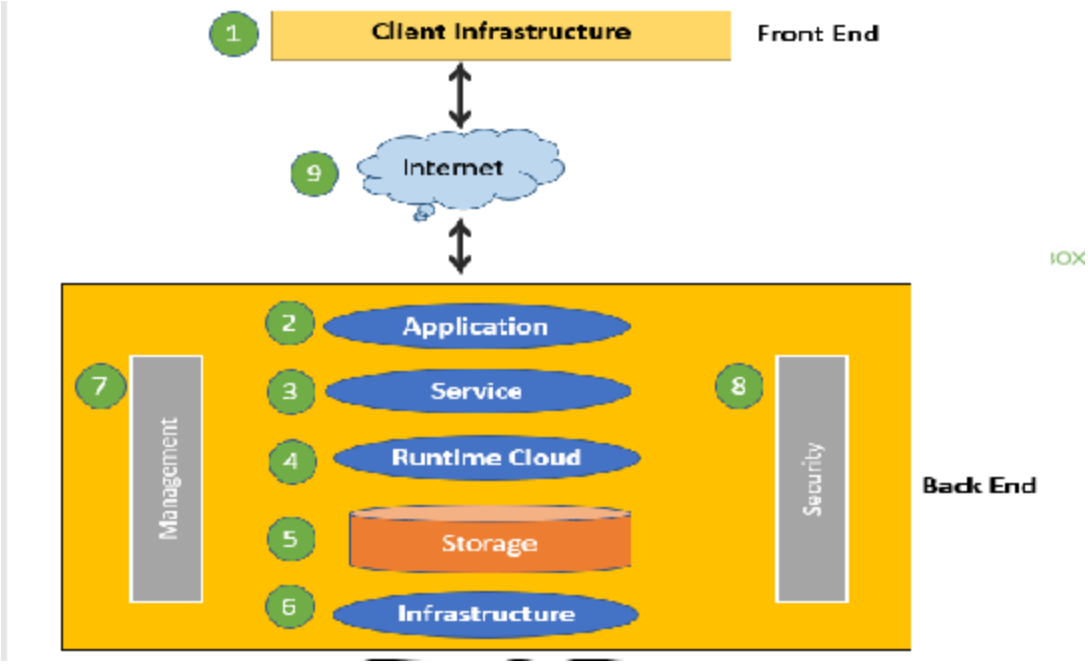
As we know, cloud infrastructure is completely owned, managed, and monitored by the service provider, so the cloud users have less control over the function and execution of services within a cloud infrastructure.

4) Security

Although cloud service providers implement the best security standards to store important information. But, before adopting cloud technology, you should be aware that

you will be sending all your organization's sensitive information to a third party, i.e., a cloud computing service provider. While sending the data on the cloud, there may be a chance that your organization's information is hacked by Hackers.

Cloud Computing Architecture



The Architecture of Cloud computing contains many different components. It includes Client infrastructure, applications, services, runtime clouds, storage spaces, management, and security. These are all the parts of a Cloud computing architecture.

Front End:

The client uses the front end, which contains a client-side interface and application. Both of these components are important to access the Cloud computing platform. The front end includes web servers (Chrome, Firefox, Opera, etc.), clients, and mobile devices.

Back End:

The backend part helps you manage all the resources needed to provide Cloud computing services. This Cloud architecture part includes a security mechanism, a large amount of data storage, servers, virtual machines, traffic control mechanisms, etc.

1. Client Infrastructure:

Client Infrastructure is a front-end component that provides a GUI. It helps users to interact with the Cloud.

2. Application:

The application can be any software or platform which a client wants to access.

3. Service:

The service component manages which type of service you can access according to the client's requirements.

Three Cloud computing services are:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

4. Runtime Cloud:

Runtime cloud offers the execution and runtime environment to the virtual machines.

5. Storage:

Storage is another important Cloud computing architecture component. It provides a large amount of storage capacity in the Cloud to store and manage data.

6. Infrastructure:

It offers services on the host level, network level, and application level. Cloud infrastructure includes hardware and software components like servers, storage, network devices, virtualization software, and various other storage resources that are needed to support the cloud computing model.

7. Management:

This component manages components like application, service, runtime cloud, storage, infrastructure, and other security matters in the backend. It also establishes coordination between them.

8. Security:

Security in the backend refers to implementing different security mechanisms for secure Cloud systems, resources, files, and infrastructure to the end-user.

9. Internet:

Internet connection acts as the bridge or medium between frontend and backend. It allows you to establish the interaction and communication between the frontend and backend.

Virtualization and Cloud Computing

The main enabling technology for Cloud Computing is Virtualization. Virtualization is the partitioning of a single physical server into multiple logical servers. Once the physical server is divided, each logical server behaves like a physical server and can run an operating system and applications independently. Many popular companies like VMware and Microsoft provide virtualization services. Instead of using your PC for storage and computation, you can use their virtual servers. They are fast, cost-effective, and less time-consuming.

For software developers and testers, virtualization comes in very handy. It allows developers to write code that runs in many different environments for testing.

Virtualization is mainly used for three main purposes: 1) Network Virtualization, 2) Server Virtualization, and 3) Storage Virtualization

Network Virtualization: It is a method of combining the available resources in a network by splitting up the available bandwidth into channels. Each channel is independent of others and can be assigned to a specific server or device in real time.

Storage Virtualization: It 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 commonly used in storage area networks (SANs).

Server Virtualization: Server virtualization is the masking of server resources like processors, RAM, operating system, etc., from server users. Server virtualization intends to increase resource sharing and reduce the burden and complexity of computation from users.

Virtualization is the key to unlock the Cloud system, what makes virtualization so important for the cloud is that it decouples the software from the hardware. For example, PCs can use virtual memory to borrow extra memory from the hard disk. Usually, a hard disk has a lot more space than memory. Although virtual disks are slower than real memory, if managed properly, the substitution works perfectly. Likewise, there is software that can imitate an entire computer, which means 1 computer can perform the functions equals to 20 computers.

EVOLUTION OF DISTRIBUTED COMPUTING

Grids enable access to shared computing power and storage capacity from your desktop.

Clouds enable access to leased computing power and storage capacity from your desktop.

- Grids are an **open source** technology. Resource users and providers alike can understand and contribute to the management of their grid
- Clouds are a **proprietary** technology. Only the resource provider knows exactly how their cloud manages data, job queues, security requirements and so on.
- The concept of grids was proposed in 1995. The Open science grid (OSG) started in 1995 The EDG (European Data Grid) project began in 2001.
- In the late 1990`s Oracle and EMC offered early private cloud solutions . However the term cloud computing didn't gain prominence until 2007.

SCALABLE COMPUTING OVER THE INTERNET

Instead of using a centralized computer to solve computational problems, a parallel and

distributed computing system uses multiple computers to solve large-scale problems over the Internet. Thus, distributed computing becomes data-intensive and network-centric.

The Age of Internet Computing

- high-performance computing (HPC) applications is no longer optimal for measuring system performance
- The emergence of computing clouds instead demands high-throughput computing (HTC) systems built with parallel and distributed computing technologies
- We have to upgrade data centers using fast servers, storage systems, and high-bandwidth networks.

The Platform Evolution

- From 1950 to 1970, a handful of mainframes, including the IBM 360 and CDC 6400
- From 1960 to 1980, lower-cost minicomputers such as the DEC PDP 11 and VAX Series
- From 1970 to 1990, we saw widespread use of personal computers built with VLSI microprocessors.
- From 1980 to 2000, massive numbers of portable computers and pervasive devices appeared in both wired and wireless applications
- Since 1990, the use of both HPC and HTC systems hidden in clusters, grids, or Internet clouds has proliferated

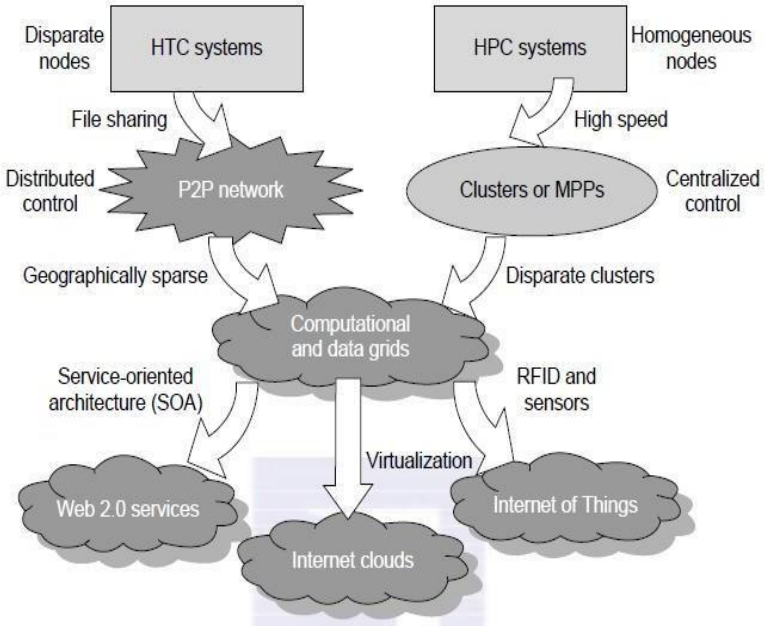


FIGURE 1.1

Evolutionary trend toward parallel, distributed, and cloud computing with clusters, MPPs, P2P networks, grids, clouds, web services, and the Internet of Things.

On the HPC side, supercomputers (massively parallel processors or MPPs) are gradually replaced by clusters of cooperative computers out of a desire to share computing resources. The cluster is often a collection of homogeneous compute nodes that are physically connected in close range to one another.

- On the HTC side, peer-to-peer (P2P) networks are formed for distributed file sharing and content delivery applications. A P2P system is built over many client machines (a concept we will discuss further in Chapter 5). Peer machines are globally distributed in nature. P2P, cloud computing, and web service platforms are more focused on HTC applications than on HPC applications. Clustering and P2P technologies lead to the development of computational grids or data grids.
- For many years, HPC systems emphasize the raw speed performance. The speed of HPC systems has increased from Gflops in the early 1990s to now Pflops in 2010.
- The development of market-oriented high-end computing systems is undergoing a strategic change from an HPC paradigm to an HTC paradigm. This HTC paradigm pays more attention to high-flux computing. The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously. The performance goal thus shifts to measure high throughput or the number of tasks completed per unit of time. HTC technology needs to not only improve in terms of batch processing speed, but also address the acute problems of cost, energy savings, security, and reliability at many data and enterprise computing centers.
- Advances in virtualization make it possible to see the growth of Internet clouds as a new computing paradigm. The maturity of radio-frequency identification (RFID), Global Positioning System (GPS), and sensor technologies has triggered the development of the Internet of Things (IoT). These new paradigms are only briefly introduced here.
- The high-technology community has argued for many years about the precise definitions of centralized computing, parallel computing, distributed computing, and cloud computing. In general, distributed computing is the opposite of centralized computing. The field of parallel computing overlaps with distributed computing to a great extent, and cloud computing overlaps with distributed, centralized, and parallel computing.

Terms.

Centralized computing

This is a computing paradigm by which all computer resources are centralized in one physical system. All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS. Many data centers and supercomputers are centralized systems, but they are used in parallel, distributed, and cloud computing applications.

- **Parallel computing**

In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory. Inter processor communication is accomplished through shared memory or via message passing. Acomputer system capable of parallel computing is commonly known as a parallel computer. Programs running in a parallel computer are called parallel programs. The process of writing parallel programs is often referred to as parallel programming.

- **Distributed computing** This is a field of computer science/engineering that studies distributed systems. A distributed system consists of multiple autonomous computers, each having its own private memory, communicating through a computer network. Information exchange in a distributed system is accomplished through message passing. A computer program that runs in a distributed system is known as a distributed program. The process of writing distributed programs is referred to as distributed programming.
- **Cloud computing** An Internet cloud of resources can be either a centralized or a distributed computing system. The cloud applies parallel or distributed computing, or both. Clouds can be built with physical or virtualized resources over large data centers that are centralized or distributed. Some authors consider cloud computing to be a form of utility computing or service computing . As an alternative to the preceding terms, some in the high-tech community prefer the term concurrent computing or concurrent programming. These terms typically refer to the union of parallel computing and distributing computing, although biased practitioners may interpret them differently.
- **Ubiquitous computing** refers to computing with pervasive devices at any place and time using wired or wireless communication. The Internet of Things (IoT) is a networked connection of everyday objects including computers, sensors, humans, etc. The IoT is supported by Internet clouds to achieve ubiquitous computing with any object at any

place and time. Finally, the term Internet computing is even broader and covers all computing paradigms over the Internet. This book covers all the aforementioned computing paradigms, placing more emphasis on distributed and cloud computing and their working systems, including the clusters, grids, P2P, and cloud systems.

Internet of Things

- The traditional Internet connects machines to machines or web pages to web pages. The concept of the IoT was introduced in 1999 at MIT .
- The IoT refers to the networked interconnection of everyday objects, tools, devices, or computers. One can view the IoT as a wireless network of sensors that interconnect all things in our daily life.
- It allows objects to be sensed and controlled remotely across existing network infrastructure

SYSTEM MODELS FOR DISTRIBUTED AND CLOUD COMPUTING

- Distributed and cloud computing systems are built over a large number of autonomous computer nodes.
- These node machines are interconnected by SANs, LANs, or WANs in a hierarchical manner. With today's networking technology, a few LAN switches can easily connect hundreds of machines as a working cluster.
- A WAN can connect many local clusters to form a very large cluster of clusters.

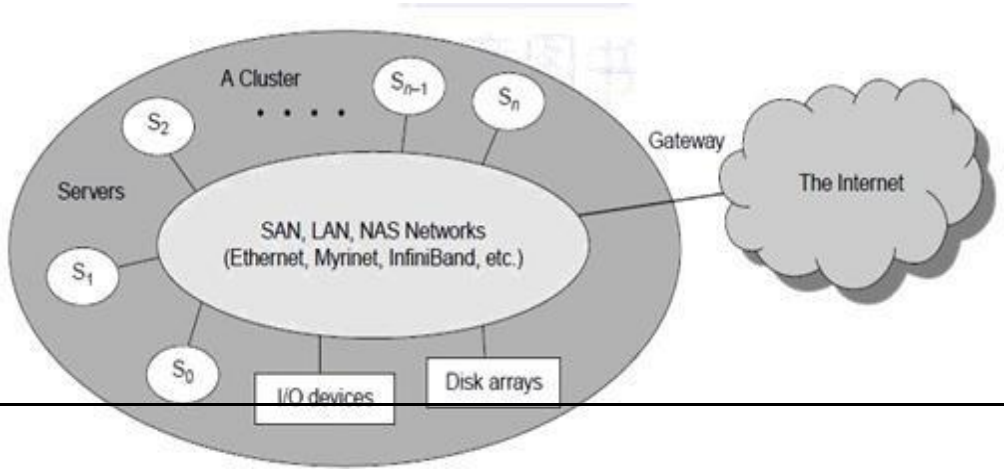
Clusters of Cooperative Computers

A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.

- In the past, clustered computer systems have demonstrated impressive results in handling heavy workloads with large data sets.

Cluster Architecture

cluster built around a low-latency, high bandwidth interconnection network. This network



can be as simple as a SAN or a LAN (e.g., Ethernet).

Figure 1.2 Clusters of Servers

Figure 1.2shows the architecture of a typical server cluster built around a low-latency, high bandwidth interconnection network. This network can be as simple as a SAN (e.g., Myrinet) or a LAN (e.g., Ethernet).

- To build a larger cluster with more nodes, the interconnection network can be built with multiple levels of Gigabit Ethernet, or InfiniBand switches.
- Through hierarchical construction using a SAN, LAN, or WAN, one can build scalable clusters with an increasing number of nodes. The cluster is connected to the Internet via a virtual private network (VPN) gateway.
- The gateway IP address locates the cluster. The system image of a computer is decided by the way the OS manages the shared cluster resources.

Most clusters have loosely coupled node computers. All resources of a server node are managed by their own OS. Thus, most clusters have multiple system images as a result of having many autonomous nodes under different OS control.

Single-System Image(SSI)

- Ideal cluster should merge multiple system images into a single-system image (SSI).
- Cluster designers desire a cluster operating system or some middleware to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.

An SSI is an illusion created by software or hardware that presents a collection of resources as one integrated, powerful resource. SSI makes the cluster appear like a single machine to the user. A cluster with multiple system images is nothing but a collection of independent computers.

Hardware, Software, and Middleware Support

- Clusters exploring massive parallelism are commonly known as MPPs. Almost all HPC clusters in the Top 500 list are also MPPs.
- The building blocks are computer nodes (PCs, workstations, servers, or SMP), special communication software such as PVM, and a network interface card in each computer node.

Most clusters run under the Linux OS. The computer nodes are interconnected by a high-bandwidth network (such as Gigabit Ethernet, Myrinet, InfiniBand, etc.). Special cluster middleware supports are needed to create SSI or high availability (HA). Both sequential and parallel applications can run on the cluster, and special parallel environments are needed to facilitate use of the cluster resources. For example, distributed memory has multiple images.

Users may want all distributed memory to be shared by all servers by forming distributed shared memory (DSM). Many SSI features are expensive or difficult to achieve at various cluster operational levels. Instead of achieving SSI, many clusters are loosely coupled machines. Using virtualization, one can build many virtual clusters dynamically, upon user demand.

Cloud Computing over the Internet

- A cloud is a pool of virtualized computer resources.
- A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications.
- A cloud allows workloads to be deployed and scaled out quickly through rapid provisioning of virtual or physical machines.
- The cloud supports redundant, self-recovering, highly scalable programming models that allow workloads to recover from many unavoidable hardware/software failures.
- Finally, the cloud system should be able to monitor resource use in real time to enable rebalancing of allocations when needed.

a. Internet Clouds

- Cloud computing applies a virtualized platform with elastic resources on demand by provisioning hardware, software, and data sets dynamically .The idea is to move desktop computing to a service-oriented platform using server clusters and huge databases at data centers.
- Cloud computing leverages its low cost and simplicity to benefit both users and providers.
- Machine virtualization has enabled such cost-effectiveness. Cloud computing intends to satisfy many user applications simultaneously.

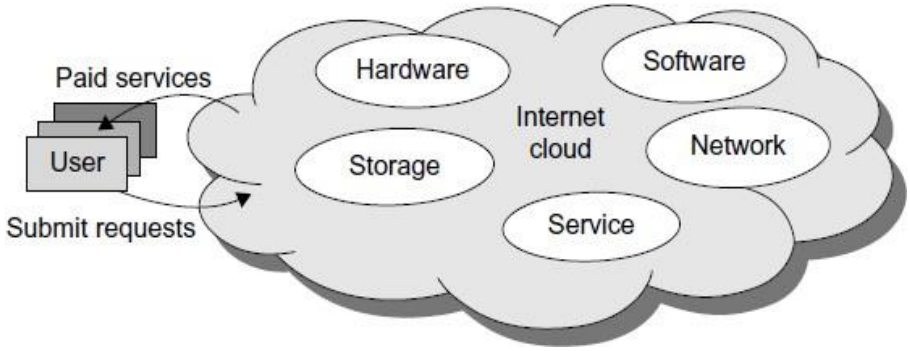


Figure 1.3 Internet Cloud

b. The Cloud Landscape

- The cloud ecosystem must be designed to be secure, trustworthy, and dependable. Some computer users think of the cloud as a centralized resource pool. Others consider the cloud to be a server cluster which practices distributed computing over all the servers Traditionally, a distributed computing system tends to be owned and operated by an autonomous administrative domain (e.g., a research laboratory or company) for on-premises computing needs.
- Cloud computing as an on-demand computing paradigm resolves or relieves us from these problems.

Three Cloud service Model in a cloud landscape

Infrastructure as a Service (IaaS)

- This model puts together infrastructures demanded by users—namely servers, storage, networks, and the data center fabric.
- The user can deploy and run on multiple VMs running guest OS on specific applications.
- The user does not manage or control the underlying cloud infrastructure, but can specify when to request and release the needed resources.

Platform as a Service (PaaS)

• This model enables the user to deploy user-built applications onto a virtualized cloud platform. PaaS includes middleware, databases, development tools, and some runtime support such as Web 2.0 and Java.

• The platform includes both hardware and software integrated with specific programming interfaces.

• The provider supplies the API and software tools (e.g., Java, Python, Web 2.0, .NET). The user is freed from managing the cloud infrastructure.

Software as a Service (SaaS)

• This refers to browser-initiated application software over thousands of paid cloud customers. The SaaS model applies to business processes, industry applications, consumer relationship management (CRM), enterprise resources planning (ERP), human resources (HR), and collaborative applications. On the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are rather low, compared with conventional hosting of user applications.

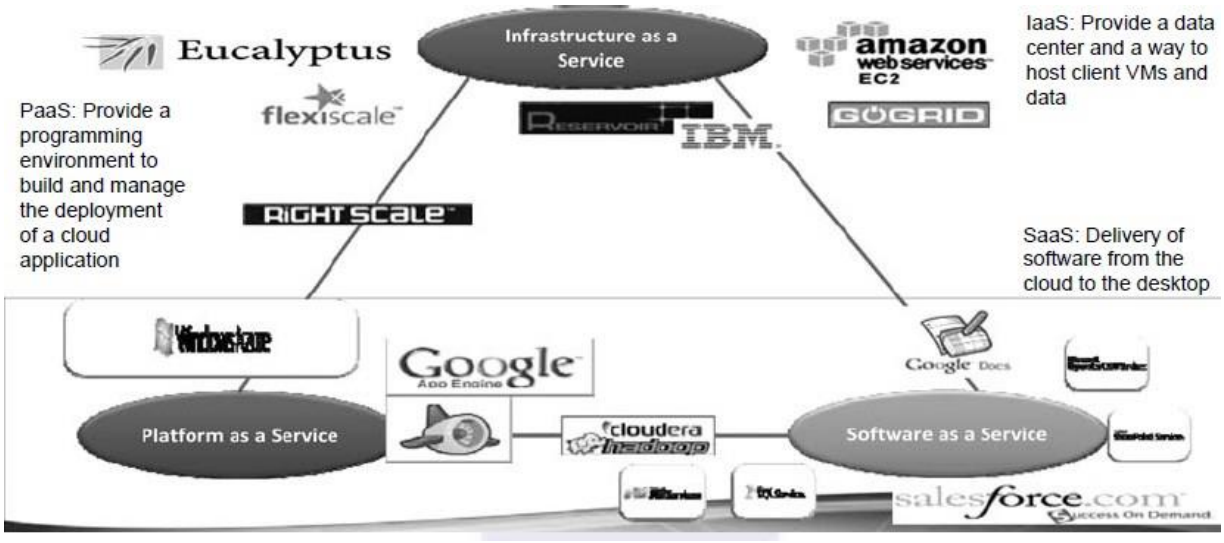


Figure 1.4 The Cloud Landscape in an application

Internet clouds offer four deployment modes: private, public, managed, and hybrid . These modes demand different levels of security implications. The different SLAs imply that the security responsibility is shared among all the cloud providers, the cloud resource consumers, and the third party cloud-enabled software providers. Advantages of cloud computing have been advocated by many IT experts, industry leaders, and computer science researchers.

Reasons to adapt the cloud for upgraded Internet applications and web services:

1. Desired location in areas with protected space and higher energy efficiency
2. Sharing of peak-load capacity among a large pool of users, improving overall utilization
3. Separation of infrastructure maintenance duties from domain-specific application development
4. Significant reduction in cloud computing cost, compared with traditional computing paradigms
5. Cloud computing programming and application development
6. Service and data discovery and content/service distribution
7. Privacy, security, copyright, and reliability issues
8. Service agreements, business models, and pricing policies

- Cloud computing is using the internet to access someone else's software running on someone else's hardware in someone else's data center.
- The user sees only one resource (HW, Os) but uses virtually multiple os. HW resources etc..
- Cloud architecture effectively uses virtualization

- A model of computation and data storage based on “pay as you go” access to “unlimited” remote data center capabilities
- A cloud infrastructure provides a framework to manage scalable, reliable, on-demand access to applications
- Cloud services provide the “invisible” backend to many of our mobile applications
- High level of elasticity in consumption
- Historical roots in today’s Internet apps
    - Search, email, social networks, e-com sites
    - File storage (Live Mesh, Mobile Me)

Definition

- **“The National Institute of Standards and Technology (NIST) defines cloud computing as a "pay-per-use model for enabling available, convenient and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."**

Cloud Computing Architecture

- Architecture consists of 3 tiers
    - **Cloud Deployment Model**
    - **Cloud Service Model**
    - **Essential Characteristics of Cloud Computing**

Essential Characteristics 1

- On-demand self-service.
    - consumer can unilaterally provision computing capabilities such as server time and network storage as needed automatically, without requiring human interaction with a service provider.

Essential Characteristics 2

- Broad network access.
    - Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms

(e.g., mobile phones, laptops, and PDAs) as well as other traditional or cloudbased software services.

Essential Characteristics 3

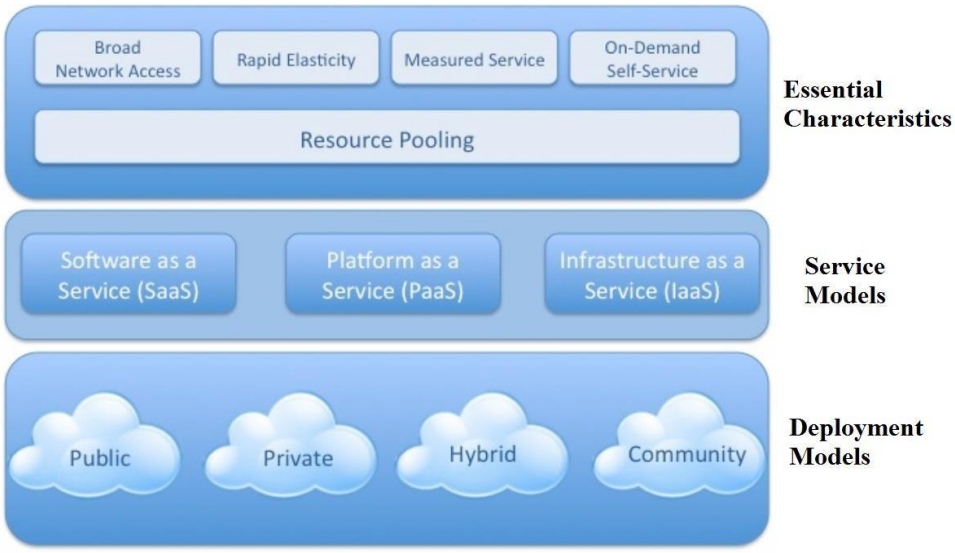
- Resource pooling.
  - The provider's computing resources are pooled to serve multiple consumers using a **multi-tenant model**, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Essential Characteristics 4

- **Rapid elasticity.**
  - Capabilities can be rapidly and elastically provisioned - in some cases automatically - to quickly scale out; and rapidly released to quickly scale in.
  - To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Essential Characteristics 5

- **Measured service.**
  - Cloud systems automatically control and optimize resource usage by leveraging a metering capability at some level of abstraction appropriate to the type of service.
  - Resource usage can be monitored, controlled, and reported - providing transparency for both the provider and consumer of the service.



Cloud Service Models

- Cloud **S**oftware as a Service (**S**aaS)
- Cloud **P**latform as a Service (**P**aaS)
- Cloud **I**nfrastructure as a Service (**I**aaS)

SaaS

- SaaS is a licensed software offering on the cloud and pay per use
- SaaS is a software delivery methodology that provides licensed multi-tenant access to software and its functions remotely as a Web-based service. Usually billed based on usage
  - Usually multi tenant environment
  - Highly scalable architecture
- Customers do not invest on software application programs
- The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.
- The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email).
- The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, data or even individual application capabilities, with the possible exception of limited user specific application configuration settings.

SaaS providers

- Google's Gmail, Docs, Talk etc
- Microsoft's Hotmail, Sharepoint
- SalesForce,
- Yahoo, Facebook

Infrastructure as a Service (IaaS)

- IaaS is the delivery of technology infrastructure (mostly hardware) as an on demand, scalable service
  - Usually billed based on usage
  - Usually multi tenant virtualized environment
  - Can be coupled with Managed Services for **OS** and application support
  - User can choose his OS, storage, deployed app, networking components

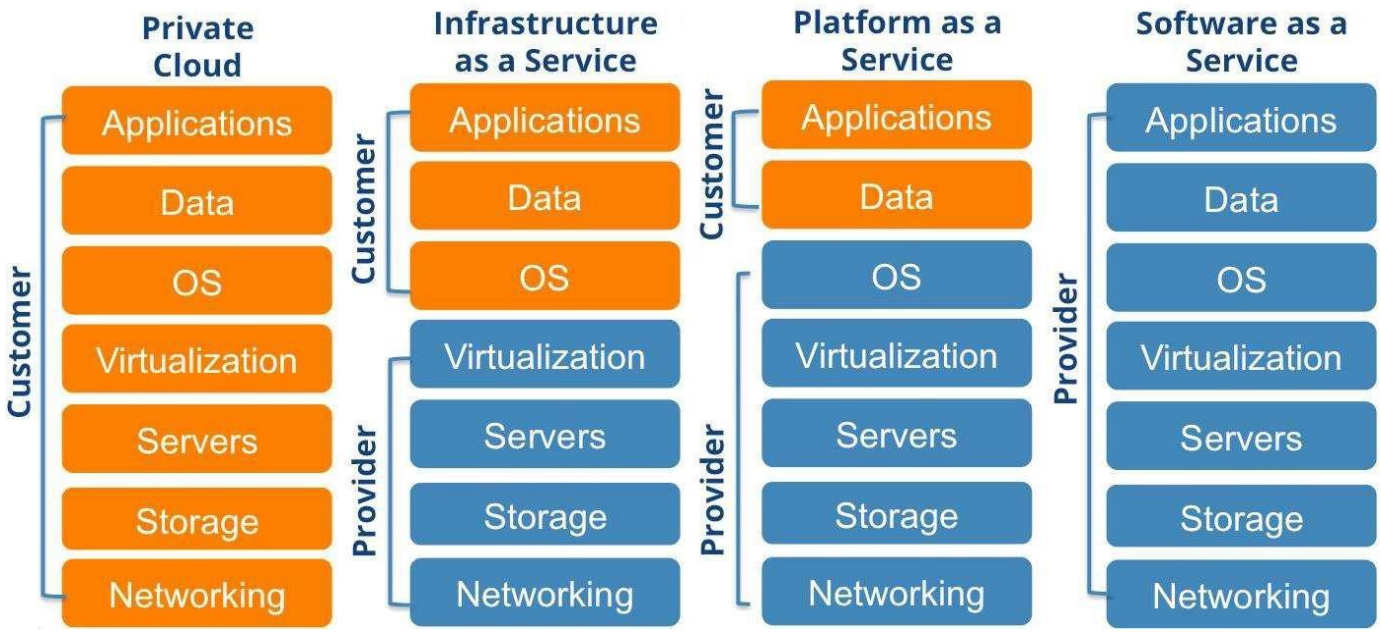


Figure 1.6 Cloud Service Model

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

IaaS providers

- Amazon Elastic Compute Cloud (EC2)
  - Each instance provides 1-20 processors, upto 16 GB RAM, 1.69TB storage
- RackSpace Hosting
  - Each instance provides 4 core CPU, upto 8 GB RAM, 480 GB storage
- Joyent Cloud
  - Each instance provides 8 CPUs, upto 32 GB RAM, 48 GB storage
- Go Grid
  - Each instance provides 1-6 processors, upto 15 GB RAM, 1.69TB storage

Platform as a Service (PaaS)

- PaaS provides all of the facilities required to support the complete life cycle of building, delivering and deploying web applications and services entirely from the Internet.

Typically applications must be developed with a particular platform in mind

- Multi tenant environments
- Highly scalable multi tier architecture

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

PaaS providers

- Google App Engine
  - Python, Java, Eclipse
- Microsoft Azure
  - .Net, Visual Studio
- Sales Force
  - Apex, Web wizard
- TIBCO,
- VMware,
- Zoho

- In cloud computing**, multitenancy** can also refer to shared hosting, in which server resources are divided among different customers. Multitenancy is the opposite of single tenancy, when a software instance or computer system has 1 end user or group of users. Multitenancy allows multiple instances of the given application to operate in a shared environment.

What are the types of tenancy in cloud computing?

Degrees of multi-tenancy

- Highest degree: IaaS and PaaS are multi-tenant. SaaS is fully multi-tenant also.
- Middle degree: IaaS and PaaS are multi-tenant. Small SaaS clusters are multi-tenant.
- Lowest degree: IaaS and PaaS are multi-tenant. SaaS is single tenant.

Types of Cloud/CLOUD DEPLYMENT MODEL

There are the following 4 types of cloud that you can deploy according to the organization'sneeds-

- Public Cloud
- Private Cloud
- Hybrid Cloud

- Community Cloud

Public Cloud

Public cloud is **open to all** to store and access information via the Internet using the pay-per-usage method.
In public cloud, computing resources are managed and operated by the Cloud Service Provider (CSP).
Example: Amazon elastic compute cloud (EC2), IBM SmartCloud Enterprise, Microsoft,Google App Engine, Windows Azure Services Platform.

Advantages of Public Cloud

There are the following advantages of Public Cloud -

- Public cloud is owned at a lower cost than the private and hybrid cloud.
- Public cloud is maintained by the cloud service provider, so do not need to worryabout the maintenance.
- Public cloud is easier to integrate. Hence it offers a better flexibility approach toconsumers.
- Public cloud is location independent because its services are delivered through theinternet.
- Public cloud is highly scalable as per the requirement of computing resources.
- It is accessible by the general public, so there is no limit to the number of users.

Disadvantages of Public Cloud

- Public Cloud is less secure because resources are shared publicly.
- Performance depends upon the high-speed internet network link to the cloud provider.
- The Client has no control of data.

Private Cloud

Private cloud is also known as an **internal cloud** or **corporate cloud**. It is used by organizations to build and manage their own data centers internally or by the third party. It can be deployed using Opensource tools such as Openstack and Eucalyptus.
Based on the location and management, National Institute of Standards and Technology (NIST) divide private cloud into the following two parts-

- On-premise private cloud
- Outsourced private cloud

Advantages of Private Cloud

There are the following advantages of the Private Cloud -

- Private cloud provides a high level of security and privacy to the users.
- Private cloud offers better performance with improved speed and space capacity.
- It allows the IT team to quickly allocate and deliver on-demand IT resources.
- The organization has full control over the cloud because it is managed by the organization itself. So, there is no need for the organization to depends on anybody.
- It is suitable for organizations that require a separate cloud for their personal use anddata security is the first priority.

Disadvantages of Private Cloud

- Skilled people are required to manage and operate cloud services.
- Private cloud is accessible within the organization, so the area of operations is limited.
  Private cloud is not suitable for organizations that have a high user base, and organizations that

do not have the prebuilt infrastructure, sufficient manpower to maintain and manage the cloud

Hybrid Cloud

Hybrid Cloud is a combination of the public cloud and the private cloud. we can say:

Hybrid Cloud = Public Cloud + Private Cloud

Hybrid cloud is partially secure because the services which are running on the public cloud can be accessed by anyone, while the services which are running on a private cloud can be accessed only by the organization's users.

Example: Google Application Suite (Gmail, Google Apps, and Google Drive), Office 365 (MS Office on the Web and One Drive), Amazon Web Services.

Advantages of Hybrid Cloud

There are the following advantages of Hybrid Cloud -

- Hybrid cloud is suitable for organizations that require more security than the publiccloud.
- Hybrid cloud helps you to deliver new products and services more quickly.
- Hybrid cloud provides an excellent way to reduce the risk.
- Hybrid cloud offers flexible resources because of the public cloud and secure resources because of the private cloud.

Disadvantages of Hybrid Cloud

- In Hybrid Cloud, security feature is not as good as the private cloud.
- Managing a hybrid cloud is complex because it is difficult to manage more than onetype of deployment model.
- In the hybrid cloud, the reliability of the services depends on cloud service providers.

Community Cloud

Community cloud allows systems and services to be accessible by a group of several organizations to share the information between the organization and a specific community. It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.

Example: Health Care community cloud

Advantages of Community Cloud

There are the following advantages of Community Cloud -

- Community cloud is cost-effective because the whole cloud is being shared by severalorganizations or communities.
- Community cloud is suitable for organizations that want to have a collaborative cloudwith more security features than the public cloud.
- It provides better security than the public cloud.
- It provdes collaborative and distributive environment.
- Community cloud allows us to share cloud resources, infrastructure, and other capabilities among various organizations.

Disadvantages of Community Cloud

- Community cloud is not a good choice for every organization.
- Security features are not as good as the private cloud.
- It is not suitable if there is no collaboration.

The fixed amount of data storage and bandwidth is shared among all communitymembers

Parameter	Public Cloud	Private Cloud	Hybrid Cloud	CommunityCloud
Host	Service provider	Enterprise (Third party)	Enterprise (Third party)	Community (Third party)
Users	General public	Selected users	Selected users	Communitymembers
Access	Internet	Internet, VPN	Internet, VPN	Internet, VPN
Owner	Service provider	Enterprise	Enterprise	Community

CloudComputing - Opportunities and Challenges

- It enables services to be used without any understanding of their infrastructure.
- Cloud computing works using economies of scale
- It potentially lowers the outlay expense for start up companies, as they would no longer need to buy their own software or servers.
- Cost would be by on-demand pricing.
- Vendors and Service providers claim costs by establishing an ongoing revenue stream.
- Data and services are stored remotely but accessible from "anywhere"

Cloud Computing – Pros

- Lower computer costs
- Instant software updates:
    - When the application is web-based, updates happen automatically
- Improved document format compatibility
- e capacity:
    - Cloud computing offers virtually limitless storage
    - • Increased data reliability:

Cloud Computing – Cons

- Need of Internet :

- A dead Internet connection means no work and in areas where Internet connections are few or inherently unreliable, this could be a deal-breaker.
- Requires a constant Internet connection

- Can be slow:
  - Even with a fast connection, web-based applications can sometimes be slower than accessing a similar software program on your desktop PC.
- Disparate Protocols :
  - Each cloud systems uses different protocols and different APIs – Standards yet to evolve.

Evolution of Cloud Computing

Evolution of Cloud Computing

- Cloud Computing Leverages dynamic resources to deliver a large number of services to end users.
- It is High Throughput Computing(HTC) paradigm
- It enables users to share access to resources from anywhere at any time

II Hardware Evolution

- In 1930, binary arithmetic was developed
  - computer processing technology, terminology, and programming languages.

• In 1939,Electronic computer was developed

- Computations were performed using vacuum-tube technology.

• In 1941, Konrad Zuse's Z3 was developed

- Support both floating-point and binary arithmetic.

There are four generations

- First Generation Computers
- Second Generation Computers
- Third Generation Computers
- Fourth Generation Computers

a.First Generation Computers

Time Period : 1942 to 1955

Technology : Vacuum Tubes

Size : Very Large System

Processing : Very Slow

- The first generation of modern computers can be traced to 1943, when the Mark I and Colossus computers were developed, albeit for quite different purposes.
- With financial backing from IBM (then International Business Machines Corporation), the Mark I was designed and developed at Harvard University. It was a general-purpose electromechanical programmable computer.
- Colossus, on the other hand, was an electronic computer built in Britain at the end 1943. Colossus was the world's first programmable, digital, electronic, computing device.
- First-generation computers were built using hard-wired circuits and vacuum tubes (thermionic valves). Data was stored using paper punch cards.
- Colossus was used in secret during World War II to help decipherteleprinter messages encrypted by German forces using the Lorenz SZ40/42 machine. British code breakers referred to encrypted German teleprinter traffic as "Fish" and called the SZ40/42 machine and its traffic "Tunny."
- To accomplish its deciphering task, Colossus compared two data streams readat high speed from a paper tape. Colossus evaluated one data stream representing the encrypted "Tunny," counting each match that was discoveredbased on a programmable Boolean function. A comparison with the other data stream was then made.
- The second data stream was generated internally and designed to be an electronic simulation of the Lorenz SZ40/42 as it ranged through various trialsettings.
- If the match count for a setting was above a predetermined threshold, that data match would be sent as character output to an electric typewriter.

Examples:

1.ENIAC (Electronic Numerical Integrator and Computer)

2.EDVAC(Electronic Discrete Variable Automatic Computer)

Advantages:

• It made use of vacuum tubes which was the advanced technology at that time

• Computations were performed in milliseconds.

Disadvantages:

• very big in size, weight was about 30 tones.

• very costly.

• Requires more power consumption

•Large amount heat was generated.

b.Second Generation Computers

Time Period : 1956 to 1965.

Technology : Transistors

Size : Smaller

Processing : Faster

o Examples

Honeywell 400

IBM 7094

- Another general-purpose computer of this era was ENIAC (Electronic Numerical Integrator and Computer, shown in Figure, which was built in 1946. This was the first Turing-complete, digital computer capable of being reprogrammed to solve a full range of computing problems, although earlier machines had been built with some of these properties.
- ENIAC's original purpose was to calculate artillery firing tables for the U.S. Army's Ballistic Research Laboratory. ENIAC contained 18,000 thermionic valves, weighed over 60,000 pounds, and consumed 25 kilowatts of electrical power per hour. ENIAC was capable of performing 100,000 calculations a second.
- Within a year after its completion, however**,** the invention of the transistor meant that the inefficient thermionic valves could be replaced with smaller, more reliable components, thus marking another major step in the history of computing.
- Transistorized computers marked the advent of second-generation computers,which dominated in the late 1950sand early 1960s. Despite using transistors and printed circuits, these computers were still bulky and expensive. They were therefore used mainly by universities and government agencies.
- The integrated circuit or microchip was developed by Jack St. Claire Kilby, an achievement for which he received the Nobel Prize in Physics in 2000.

Advantages

- Less heat than first generation.
- Assembly language and punch cards were used for input.
- Low cost than first generation computers.
- Computations was performed in microseconds.
- Better Portability as compared to first generation

Disadvantages:

- A cooling system was required.
- Constant maintenance was required.
- Only used for specific purposes

c.Third Generation Computers

Time Period : 1966 to 1975

Technology : ICs (Integrated Circuits)

Size : Small as compared to 2nd generation computers

Processing : Faster than 2nd generation computers

Examples

• PDP-8 (Programmed Data Processor)

• PDP-11

- Kilby's invention started an explosion in third-generation computers. Even though the first integrated circuit was produced in September 1958, microchips were not used in computers until 1963.
- While mainframe computers like the IBM 360 increased storage and processing capabilities even further, the integrated circuit allowed the development of minicomputers that began to bring computing into many smaller businesses.
- Large-scale integration of circuits led to the development of very smallprocessing units, the next step along the evolutionary trail of computing. In November 1971, Intel released the world's first commercial microprocessor, the Intel 4004. The 4004 was the first complete CPU on one chip and became the first commercially available microprocessor.
- It was possible because of the development of new silicon gate technology that enabled engineers to integrate a much greater number of transistors on a chip that would perform at a much faster speed. This development enabled the rise of the fourth-generation computer platforms.

Advantages

• These computers were cheaper as compared to generation computers.

• They were fast and reliable.

• IC not only reduce the size of the computer but it also improves the performance of the computer

• Computations was performed in nanoseconds

Disadvantages

• IC chips are difficult to maintain.

• The highly sophisticated technology required for the manufacturing of IC chips.

•Air Conditioning is required

d.Fourth Generation Computers

Time Period : 1975 to Till Date

Technology : Microprocessor

Size : Small as compared to third generation computer

Processing : Faster than third generation computer

- The fourth-generation computers that were being developed at this time utilized a microprocessor that put the computer's processing capabilities on a single integrated circuit chip. By combining random access memory(RAM), developed by Intel, fourth-generation computers were faster than ever before and had much smaller footprints.
- The 4004 processor was capable of "only" 60,000 instructions per second. As technology progressed, however, new processors brought even more speedand computing capability to users. The microprocessors that evolved from the 4004 allowed manufacturers to begin developing personal computers small enough and cheap enough to be purchased by the general public.

- The first commercially available personal computer was the MITS Altair 8800, released at the end of 1974. What followed was a flurry of other personal computers to market, such as the Apple I and II, the Commodore PET, the VIC-20, the Commodore 64, and eventually the original IBM PC in 1981. The PC era had begun in earnest by the mid-1980s.
- Even though microprocessing power, memory and data storage capacities have increased by many orders of magnitude since the invention of the 4004 processor, the technology for large-scale integration (LSI) or very-large- scale integration (VLSI) microchips has not changed all that much. For this reason, most of today's computers still fall into the category of fourth-generation computers.

Internet Software Evolution

- The Internet is named after the Internet Protocol, the standard communications protocol used by every computer on the Internet. The conceptual foundation for creation of the Internet was significantly developedby three individuals.
- The first, Vannevar Bush, wrote a visionary description of the potential uses for information technology with his description of an automated library system named MEMEX**.**
- Bush introduced the concept of the MEMEX in the 1930s as a microfilm- based "device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility."

Examples

• IBM 4341

• DEC 10

Advantages:

- Fastest in computation and size get reduced as compared to the previous generation of computer. Heat generated is small.
- Less maintenance is required.

Disadvantages:

- The Microprocessor design and fabrication are very complex.
- Air Conditioning is required in many cases

III Internet Hardware Evolution

- Internet Protocol is the standard communications protocol used by every computer on the Internet.
- The conceptual foundation for creation of the Internet was significantly developed by three individuals.
  - Vannevar Bush — MEMIX (1930)
  - Norbert Wiener
  - Marshall McLuhan

- Licklider was founder for the creation of the AR PANET (Advanced Research Projects Agency Network)
- Clark deployed a minicomputer called an Interface Message Processor (IMP) at each site.
- Network Control Program (NCP)- first networking protocol that was used on the ARPANET

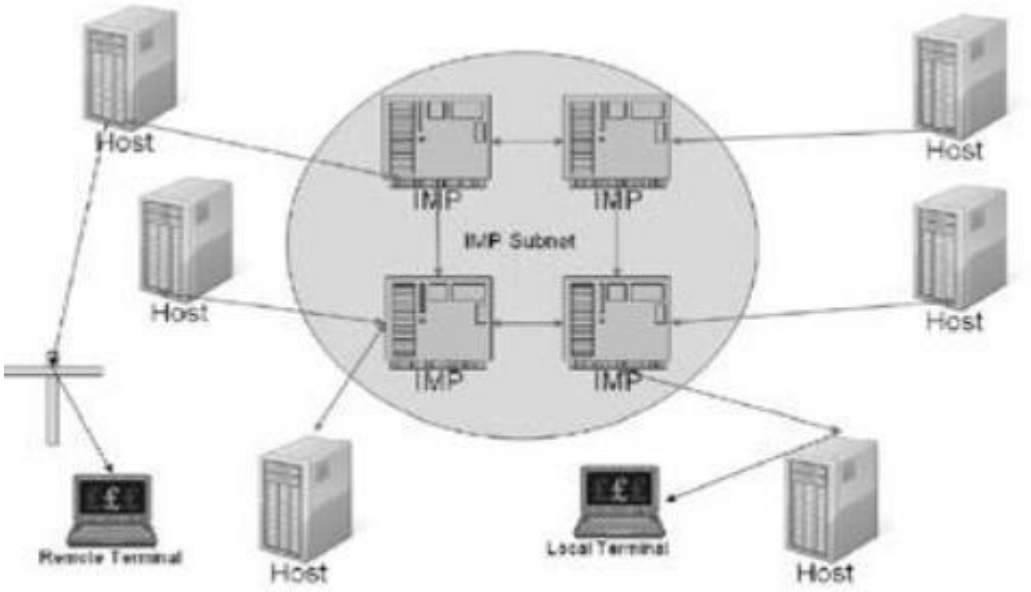


Figure 1.7 **IMP Architecture**

Internet Hardware Evolution

- Establishing a Common Protocol for the Internet
- Evolution of Ipv6
- Finding a Common Method to Communicate Using the Internet Protocol
- Building a Common Interface to the Internet
- The Appearance of Cloud Formations From One Computer to a Grid of Many

a.Establishing a Common Protocol for the Internet

- NCP essentially provided a transport layer consisting of the ARPANET Host-to-Host Protocol (AIIIIP) and the Initial Connection Protocol (ICP)
- Application protocols
    - File Transfer Protocol (FTP), used for file transfers,
    - Simple Mail Transfer Protocol (SMTP), used for sending email

Four versions of TCP/IP

• TCP vl

- TCP v2
- TCP v3 and IP v3,
- TCP v4 and IP v4

b.Evolution of Ipv6

- IPv4 was never designed to scale to global levels.
- To increase available address space, it had to process large data packets (i.e., more bits of data).
- To overcome these problems, Internet Engineering Task Force (IETF) developed IPv6, which was released in January 1995.
- Ipv6 is sometimes called the Next Generation Internet Protocol (IPNG) or TCP/IP v6.

c.Finding a Common Method to Communicate Using the Internet Protocol

- In the 1960s,the word ktpertext was created by Ted Nelson.
- In 1962, Engelbart's first project was Augment, and its purpose was to develop computer tools to augment human capabilities.
- He developed the mouse, Graphical user interface (GUI), and the first working hypertext system, named NLS (oN-Line System).
- NLS was designed to cross-reference research papers for sharing among geographically distributed researchers.
- In the 1980s, Web was developed in Europe by Tim Berners-Lee and Robert Cailliau

d.Building a Common Interface to the Internet

- Betters-Lee developed the first web browser featuring an integrated editor that could create hypertext documents.
- Following this initial success, Berners-Lee enhanced the server and browser by adding support for the FTP (File Transfer protocol)

Figure 1.8 First Web Browser

- Mosaic was the first widely popular web browser available to the general public. Mosaic

support for graphics, sound, and video clips.

- In October 1994, Netscape released the first beta version of its browser, Mozilla 0.96b, over the Internet.
- In 1995, Microsoft Internet Explorer was developed that supports both a graphical Web browser and the name for a set of technologies.
- Mozilla Firefox. released in November 2004, became very popular almost immediately.

e.The Appearance of Cloud Formations From One Computer to a Grid of Many

- Two decades ago, computers were clustered together to form a single larger computer in order to simulate a supercomputer and greater processing power.
- In the early 1990s, Ian Foster and Carl Kesselman presented their concept of "The Grid." They used an analogy to the electricity grid, where users could plug in and use a (metered) utility service.
- A major problem in clustering model was data residency. Because of the distributed nature of a grid, computational nodes could be anywhere in the world.
- The Globus Toolkit is an open source software toolkit used for building grid systems and applications

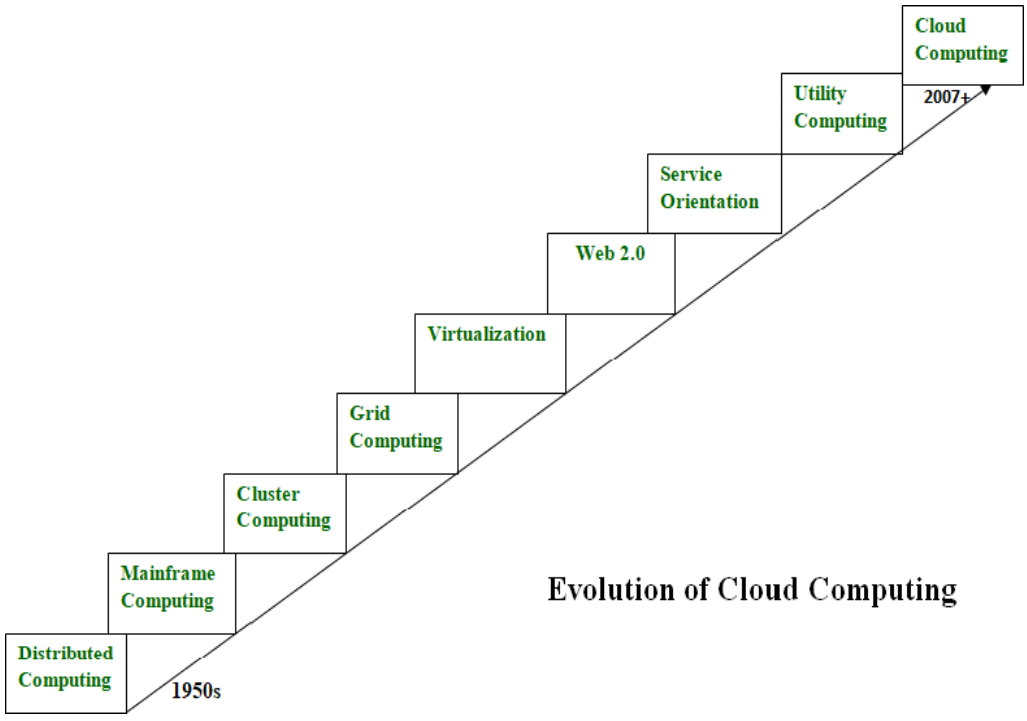


Figure 1.9 Evolution

Evolution of Cloud Services

2008-2009	**Google Application Engine** **Microsoft Azure**
2006	**S3 launches EC2**
2002	**Launch of Amazon Web Services**
1990	**The first milestone of cloud computing arrival of salesforce.com**
1960	**Super Computers** **Mainframes**

IV. SERVER VIRTUALIZATION

- Virtualization is a method of running multiple independent virtual operating systems on a single physical computer.
- This approach maximizes the return on investment for the computer.
- Virtualization technology is a way of reducing the majority of hardware acquisition and maintenance costs, which can result in significant savings for any company.
    - Parallel Processing
    - Vector Processing
    - Symmetric Multiprocessing Systems
    - Massively Parallel Processing Systems

a.Parallel Processing

- Parallel processing is performed by the simultaneous execution of program instructions that have been allocated across multiple processors.
- Objective: running a progran in less time.
- The next advancement in parallel processing-multiprogramming
- In a multiprogramming system, multiple programs submitted by users but each allowed to use the processor for a short time.
- This approach is known as "round-robin scheduling"(RR scheduling)

b.Vector Processing

- Vector processing was developed to increase processing performance by operating in a multitasking manner.
- Matrix operations were added to computers to perform arithmetic operations.
- This was valuable in certain types of applications in which data occurred in the form of vectors or matrices.

- In applications with less well-formed data, vector processing was less valuable.

c.Symmetric Multiprocessing Systems

- Symmetric multiprocessing systems (SMP) was developed to address the problem of resource management in master/slave models.
- In SMP systems, each processor is equally capable and responsible for managing the workflow as it passes through the system.
- The primary goal is to achieve sequential consistency

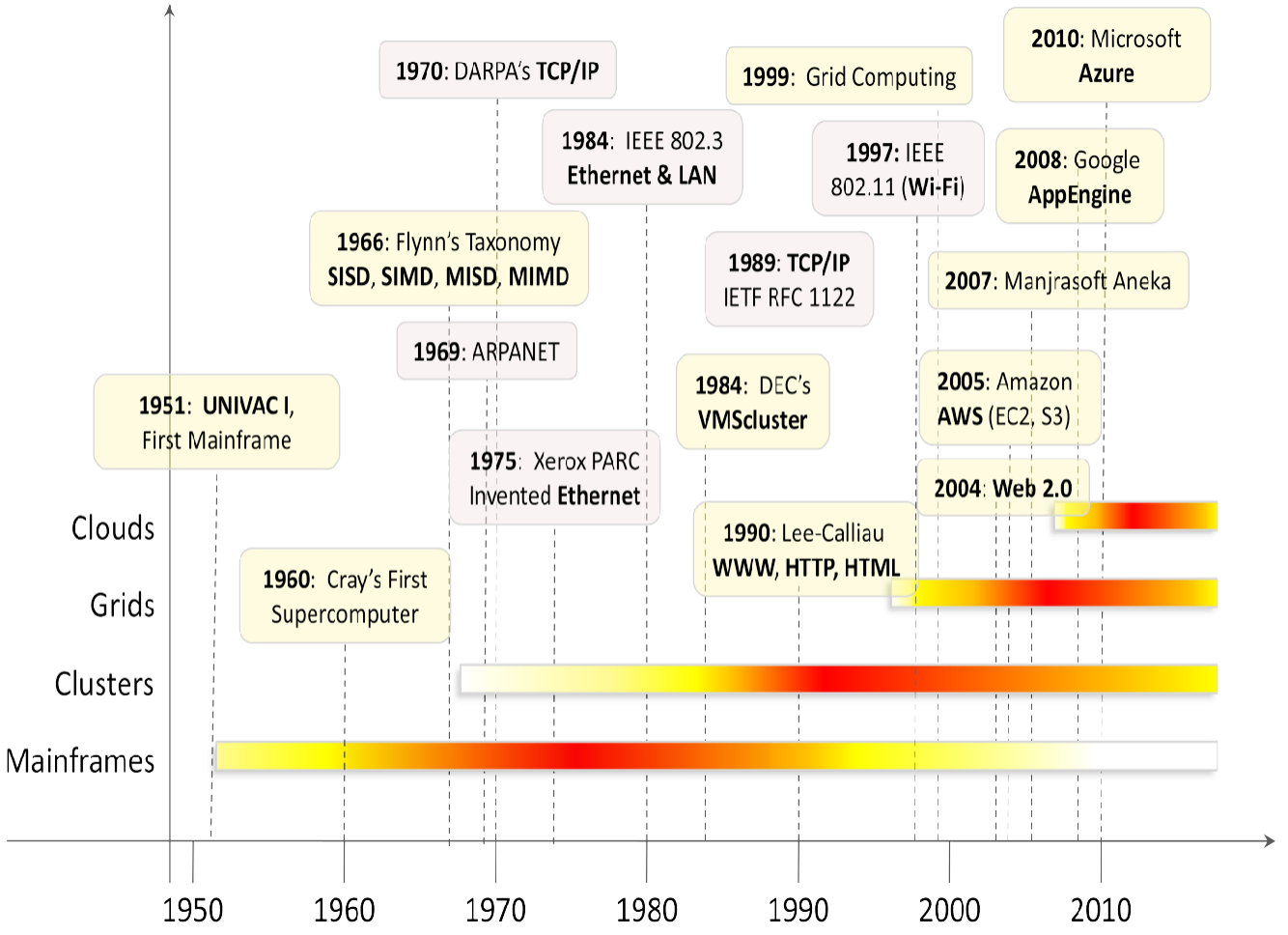
d.Massively Parallel Processing Systems

- In Massively Parallel Processing Systems, a computer system with many independent arithmetic units, which run in parallel.
- All the processing elements are interconnected to act as one very large computer.
- Early examples of MPP systems were the Distributed ArrayProcessor, the Goodyear MPP, the Connection Machine, and the Ultracomputer
- MPP machines are not easy to program, but for certain applications, such as data mining, they are the best solution

Principles of Parallel and Distributed Computing

- **Three major milestones have led to cloud computing evolution**
  - Mainframes: Large computational facilities leveraging multiple processing units. Even though mainframes cannot be considered as distributed systems, they offered large computational power by using multiple processors, which were presented as a single entity to users.

Mile Stones to Cloud computing Evolution

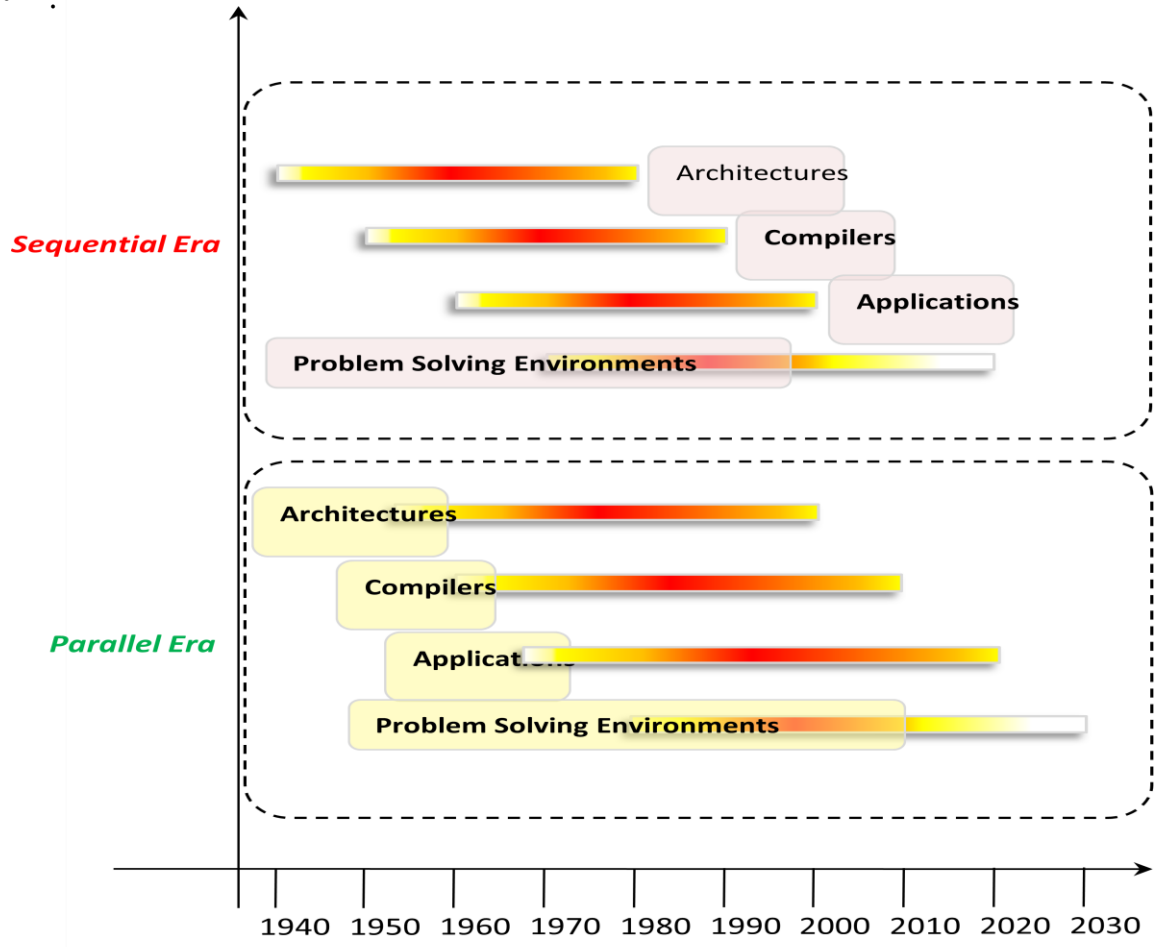


- Clusters: An alternative technological advancement to the use of mainframes and super computers.
- Grids

– Clouds

Eras of Computing

- Two fundamental and dominant models of computing are ***sequential*** and ***parallel***.
  - The sequential era began in the 1940s, and Parallel(and distributed) computing era followed it within a decade.
- Four key elements of computing developed during three eras are
  - Architecture
  - Compilers
  - Applications
  - Problem solving environments
- .



- The computing era started with development in ***hardware architectures***, which actually enabled the creation of ***system software*** – particularly in the area of **compilers and operating systems** – which support the management of such systems and the development of ***applications***

- The term parallel computing and distributed computing are **often used interchangeably**, even though they mean **slightly different things**.
- The term **parallel implies a tightly coupled system**, where as **distributed systems refers to a wider class of system, including those that are tightly coupled**.
- More precisely, the term **parallel computing** refers to a model in which **the computation is divided among several processors sharing the same memory**.
- The architecture of **parallel computing system** is often characterized by the **homogeneity of components**: **each processor is of the same type and it has the same capability as the others.**

- The shared memory has a single address space, which is accessible to all the processors.
- Parallel programs are then broken down into several units of execution that can be allocated to different processors and can communicate with each other by means of shared memory.
- Originally parallel systems are considered as those architectures that featured multiple processors sharing the same physical memory and that were considered a single computer.
  - Over time, these restrictions have been relaxed, and parallel systems now include all architectures that are based on the concept of shared memory, whether this is physically present or created with the support of libraries, specific hardware, and a highly efficient networking infrastructure.
  - For example: a cluster of which of the nodes are connected through an InfiniBand network and configured with distributed shared memory system can be considered as a parallel system.
- The term distributed computing encompasses any architecture or system that allows the computation to be broken down into units and executed concurrently on different

computing elements, whether these are processors on different nodes, processors on the same computer, or cores within the same processor.

- Distributed computing includes a wider range of systems and applications than parallel computing and is often considered a more general term.
- Even though it is not a rule, the term distributed often implies that the locations of the computing elements are not the same and such elements might be heterogeneous in terms of hardware and software features.
- Classic examples of distributed computing systems are
  - Computing Grids
  - Internet Computing Systems

Elements of Parallel computing

- Silicon-based processor chips are reaching their physical limits. Processing speed is constrained by the speed of light, and the density of transistors packaged in a processor is constrained by thermodynamics limitations.
- A viable solution to overcome this limitation is to connect multiple processors working in coordination with each other to solve "Grand Challenge" problems.
- The first step in this direction led
  - To the development of parallel computing, which encompasses techniques, architectures, and systems for performing multiple activities in parallel.

a.Parallel Processing

- Processing of multiple tasks simultaneously on multiple processors is called ***parallel processing***.
- The parallel program consists of multiple active processes (tasks) simultaneously solving a given problem.
- A given task is divided into multiple subtasks using a divide-and-conquer technique, and each subtask is processed on a different central processing unit (CPU).
- Programming on multi processor system using the divide-and-conquer technique is called ***parallel programming***.
- Many applications today require more computing power than a traditional sequential computer can offer.

- Parallel Processing provides a cost effective solution to this problem by increasing the number of CPUs in a computer and by adding an efficient communication system between them.
- The workload can then be shared between different processors. This setup results in higher computing power and performance than a single processor a system offers.

Parallel Processing influencing factors

- The development of parallel processing is being influenced by many factors. The prominent among them include the following:
  - Computational requirements are ever increasing in the areas of both scientific and business computing. The technical computing problems, which require high-speed computational power, are related to
    - life sciences, aerospace, geographical information systems, mechanical design and analysis etc.
  - Sequential architectures are reaching mechanical physical limitations as they are constrained by the speed of light and thermodynamics laws.
    - The speed which sequential CPUs can operated is reaching saturation point (no more vertical growth), and hence an alternative way to get high computation speed is to connect multiple CPUs (opportunity for horizontal growth).
  - Hardware improvements in pipelining , super scalar, and the like are non scalable and require sophisticated compiler technology.
    - Developing such compiler technology is a difficult task.
  - Vector processing works well for certain kinds of problems. It is suitable mostly for scientific problems (involving lots of matrix operations) and graphical processing.
    - It is not useful for other areas, such as databases.
  - The technology of parallel processing is mature and can be exploited commercially
    - here is already significant R&D work on development tools and environments.
  - Significant development in networking technology is paving the way for

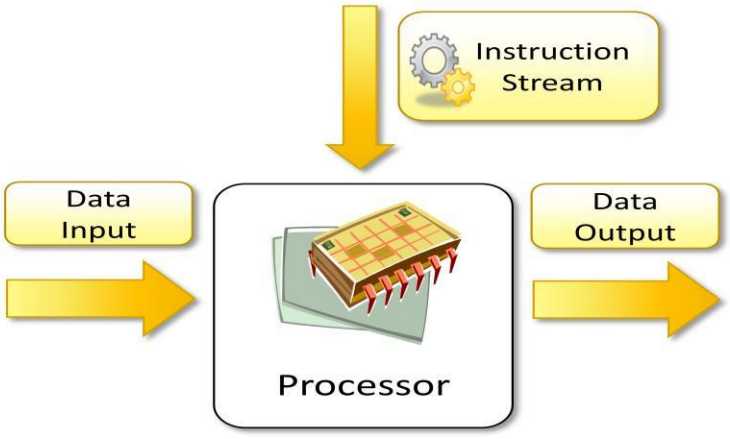
- heterogeneous computing.

b.Hardware architectures for parallel Processing

- The core elements of parallel processing are CPUs. Based on the number of instructions and data streams, that can be processed simultaneously, computing systems are classified into the following four categories:
  - **Single-instruction, Single-data (SISD) systems**
  - **Single-instruction, Multiple-data (SIMD) systems**
  - **Multiple-instruction, Single-data (MISD) systems**
  - **Multiple-instruction, Multiple-data (MIMD) systems**

(i) Single – Instruction , Single Data (SISD) systems

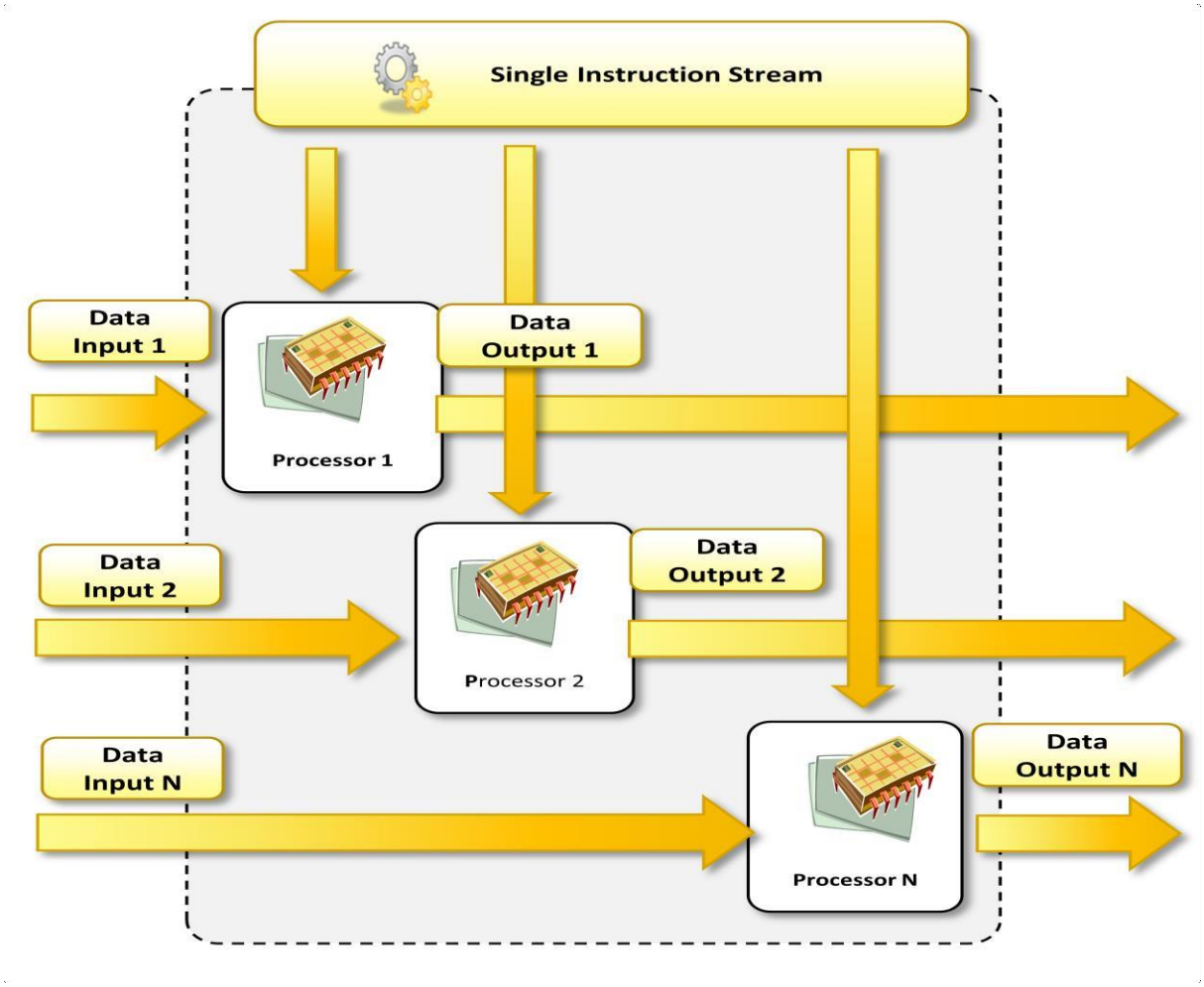
- SISD computing system is a uni-processor machine capable of executing a single instruction, which operates on a single data stream.
- Machine instructions are processed sequentially, hence computers adopting this model are popularly called sequential computers.
- Most conventional computers are built using SISD model.
- All the instructions and data to be processed have to be stored in primary memory.
- The speed of processing element in the SISD model is limited by the rate at which the computer can transfer information internally.
- Dominant representative SISD systems are IBM PC, Macintosh, and workstations.



(ii) Single – Instruction , Multiple Data (SIMD) systems

- SIMD computing system is a multiprocessor machine capable of executing the same instruction on all the CPUs but operating on different data streams.
- Machines based on this model are well suited for scientific computing since they involve lots of vector and matrix operations.
- For instance statement Ci = Ai * Bi, can be passed to all the processing elements (PEs), organized data elements of vectors A and B can be divided into multiple sets (N- sets for N PE systems), and each PE can process one data set.

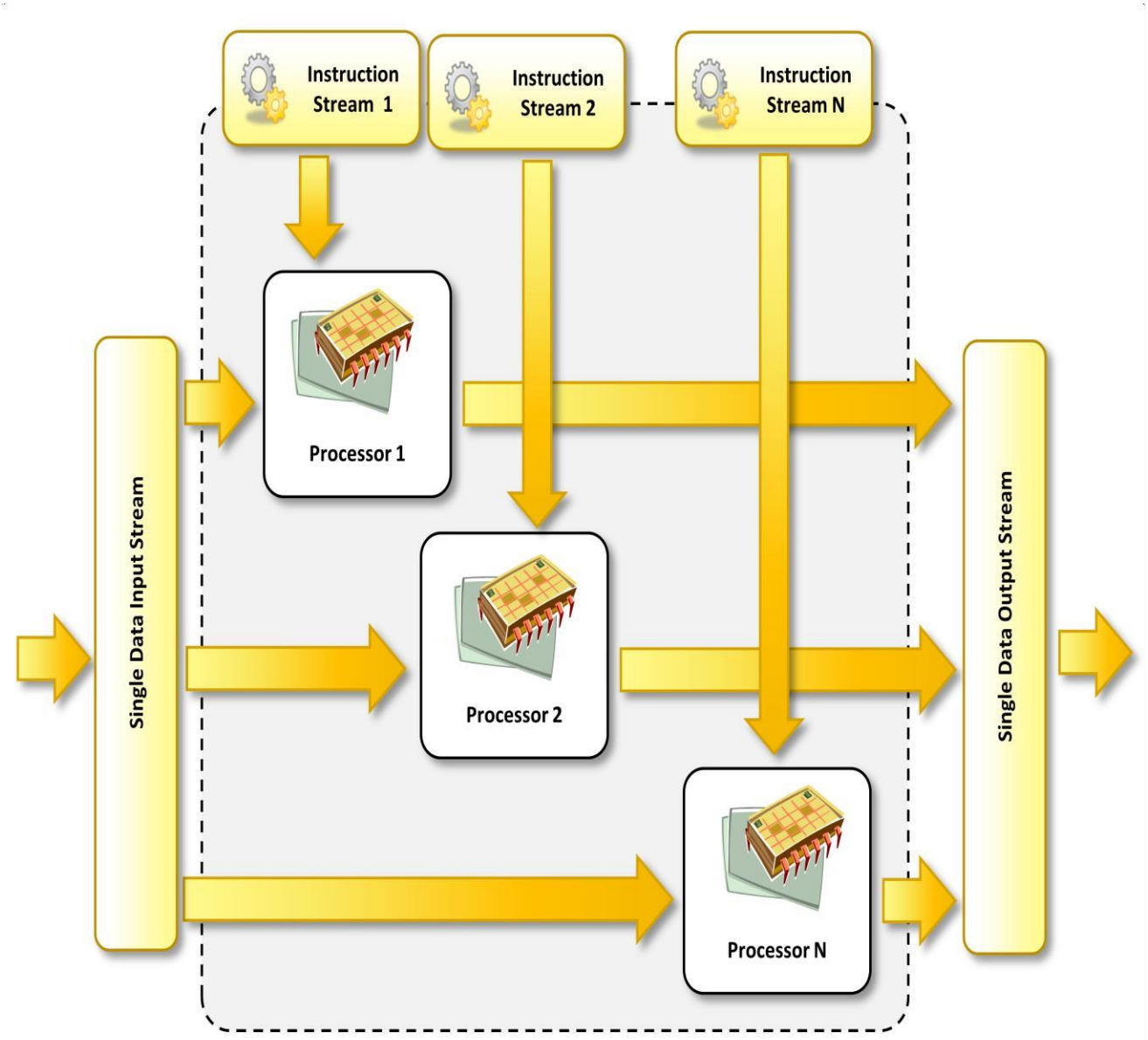
Dominant representative SIMD systems are Cray's Vector processing machine and Thinking Machines Cm*, and GPGPU accelerators



-

(iii) Multiple – Instruction , Single Data (MISD) systems

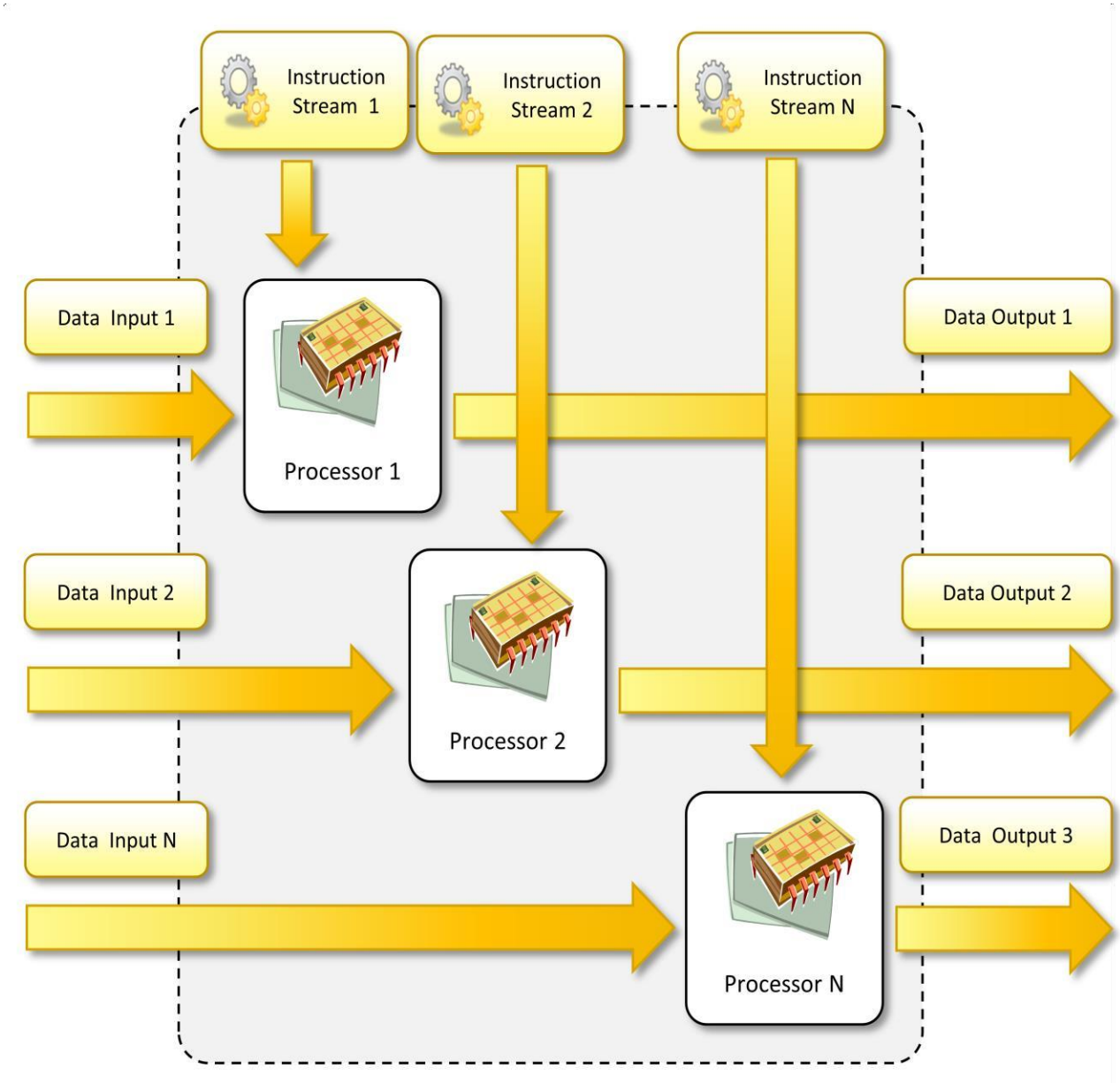
- MISD computing system is a multi processor machine capable of executing different instructions on different Pes all of them operating on the same data set.
- Machines built using MISD model are not useful in most of the applications.
- Few machines are built but none of them available commercially.
- This type of systems are more of an intellectual exercise than a practical configuration.



(iv) Multiple – Instruction , Multiple Data (MIMD) systems

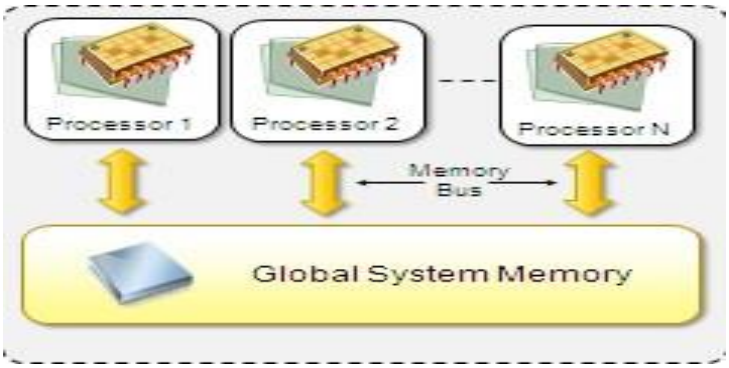
- MIMD computing system is a multi processor machine capable of executing multiple instructions on multiple data sets.
- Each PE in the MIMD model has separate instruction and data streams, hence machines built using this model are well suited to any kind of application.
- Unlike SIMD, MISD machine, PEs in MIMD machines work asynchronously,

MIMD machines are broadly categorized into shared-memory MIMD and distributed memory MIMD based on the way PEs are coupled to the main memory



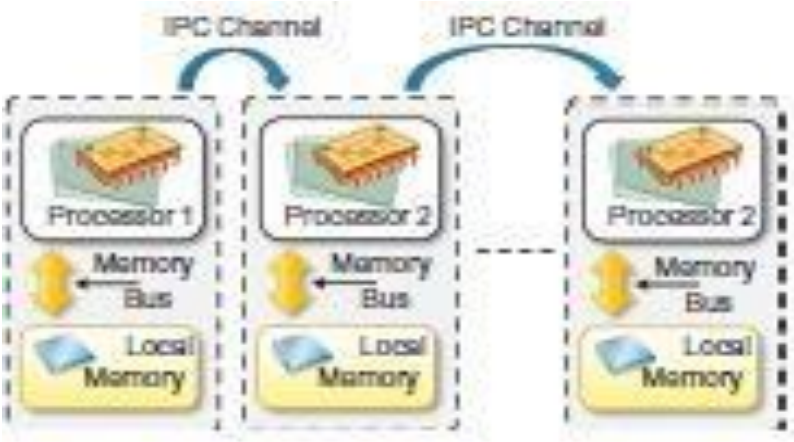
Shared Memory MIMD machines

- All the PEs are connected to a single global memory and they all have access to it.
- Systems based on this model are also called tightly coupled multi processor systems.
- The communication between PEs in this model takes place through the shared memory.
- Modification of the data stored in the global memory by one PE is visible to all other PEs.
- Dominant representative shared memory MIMD systems are silicon graphics machines and Sun/IBM SMP (Symmetric Multi-Processing).



Distributed Memory MIMD machines

- All PEs have a local memory. Systems based on this model are also called loosely coupled multi processor systems.
- The communication between PEs in this model takes place through the interconnection network, the inter process communication channel, or IPC.
- The network connecting PEs can be configured to tree, mesh, cube, and so on.
- Each PE operates asynchronously, and if communication/synchronization among tasks is necessary, they can do so by exchanging messages between them.



Shared Vs Distributed MIMD model

- The shared memory MIMD architecture is easier to program but is less tolerant to failures and harder to extend with respect to the distributed memory MIMD model.
- Failures, in a shared memory MIMD affect the entire system, whereas this is not the case of the distributed model, in which each of the PEs can be easily isolated.
- Moreover, shared memory MIMD architectures are less likely to scale because the addition of more PEs leads to memory contention.
- This is a situation that does not happen in the case of distributed memory, in which each PE has its own memory.

As a result, distributed memory MIMD architectures are most popular today

c.Approaches to Parallel Programming

- A sequential program is one that runs on a single processor and has a single line of control.
- To make many processors collectively work on a single program, the program must be divided into smaller independent chunks so that each processor can work on separate chunks of the problem.
- The program decomposed in this way is a parallel program.
- A wide variety of parallel programming approaches are available.
- The most prominent among them are the following.
- Data Parallelism
- Process Parallelism
- Farmer-and-worker model
- The above said three models are suitable for task-level parallelism. In the case of data level parallelism, the divide-and-conquer technique is used to split data into multiple sets, and each data set is processed on different PEs using the same instruction.
- This approach is highly suitable to processing on machines based on the SIMD model.
- In the case of Process Parallelism, a given operation has multiple (but distinct) activities that can be processed on multiple processors.
- In the case of Farmer-and-Worker model, a job distribution approach is used, one processor is configured as master and all other remaining PEs are designated as slaves,

the master assigns the jobs to slave PEs and, on completion, they inform the master, which in turn collects results.

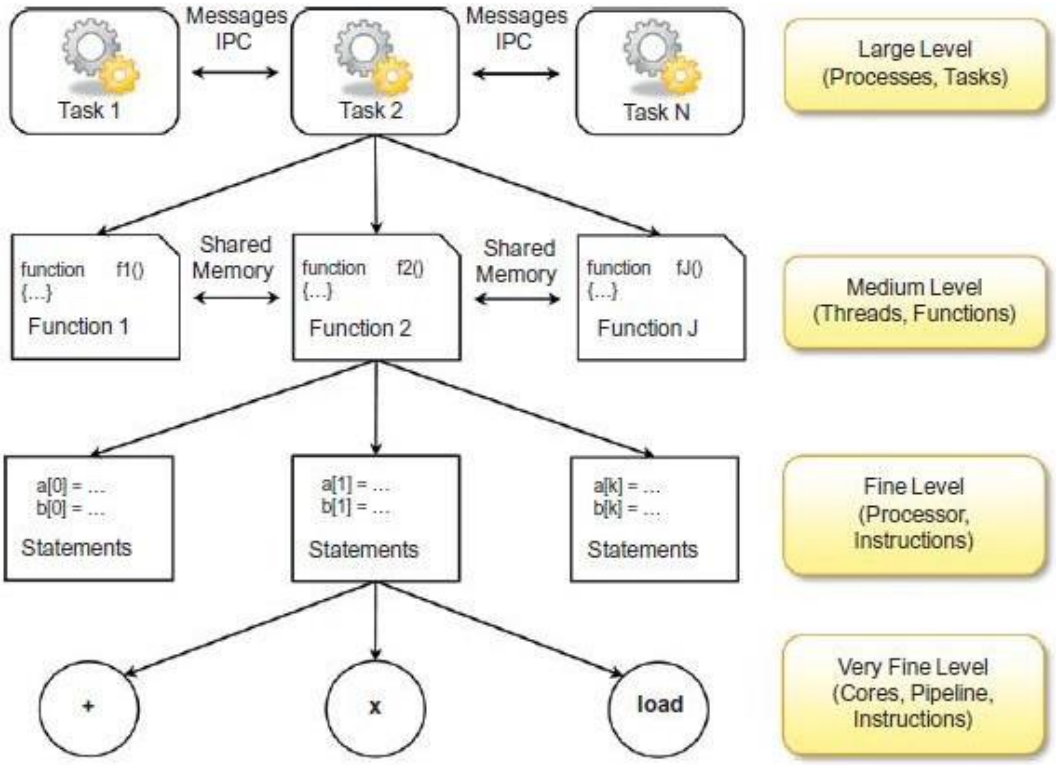
- These approaches can be utilized in different levels of parallelism.

d. Levels of Parallelism

- Levels of Parallelism are decided on the lumps of code (grain size) that can be a potential candidate of parallelism.
- The table shows the levels of parallelism.
- All these approaches have a common goal
  - To boost processor efficiency by hiding latency.
  - To conceal latency, there must be another thread ready to run whenever a lengthy operation occurs.
- The idea is to execute concurrently two or more single-threaded applications. Such as compiling, text formatting, database searching, and device simulation.

Grain Size	Code Item	Parallelized By
Large	Separate and heavy weight process	Programmer
Medium	Function or procedure	Programmer
Fine	Loop or instruction block	Parallelizing compiler
Very Fine	Instruction	Processor

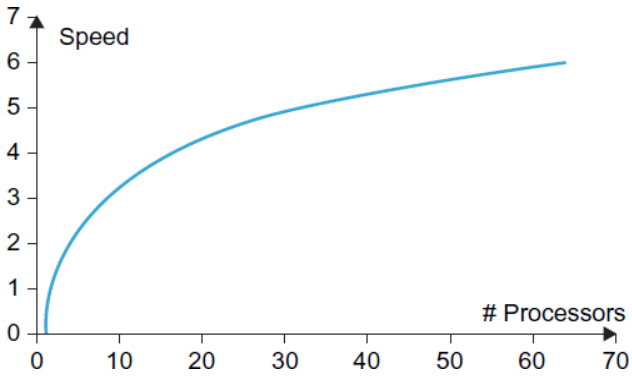
Levels of Parallelism



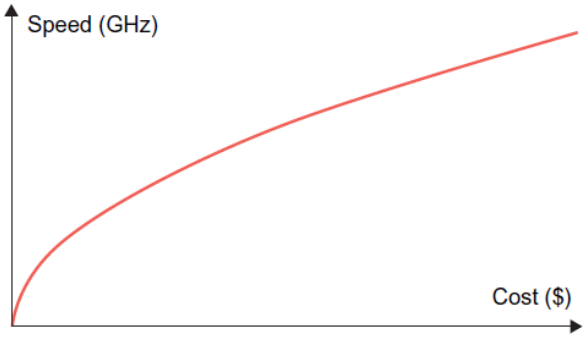
e. Laws of Caution

- Studying how much an application or a software system can gain from parallelism.
- In particular what need to keep in mind is that parallelism is used to perform multiple activities together so that the system can increase its throughput or its speed.
- But the relations that control the increment of speed are not linear.
- For example: for a given n processors, the user expects speed to be increase by in times. This is an ideal situation, but it rarely happens because of the communication overhead.
- Here two important guidelines to take into account.
  - Speed of computation is proportional to the square root of the system cost; they never increase linearly. Therefore, the faster a system becomes, the more expensive it is to increase its speed

- Speed by a parallel computer increases as the logarithm of the number of processors (i.e. y=k*log(N)).



Number processors versus speed



Cost versus speed

Elements of Distributed Computing

a.General concepts and definitions

- Distributed computing studies the models, architectures, and algorithms used for building and managing distributed systems.
- As general definition of the term distributed system, we use the one proposed by Tanenbaum
  - A distributed system is a collection of independent computers that appears to its users as a single coherent system.
- This definition is general enough to include various types of distributed computing systems that are especially focused on unified usage and aggregation of distributed resources.
- Communications is another fundamental aspect of distributed computing. Since distributed systems are composed of more than one computer that collaborate together, it is necessary to provide some sort of data and information exchange between them, which generally occurs through the network.
  - A distributed system is one in which components located at networked computers communicate and coordinate their action only by passing messages.
- As specified in this definition, the components of a distributed system communicate with some sort of message passing. This is a term that encompasses several communication models.

b.Components of distributed System

- A distributed system is the result of the interaction of several components that traverse the entire computing stack from hardware to software.
- It emerges from the collaboration of several elements that- by working together- give users the illusion of a single coherent system.
- The figure provides an overview of the different layers that are involved in providing the services of a distributed system.

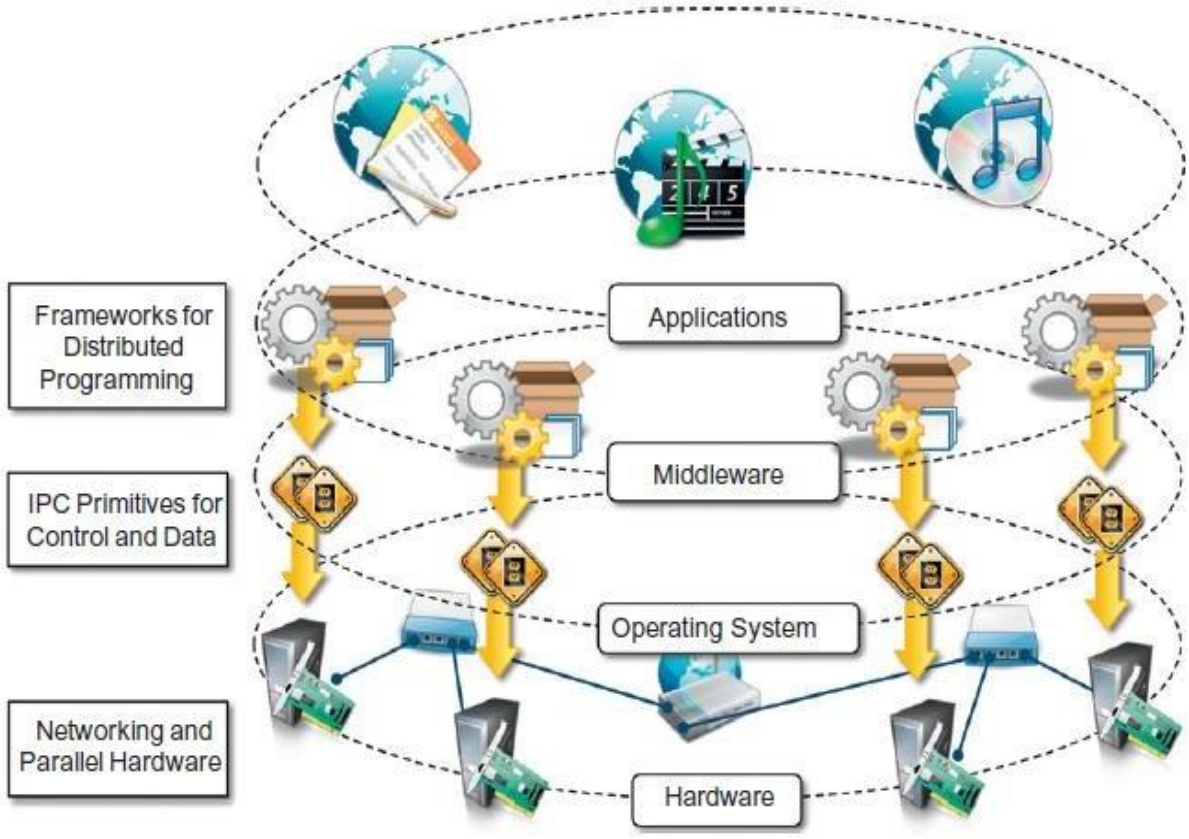


Figure 1.10 A layered view of a distributed system.

c.Architectural styles for distributed computing

- At the very bottom layer, computer and network hardware constitute the physical infrastructure; these components are directly managed by the operating system, which

provides the basic services for inter process communication (IPC), process scheduling and management, and resource management in terms of file system and local devices.

- Taken together these two layers become the platform on top of which specialized software is deployed to turn a set of networked computers into a distributed system
- Although a distributed system comprises the interaction of several layers, the middleware layer is the one that enables distributed computing, because it provides a coherent and uniform runtime environment for applications.
- There are many different ways to organize the components that, taken together, constitute such an environment.
- The interactions among these components and their responsibilities give structure to the middleware and characterize its type or, in other words, define its architecture.
- Architectural styles aid in understanding the classifying the organization of the software systems in general and distributed computing in particular.
- The use of well-known standards at the operating system level and even more at the hardware and network levels allows easy harnessing of heterogeneous components and their organization into a coherent and uniform system.
- For example; network connectivity between different devices is controlled by standards, which allow them into interact seamlessly.
- Design patterns help in creating a common knowledge within the community of software engineers and developers as to how to structure the relevant of components within an application and understand the internal organization of software applications.
- Architectural styles do the same for the overall architecture of software systems.
- The architectural styles are classified into two major classes
    - Software Architectural styles : Relates to the logical organization of the software.
    - System Architectural styles: styles that describe the physical organization of distributed software systems in terms of their major components.

Software Architectural Styles

- Software architectural styles are based on the logical arrangement of software components.
- They are helpful because they provide an intuitive view of the whole system, despite its physical deployment.

- They also identify the main abstractions that are used to shape the components of the system and the expected interaction patterns between them.

Data Centered Architectures

- These architectures **identify the data as the fundamental element of the software system,** and access to shared data is the core characteristics of the data-centered architectures.
- Within the context of distributed and parallel computing systems, **integrity of data is overall goal for such systems.**
- The **repository architectural style** is the most relevant reference model in this category. It is characterized by two main components – the central data structure, which represents the current state of the system, and a collection of independent component, which operate on the central data.
- The ways in which the independent components interact with the central data structure can be very heterogeneous.
- In particular repository based architectures differentiate and specialize further into subcategories according to the choice of control discipline to apply for the shared data structure. Of particular interest are databases and blackboard systems.

Black board Architectural Style

- The black board architectural style is characterized by three main components:
  - Knowledge sources: These are entities that update the knowledge base that is maintained in the black board.
  - Blackboard: This represents the data structure that is shared among the knowledge sources and stores the knowledge base of the application.
  - Control: The control is the collection of triggers and procedures that govern the interaction with the blackboard and update the status of the knowledge base.

Data Flow Architectures

- Access to data is the core feature; data-flow styles explicitly incorporate the pattern of data-flow, since their design is determined by an orderly motion of data from component to component, which is the form of communication between them.
- Styles within this category differ in one of the following ways: how the control is exerted, the degree of concurrency among components, and the topology that describes the flow of data.

- **Batch Sequential:** The batch sequential style is characterized by an ordered sequence of separate programs executing one after the other. These programs are chained together by providing as input for the next program the output generated by the last program after its completion, which is most likely in the form of a file. This design was very popular in the mainframe era of computing and still finds applications today. For example, many distributed applications for scientific computing are defined by jobs expressed as sequence of programs that, for example, pre-filter, analyze, and post process data. It is very common to compose these phases using the batch sequential style.
- **Pipe-and-Filter Style:** It is a variation of the previous style for expressing the activity of a software system as sequence of data transformations. Each component of the processing chain is called a filter, and the connection between one filter and the next is represented by a data stream.

Virtual Machine architectures

- The virtual machine class of architectural styles is characterized by the presence of an abstract execution environment (generally referred as a virtual machine) that simulates features that are not available in the hardware or software.
- Applications and systems are implemented on top of this layer and become portable over different hardware and software environments.
- The general interaction flow for systems implementing this pattern is – the program (or the application) defines its operations and state in an abstract format, which is interpreted by the virtual machine engine. The interpretation of a program constitutes its execution. It is quite common in this scenario that the engine maintains an internal representation of the program state.
- Popular examples within this category are rule based systems, interpreters, and command language processors.
- **Rule-Based Style:**
    - This architecture is characterized by representing the abstract execution environment as an inference engine. Programs are expressed in the form of rules or predicates that hold true. The input data for applications is generally represented by a set of assertions or facts that the inference engine uses to activate rules or to apply predicates, thus transforming data. The examples of rule-based systems can be found in the networking domain: Network Intrusion Detection

Systems (NIDS) often rely on a set of rules to identify abnormal behaviors connected to possible intrusion in computing systems.

- Interpreter Style: The presence of engine to interpret the style.

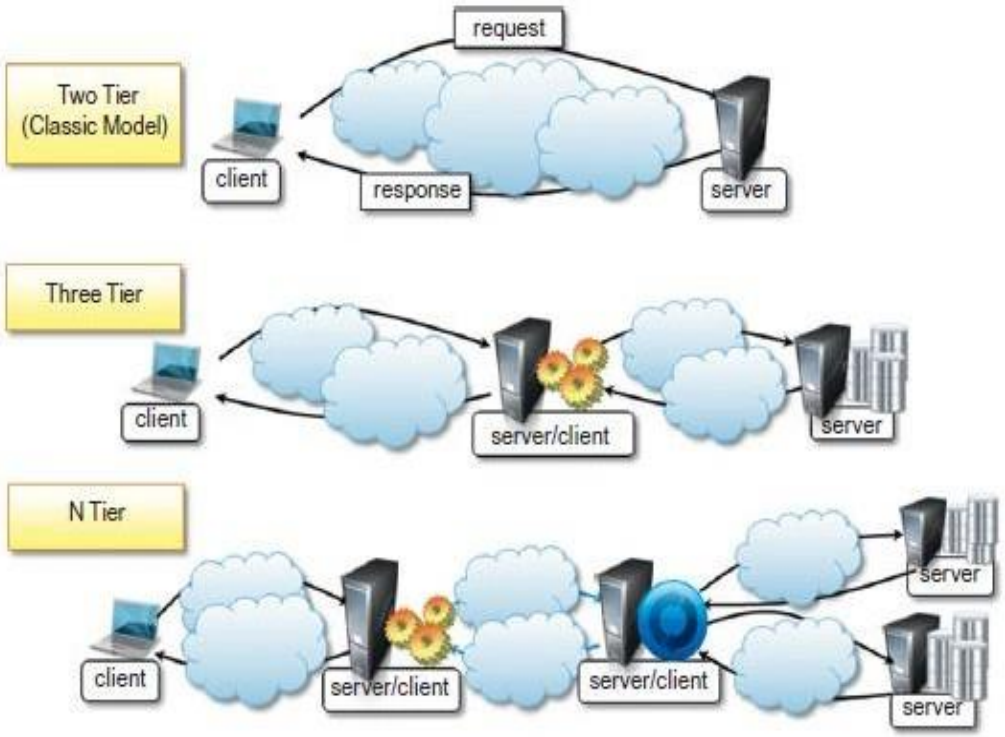
Call and return architectures

- This identifies all systems that are organized into components mostly connected together by method calls.
- The activity of systems modeled in this way is characterized by a chain of method calls whose overall execution and composition identify the execution one or more operations.
- There are three categories in this
  - Top down Style : developed with imperative programming
  - Object Oriented Style: Object programming models
  - Layered Style: provides the implementation in different levels of abstraction of the system.

System Architectural Styles

- System architectural styles cover the physical organization of components and processes over a distributed infrastructure.
- Two fundamental reference style
  - Client / Server
  - Peer- to -Peer
  - The information and the services of interest can be centralized and accessed through a single access point: the server.
  - Multiple clients are interested in such services and the server must be appropriately designed to efficiently serve requests coming from different clients.

Client / Server architectural Styles



- Symmetric architectures in which all the components, called peers, play the same role and incorporate both client and server capabilities of the client/server model.
- More precisely, each peer acts as a server when it processes requests from other peers and as a client when it issues requests to other peers.

Peer-to-Peer architectural Style

d.Models for Inter process Communication

- Distributed systems are composed of a collection of concurrent processes interacting with each other by means of a network connection.
- IPC is a fundamental aspect of distributed systems design and implementation.
- IPC is used to either exchange data and information or coordinate the activity of processes.
- IPC is what ties together the different components of a distributed system, thus making them act as a single system.
- There are several different models in which processes can interact with each other – these maps to different abstractions for IPC.
- Among the most relevant that we can mention are shared memory, remote procedure call (RPC), and message passing.
- At lower level, IPC is realized through the fundamental tools of network programming.
- Sockets are the most popular IPC primitive for implementing communication channels between distributed processes.

Message-based communication

- The abstraction of message has played an important role in the evolution of the model and technologies enabling distributed computing.
- The definition of distributed computing – is the one in which components located at networked computers communicate and coordinate their actions only by passing messages. The term messages, in this case, identify any discrete amount of information that is passed from one entity to another. It encompasses any form of data representation that is limited in size and time, whereas this is an invocation to a remote procedure or a serialized object instance or a generic message.
- The term message-based communication model can be used to refer to any model for IPC.
- Several distributed programming paradigms eventually use message-based communication despite the abstractions that are presented to developers for programming the interactions of distributed components.
- Here are some of the most popular and important:

Message Passing: This paradigm introduces the concept of a message as the main abstraction of the model. The entities exchanging information explicitly encode in the form of a message the data to be exchanged. The structure and the content of a message vary according to the model. Examples of this model are the Message-Passing-Interface (MPI) and openMP.

- **Remote Procedure Call (RPC):** This paradigm extends the concept of procedure call beyond the boundaries of a single process, thus triggering the execution of code in remote processes.
- **Distributed Objects:** This is an implementation of the RPC model for the object-oriented paradigm and contextualizes this feature for the remote invocation of methods exposed by objects. Examples of distributed object infrastructures are Common Object Request Broker Architecture (CORBA), Component Object Model (COM, DCOM, and COM+), Java Remote Method Invocation (RMI), and .NET Remoting.
- **Distributed agents and active Objects:** Programming paradigms based on agents and active objects involve by definition the presence of instances, whether they are agents of objects, despite the existence of requests.
- **Web Service**: An implementation of the RPC concept over HTTP; thus allowing the interaction of components that are developed with different technologies. A Web service is exposed as a remote object hosted on a Web Server, and method invocation are transformed in HTTP requests, using specific protocols such as Simple Object Access Protocol (SOAP) or Representational State Transfer (REST).

e. Technologies for distributed computing

- Remote Procedure Call (RPC)
  - RPC is the fundamental abstraction enabling the execution procedures on clients' request.
  - RPC allows extending the concept of a procedure call beyond the boundaries of a process and a single memory address space.
  - The called procedure and calling procedure may be on the same system or they may be on different systems..
- Distributed Object Frameworks

– Extend object-oriented programming systems by allowing objects to be distributed across a heterogeneous network and provide facilities so that they can be coherently act as though they were in the same address space.

Web Services

- Web Services are the prominent technology for implementing SOA systems and applications.
- They leverage Internet technologies and standards for building distributed systems.
- Several aspects make Web Services the technology of choice for SOA.
- First, they allow for interoperability across different platforms and programming languages.
- Second, they are based on well-known and vendor-independent standards such as HTTP, SOAP, and WSDL.
- Third, they provide an intuitive and simple way to connect heterogeneous software systems, enabling quick composition of services in distributed environment.

PARALLEL COMPUTING	DISTRIBUTED COMPUTING
Many operations are performed simultaneously	System components are located at different locations
Single computer is required	Uses multiple computers
Multiple processors perform multipleoperations	Multiple computers perform multiple operations
It may have shared or distributedmemory	It have only distributed memory
Processors communicate with eachother through bus	Computer communicate with each other throughmessage passing.
Improves the system performance	Improves system scalability, fault tolerance andresource sharing capabilities

.

ELASTICITY IN CLOUD COMPUTING

- Elasticity is defined as the ability of a system to add and remove resources (such as CPU cores, memory, VM and container instances) to adapt to the load variation in real time.
- Elasticity is a dynamic property for cloud computing.
- Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible.

Elasticity = Scalability + Automation + Optimization

- Elasticity is built on top of scalability.
- It can be considered as an automation of the concept of scalability and aims to optimize at best and as quickly as possible the resources at a given time.
- Another term associated with elasticity is the efficiency, which characterizes how cloud resource can be efficiently utilized as it scales up or down.
- It is the amount of resources consumed for processing a given amount of work, the lower this amount is, the higher the efficiency of a system.
- Elasticity also introduces a new important factor, which is the speed.
- Rapid provisioning and deprovisioning are key to maintaining an acceptable performance in the context of cloud computing
- Quality of service is subjected to a service level agreement

Purpose

- Elasticity aims at matching the amount of resource allocated to a service with theamount of resource it actually requires, avoiding over- or under-provisioning.
  - **Over-provisioning**, i.e., allocating more resources than required, should be avoided as the service provider often has to pay for the resources that are allocated to the service. For example, an Amazon EC2 M4 extra-large instance costs US$0.239/hour. If a service has allocated two virtual machines

    when only one is required, the service provider wastes $2,095 every year. Hence, the service provider's expenses are higher than optimal and their profit is reduced.
  - **Under-provisioning**, i.e., allocating fewer resources than required, must be avoided, otherwise the service cannot serve its users with a good service. In the above example, under-provisioning the website may make it seem slowor unreachable. Web users eventually give up on accessing it, thus, the service provider loses customers. On the long term, t he provider's income will decrease, which also reduces their profit.

Evaluation

When evaluating elasticity, the following points need to be checked earlier: **Autonomic Scaling**:

What adaptation process is used for autonomic scaling?**Elasticity Dimensions**: What is the set of resource types scaled as part of the adaptation process?

- **Resource Scaling Units**: For each resource type, in what unit is the amountof allocated resources varied?
- **Scalability Bounds:** For each resource type, what is the upper bound onthe amount of resources that can be allocated?

Amazon Elastic cloud compute (EC2)

- It is a heart of the Amazon cloud. It provides a web services API for provisioning, managing, and deprovisioning virtual servers inside the Amazon cloud. In other word, any application anywhere on the Internet can launch a virtual server in the Amazon cloud with a single web services call.
- When you want to start up a virtual server in the Amazon environment, you launcha new node based on a predefined Amazon machine inage (AMI). The AMI includes your OS and any other prebuilt software.

Classification

Elasticity solutions can be arranged in different classes based on

- Scope
- Policy
- Purpose
- Method

a.Scope

- Elasticity can be implemented on any of the cloud layers.
- Most commonly, elasticity is achieved on the IaaS level, where the resources to be provisioned are virtual machine instances.
- Other infrastructure services can also be scaled
- On the PaaS level, elasticity consists in scaling containers or databases for instance.
- Finally, both PaaS and IaaS elasticity can be used to implement elastic applications, be it for private use or in order to be provided as a SaaS
- The elasticity actions can be applied either at the infrastructure or application/platform level.
- The elasticity actions perform the decisions made by the elasticity strategy or management system to scale the resources.
- Google App Engine and Azure elastic pool are examples of elastic Platform as a Service (PaaS).
- Elasticity actions can be performed at the infrastructure level where the elasticity controller monitors the system and takes decisions.
- The cloud infrastructures are based on the virtualization technology, which can be VMs or containers.

- In the embedded elasticity, elastic applications are able to adjust their own resources according to runtime requirements or due to changes in the execution flow.
- There must be a knowledge of the source code of the applications.
- Application Map: The elasticity controller must have a complete map of the application components and instances.
- Code embedded: The elasticity controller is embedded in the application source code.
- The elasticity actions are performed by the application itself.
- While moving the elasticity controller to the application source code eliminates the use of monitoring systems
- There must be a specialized controller for each application.

b.Policy

- Elastic solutions can be either manual or automatic.
- A manual elastic solution would provide their users with tools to monitor their systems and add or remove resources but leaves the scaling decision to them.

Automatic mode: All the actions are done automatically, and this could be classified into reactive and proactive modes.

Elastic solutions can be either reactive or predictive

Reactive mode: The elasticity actions are triggered based on certain thresholds or rules, the system reacts to the load (workload or resource utilization) and triggers actions to adapt changes accordingly.

- An elastic solution is reactive when it scales a posteriori, based on a monitored change in the system.
- These are generally implemented by a set of Event-Condition-Action rules.

Proactive mode: This approach implements forecasting techniques, anticipates the future needs and triggers actions based on this anticipation.

- A predictive or proactive elasticity solution uses its knowledge of either recent history or load patterns inferred from longer periods of time in order to predict the upcoming load of the system and scale according to it.

c.Purpose

- An elastic solution can have many purposes.
- The first one to come to mind is naturally performance, in which case the focus should be put on their speed.

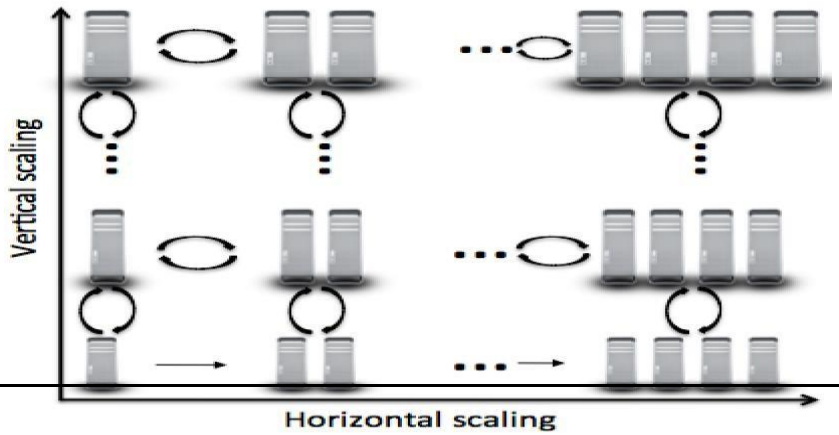
- Another purpose for elasticity can also be energy efficiency, where using the minimum amount of resources is the dominating factor.
- Other solutions intend to reduce the cost by multiplexing either resource providers or elasticity methods
- Elasticity has different purposes such as improving performance, increasing resource capacity, saving energy, reducing cost and ensuring availability.
- Once we look to the elasticity objectives, there are different perspectives.
- Cloud IaaS providers try to maximize the profit by minimizing the resources while offering a good Quality of Service (QoS),
- PaaS providers seek to minimize the cost they pay to the

Cloud.

- The customers (end-users) search to increase their Quality of Experience (QoE) and to minimize their payments.
- QoE is the degree of delight or annoyance of the user of an application or service

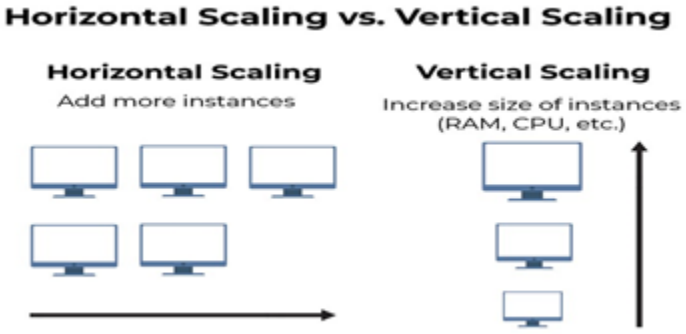
d.Method

- Vertical elasticity, changes the amount of resources linked to existing instances on-the-fly.
- This can be done in two manners.
- The first method consists in explicitly redimensioning a virtual machine instance, i.e., changing the quota of physical resources allocated to it.
- This is however poorly supported by common operating systems as they fail to take into account changes in CPU or memory without rebooting, thus resulting in service interruption.
- The second vertical scaling method involves VM migration: moving a virtual machine instance to another physical machine with a different overall load changes its available resources



- Horizontal scaling is the process of adding/removing instances, which may be located at different locations.
- Load balancers are used to distribute the load among the different instances.
- Vertical scaling **is** the process of modifying resources (CPU, memory, storage or both) size for an instance at run time.
- It gives more flexibility for the cloud systems to cope with the varying workloads

- Horizontal scaling refers to provisioning additional servers to meet your needs, often splitting workloads between servers to limit the number of requests any individual server is getting. Horizontal scaling in cloud computing means adding additional instances instead of moving to a larger instance size.
- Vertical scaling refers to adding more or faster CPUs, memory, or I/O resources to an existing server, or replacing one server with a more powerful server. In a data center, administrators traditionally achieved vertical scaling by purchasing a new, more powerful server and discarding or repurposing the old one. Today's cloud architects can accomplish AWS vertical scaling and Microsoft Azure vertical scaling by changing instance sizes. AWS and Azure cloud services have many different instance sizes, so vertical scaling in cloud computing is possible for everything from EC2 instances to RDS databases.



HORIZONTAL	VERTICAL
1. Load balancing required	1. Load balancing unnecessary
2. Resilient to system failure	2. Single point of failure
3. Utilizes Network Calls	3. Inter process communication
4. Data Inconsistency	4. Data consistent
5. Scales well	5. Hardware limit

Migration

- Migration can be also considered as a needed action to further allow the vertical scaling when there is no enough resources on the host machine.

- It is also used for other purposes such as migrating a VM to a less loaded physical machine just to guarantee its performance.
- Several types of migration are deployed such as live migration and no-live migration.
- Live migration has two main approaches
    - post-copy
    - pre-copy
- Post-copy migration suspends the migrating VM, copies minimal processor state to the target host, resumes the VM and then begins fetching memory pages from the source.
- In pre-copy approach, the memory pages are copied while the VM is running on the source.
- If some pages are changed (called dirty pages) during the memory copy process, they will be recopied until the number of recopied pages is greater than dirty pages, or the source VM will be stopped.
- The remaining dirty pages will be copied to the destination VM.

Architecture

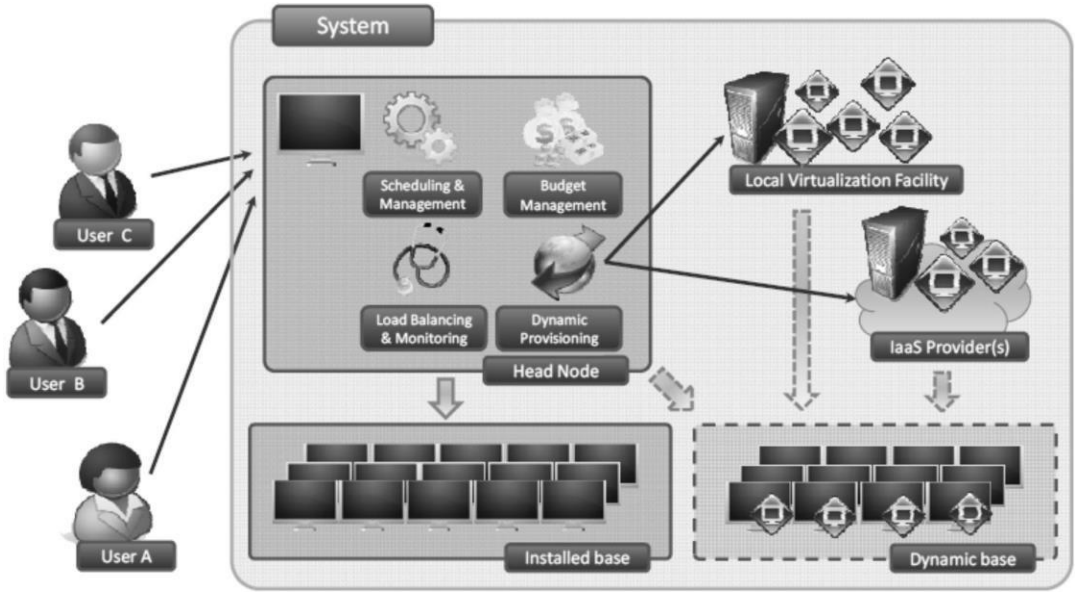
- The architecture of the elasticity management solutions can be either centralized or decentralized.
- Centralized architecture has only one elasticity controller, i.e., the auto scaling system that provisions and deprovisions resources.
- In decentralized solutions, the architecture is composed of many elasticity controllers or application managers, which are responsible for provisioning resources for different cloud-hosted platforms

Provider

- Elastic solutions can be applied to a single or multiple cloud providers.
- A single cloud provider can be either public or private with one or multiple regions or datacenters.
- Multiple clouds in this context means more than one cloud provider.
- It includes hybrid clouds that can be private or public, in addition to the federated clouds and cloud bursting.
- Most of the elasticity solutions support only a single cloud provider

On-demand Provisioning.

- **Dynamic provisioning / On demand Provisioning**
  Dynamic provisioning refers to the ability of dynamically acquiring new resources and integrating them into the existing infrastructure and software system. Resources can be of different nature: hardware or software. In the most common cases resources are virtual machines of software services and they identify two very well known segments in the Cloud Computing

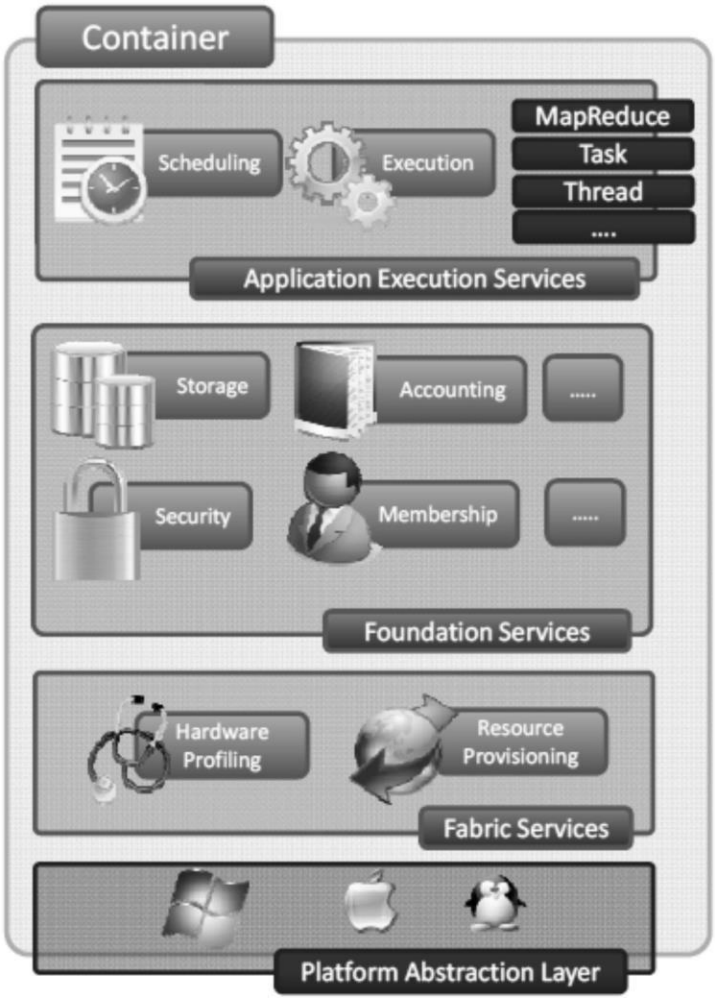


market:Infrastructure-as-a-Service and Software-as-a-Service.

Aneka Service Model

- Aneka provides its services for application execution by deploying on each managed computing node a software container, called Aneka Container, which hosts a collection of services. The Aneka Container is in charge of managing the basic interaction with the hosting environment and operating system and provides accessibility to the services that constitute the "*intelligence*" and the "*horse power*" of the Aneka Cloud.
- Aneka provides a large collection of services that cover all the needs for managing a distributed computing platform. Some of these services are part of the infrastructure and not of direct interest for end users others are primarily designed for interacting with users and applications. Among these services we can consider:
  - *Scheduling and Execution services*: they are configured and managed as a pair and control the execution of applications.
  - *Storage services*: they are in charge of the management of data and provide an internal storage for applications.
  - *Reporting, Logging and Monitoring services*: they are configured and managed for collecting information about users, applications, andsystem components.
  - *Security services*: they enforce a specific level of security in the Aneka Cloud.

Figure provides a reference model of the Aneka Container together with some of the services that characterize a possible configuration. Aneka containers can be configured and deployed by using the Aneka Cloud Management Studio, which providesthe basic configuration templates (*Master* and *Worker*) that are commonly used to setup Aneka Clouds as well as advanced capabilities for customizing these templates.



Container architecture.

- The installation and the configuration of the dynamic provisioning infrastructure mostly refer to the configuration of the Master template, and of particular interest are the *Scheduling Services* (*Task* and *Thread Programming Models*) and the Resource Provisioning Service. There are also other services involved in the provisioning of dynamic resources, but these services are not under the control of the user and operate as part of infrastructure.

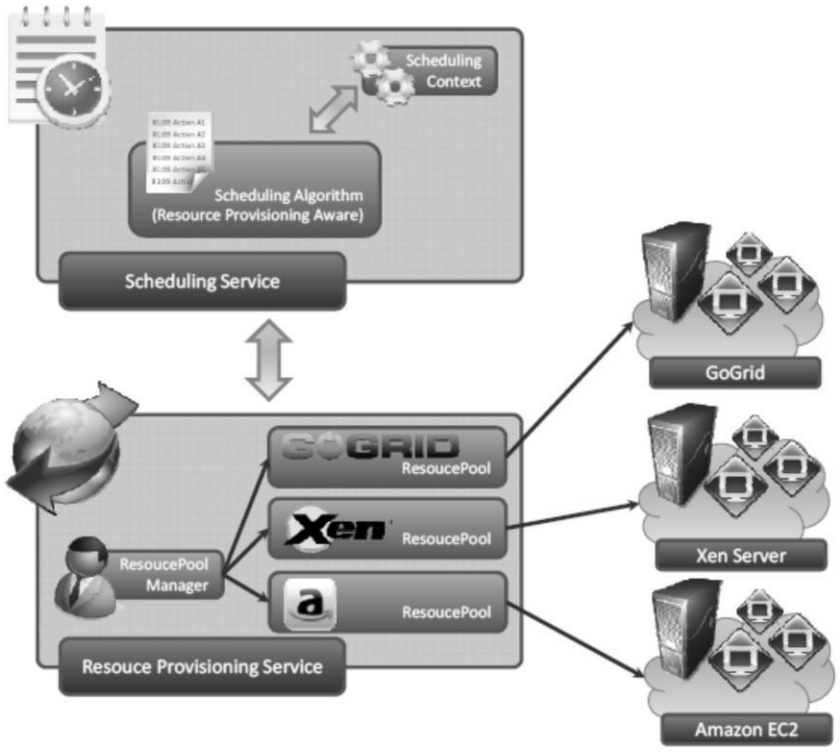
Dynamic Provisioning Infrastructure Overview

- Figure shows an overview of the dynamic provisioning infrastructure implemented in Aneka. The dynamic provisioning infrastructure is mainly the result of the collaboration between two services: the *Scheduling Service* and the *Resource Provisioning Service*. The former triggers the provisioning request according to the status of the system and the requirements of applications while the latter is in charge of making the provisioning happen by interacting with IaaS providers and local virtualization facilities.

Resource Provisioning means the selection, deployment, and run-time management of software (e.g., database server management systems, load balancers) and hardware resources (e.g., CPU, storage, and network) for ensuring guaranteed performance for applications.

- Resource Provisioning is an important and challenging problem in the large-scale distributed systems such as Cloud computing environments.
- There are many resource provisioning techniques, both static and dynamic each one having its own advantages and also some challenges.

- These resource provisioning techniques used must meet Quality of Service (QoS) parameters like availability, throughput, response time, security, reliability etc., and thereby avoiding Service Level Agreement (SLA) violation.
- Over provisioning and under provisioning of resources must be avoided.
- Another important constraint is power consumption.
- The ultimate goal of the cloud user is to minimize cost by renting the resources and from the cloud service provider's perspective to maximize profit by efficiently allocating the resources.
- In order to achieve the goal, the cloud user has to request cloud service provider to make a provision for the resources either statically or dynamically.



- So that the cloud service provider will know how many instances of the resources and what resources are required for a particular application.

Dynamic provisioning infrastructure overview

- By provisioning the resources, the QoS parameters like availability, throughput, security, response time, reliability, performance etc must be achieved without violating SLA

There are two types

- **Static Provisioning**

- **Dynamic Provisioning**

Static Provisioning

- For applications that have predictable and generally unchanging demands/workloads, it is possible to use “static provisioning" effectively.
- With advance provisioning, the customer contracts with the provider for services.
- The provider prepares the appropriate resources in advance of start of service.
- The customer is charged a flat fee or is billed on a monthly basis.

Dynamic Provisioning

- In cases where demand by applications may change or vary, “dynamic provisioning" techniques have been suggested whereby VMs may be migrated on-the-fly to new compute nodes within the cloud.
- The provider allocates more resources as they are needed and removes them when they are not.
- The customer is billed on a pay-per-use basis.
- When dynamic provisioning is used to create a hybrid cloud, it is sometimes referred to as cloud bursting.

Parameters for Resource Provisioning

- Response time
- Minimize Cost
- Revenue Maximization
- Fault tolerant
- Reduced SLA Violation
- Reduced Power Consumption

Response time: The resource provisioning algorithm designed must take minimal time to respond when executing the task.

Minimize Cost: From the Cloud user point of view cost should be minimized.

Revenue Maximization: This is to be achieved from the Cloud Service Provider’s view.

Fault tolerant: The algorithm should continue to provide service in spite of failure of nodes.

Reduced SLA Violation: The algorithm designed must be able to reduce SLA violation.

Reduced Power Consumption: VM placement & migration techniques must lower power consumption

Dynamic Provisioning Types

1. Local On-demand Resource Provisioning

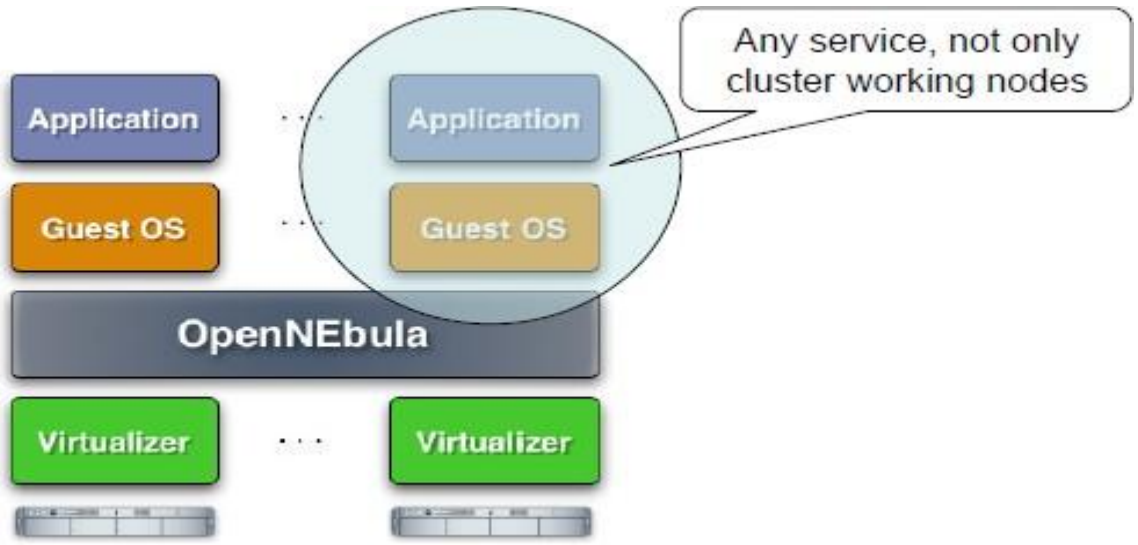
2. Remote On-demand Resource Provisioning

Local On-demand Resource Provisioning

1. The Engine for the Virtual Infrastructure

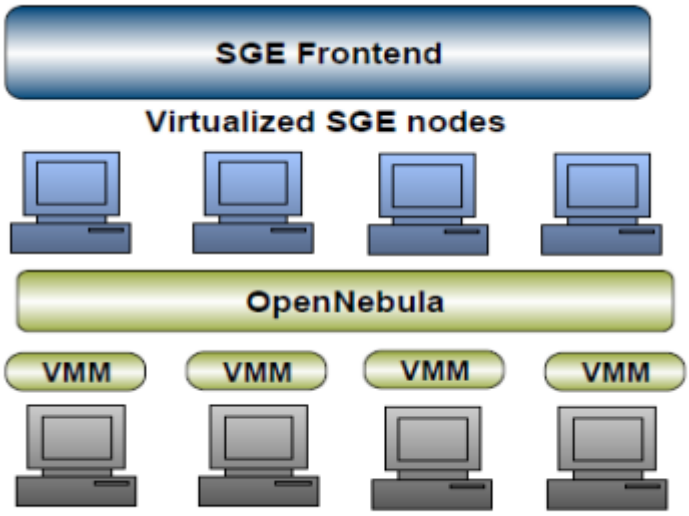
The OpenNebula Virtual Infrastructure Engine

- OpenNEbula creates a distributed virtualization layer
  - Extend the benefits of VM Monitors from one to multiple resources
  - Decouple the VM (service) from the physical location
- Transform a distributed physical infrastructure into a flexible and elastic virtual infrastructure, which adapts to the changing demands of the VM (service) workloads



Separation of Resource Provisioning from Job Management

- New virtualization layer between the service and the infrastructure layers
- Seamless integration with the existing middleware stacks.
- Completely transparent to the computing service and so end users



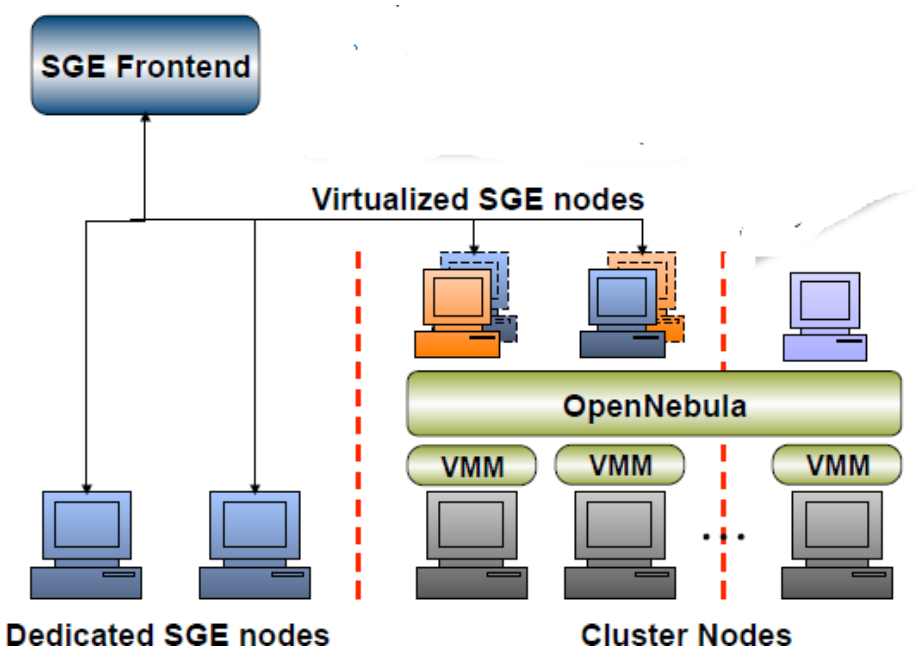
Cluster Partitioning

- Dynamic partition of the infrastructure
- Isolate workloads (several computing clusters)
- Dedicated HA partitions

Benefits for Existing Grid Infrastructures

- The **virtualization of the local infrastructure** supports a virtualized alternative to contribute resources to a Grid infrastructure
    - Simpler deployment and operation of new middleware distributions
    - Lower operational costs
    - Easy provision of resources to more than one infrastructure
    - Easy support for VO-specific worker nodes

  Performance partitioning between local and grid clusters



-

Other Tools for VM Management

- VMware DRS, Platform Orchestrator, IBM Director, Novell ZENworks, Enomalism, Xenoserver
- **Advantages**:
  - Open-source (Apache license v2.0)
  - Open and flexible architecture to integrate new virtualization technologies
  - Support for the definition of any scheduling policy (consolidation, workload balance, affinity, SLA)
  - LRM-like CLI and API for the integration of third-party tools

Remote on-Demand Resource Provisioning

Access to Cloud Systems

- Provision of virtualized resources as a service

VM Management Interfaces

The processes involved are

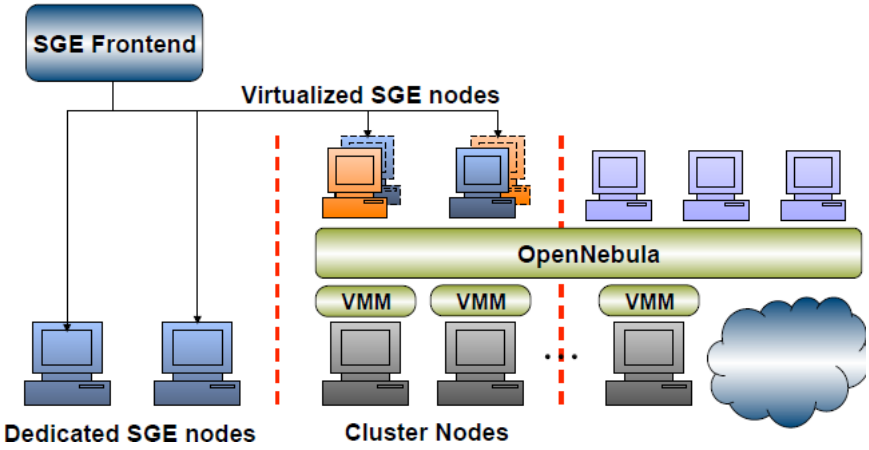
- Submission
- Control
- Monitoring

Infrastructure Cloud Computing Solutions

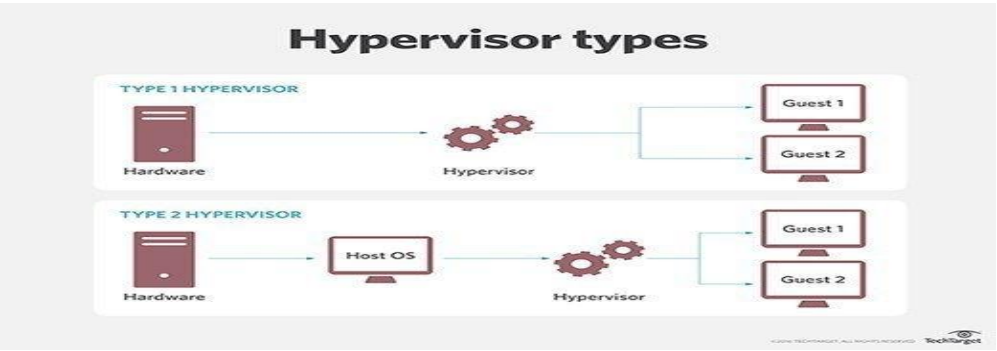
- Commercial Cloud: Amazon EC2
- Scientific Cloud: Nimbus (University of Chicago)
- Open-source Technologies
    - Globus VWS (Globus interfaces)
    - Eucalyptus (Interfaces compatible with Amazon EC2)
    - OpenNEbula (Engine for the Virtual Infrastructure)

On-demand Access to Cloud Resources

- Supplement local resources with cloud resources to satisfy peak or fluctuating demands

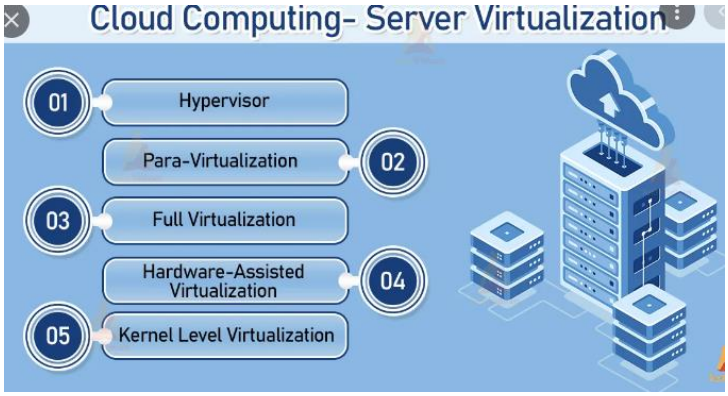


- A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware. A Cloud Hypervisor is software that enables the sharing of cloud provider's physical compute and memory resources across multiple virtual machines (VMs). There are two main hypervisor types, referred to as “Type 1” (or “bare metal”) and “Type 2” (or “hosted”). A type 1 hypervisor acts like a lightweight operating system and runs directly on the host's hardware, while a type 2 hypervisor runs as a software layer on an operating system, like other computer programs. he hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. A hypervisor is sometimes also called a virtual machine manager(VMM).



What is Server Virtualization in Cloud Computing?

Server Virtualization is the process of dividing a physical server into several individuals and isolated virtual servers with software applications. Every virtual server can run its own operating systems individually.



Why Server Virtualization?

Server Virtualization is one of the most cost-effective methods to offer Web hosting services and uses the existing resources effectively in IT Infrastructure.

If there is no server Virtualization, the servers will only use a tiny section of their processing power. It will result in idle servers because the workload is divided into one portion of the network servers.

Data centers have become overcrowded with unutilized servers, resulting in wasting resources and heavy power consumption.

By having every physical server divided into multiple virtual servers, server virtualization will authorize each virtual server to behave as a unique device.

Every Virtual Server is capable of running its own application and operating systems.

The following process helps to increase resource Utilization by creating each virtual server to behave as a physical server, and it develops the capacity of every physical device.

Key Benefits of Server Virtualization

- Server Virtualization contains higher server capability
- Organizations experience cheaper operational cost
- It eliminates the complexity of the server
- It helps in developing the application performance
- With Software Virtualization, users can deploy their workload swiftly.

Types of Server Virtualizations in Computing Network

1. Hypervisor in cloud computing

The Hypervisor is also popularly known as Virtual Machine Monitor. Basically, it is a layer that lies in between Operating System and Hardware. It offers essential services and features for a smooth process to run multiple operating systems.

It will identify the traps, respond to the priority CPU instructions, handle the ques, and dispatch and answer the requests of Hardware.

A Host Operating System will also work on the top of the Hypervisor to administer and manage the Virtual Machines.

2. Para-Virtualization in cloud computing

Para-Virtualization is dependent on Hypervisor. In this Virtualization model, many emulated and trapping overhead software are implemented and handled.

The guest Operating System will be modified and recompiled before the installation process inside the virtual machine. Because of this modification in the Guest Operating system, the performance increases, and the system will communicate directly with the Hypervisor, and the Emulation overhead gets eliminated.

Advantages

- Para-Virtualization is easy
- It has no enhanced performance
- It has no Emulation overhead.

Disadvantages

- Para-Virtualization requires modification to run the guest Operating System.

3. Full Virtualization in cloud computing

Full Virtualization is almost similar to Para-Virtualization. It has the capability to simulate the underlying Hardware when required. The Hypervisor will trap the machine operations done by the Operating Systems to perform I/O operations or modify the system's status.

Once the trapping is completed, the following operations will be simulated in the software, and the status codes will be returned invariably as per the hardware delivery. It is the key reason why a whole operating system is capable of running on top of Hypervisor.

Advantages

- In total Virtualization Guest Operating System does not require modification.

Disadvantages

- Full Virtualization is complex
- It is slower
- Installing a new device is challenging

4. Hardware-Assisted Virtualization in Cloud Computing

Hardware-Assisted Virtualization can be related to Full Virtualization and Paravirtualization in operational terms except that it requires Hardware support.

Many Hypervisors are overhead because of trapping and Emulation of I/O Operations, and status instructions get executed and processed within the guest OS. It deals by depending upon the hardware extensions of x86 Architecture.

Also, unmodified OS can be run as the Hardware support for Virtualization purposes. Secondly, it can be utilized to manage the hardware access and it will request the privileged and protected operations to communicate with the Virtual Machine.

Advantages

- In Hardware-Assisted Virtualization, the Guest Operating System has to perform any modifications.
- The Hypervisor is overhead with a lower rate.

Disadvantages

- Hardware-Assisted Virtualization requires Hardware support.

5. Kernel Level Virtualization in Cloud Computing

Apart from using a Hypervisor, it will run a different version of Linux Kernel, and it is associated with Virtual Machine as a user's space process on the physical machine.

It makes it easy to run several virtual machines on a single host device. A device driver is used to establish communication between the main Linux Kernel and virtual machine. But remember, Processor support is essential for Virtualization.

A slightly modified QEMU process is made and utilized to display and execute the containers for virtual machines.

In multiple methods, the kernel Level Virtualization is a specialized form of Server Virtualization.

Advantages

- In Kernel Level Virtualization, there is no requirement for administrative software.
- It has significantly less overhead.

Disadvantages

Linux Kernel Virtualization requires Hardware Support.

6. System Level Virtualization in Cloud Computing

This will run several but logically separated environments in a single machine of the operating system kernel. It is also known as the shared kernel approach, as all virtual machines share the same kernel to host the operating system. It is dependent on the change root concept, and it is known as "chroot."

Basically, "chroot" will begin during system boot-up. The kernel will make use of root filesystems to load the drivers and perform several initial stage system tasks.

It will later switch into another root filesystem using the chroot command to mount on an -disk file system and the final root filesystem. And it will continue system initialization and configure within the file system.

The chroot mechanism of system Level Virtualization is considered as an extension of the following concept. It will allow the system to begin the virtual servers by their processors, executing its relative own filesystem root directories.

The critical difference between system Level and Server Virtualization is that multiple operating systems can run on various virtual systems.

Suppose all the virtual servers share an exact copy of the operating system. In that case, it is known as system Level Virtualization, and if various servers have multiple operating systems it is titled Server Virtualization.

Advantages of system level virtualization

- System-Level Virtualization is lightweight compared to entire machines.
- It can host various virtual servers
- It consists of developed security and isolation

Disadvantages of system level virtualization

How does Server Virtualization Works?

In Server Virtualization, the virtual servers are wholly dedicated to performing a specific task. Each virtual server behaves like a physical device that is capable of running its Operating System.

The administrator of the software is capable of converting one physical server into various virtual servers.

Later these several servers are capable of using the processing power of all machines.
The computer's CPU can run various processors, which offer the capability to run many complex tasks efficiently.

Advantages of Server Virtualization

1. Economic

When the division takes place in the physical server is partitioned into multiple virtual machines, they then further and then several operating system instances can launch, operate, and be managed inside the physical servers.

Thus, using a less physical server will help in reducing the cost of the server, and as a result, it will benefit the overall business.

2. Disaster Recovery

The data virtually travels from one server to another quickly and safely. Users can store the data at any location and retrieve it back from anywhere. The following process takes less time, and also the downtime is reduced.

3. Increased Productivity

If the physical servers are fewer, then they are easy to manage. So there are several tools available and will convert the services smoothly.

4. Quick Deployment and Provisioning

In some seconds, the user is capable of performing provisioning and deployment tasks. Server Virtualization permits replicating the existing Virtual Machine.

5. Energy Consumption is reduced.

Less Physical servers will lead to lower consumption energy as they produce a lot of heat if it is not used efficiently. It raises an important issue because of environmental changes.

6. Affordable Web Hosting

When multiple servers are adjusted on one computer, the servers rise virtually, and there is no additional expenditure.

7. Increased Efficiency

As there are fewer physical servers, the IT team will focus on those specific servers and it will thereby benefit them to focus on critical tasks and increase their efficiency.

8. Independent Restart of Each Server

Every server can be restarted individually, which will not affect other working servers.

9. Centralized Management

Software Virtualization has centralized management, and there is Full compatibility with the applications.

10. Migration

In Server Virtualization, live Migration is possible from the servers, and it can be carried out very quickly.

11. Easy to Update

In Server Virtualization, installing, updating, and setting up the software is done efficiently.

Disadvantages of Server Virtualization

- If any servers go offline because of technical issues, the website hosted with its help will also be down.
- Users must manage the Virtual Machines accurately by configuring and monitoring the movements.
- The RAM requirement is more as RAM consumption increases as every Virtual Machine will occupy its location.
- The disk space also increases during the increase of files in virtual machines.
- Several links in a Single-chain must work together.
- When several virtual machines perform their operations together under the same host, their performance might get affected.
- The cost of software licenses may be an issue.

UNIT 2 CLOUD ENABLING TECHNOLOGIES

ServiceOrientedArchitecture–RESTandSystemsof Systems–WebServices - Publish Subscribe Model – Basics of Virtualization – Types of Virtualization Implementation Levels of Virtualization – Virtualization Structures – Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices – Virtualization Support and Disaster Recovery.

2.1 SERVICE-ORIENTED ARCHITECTURE

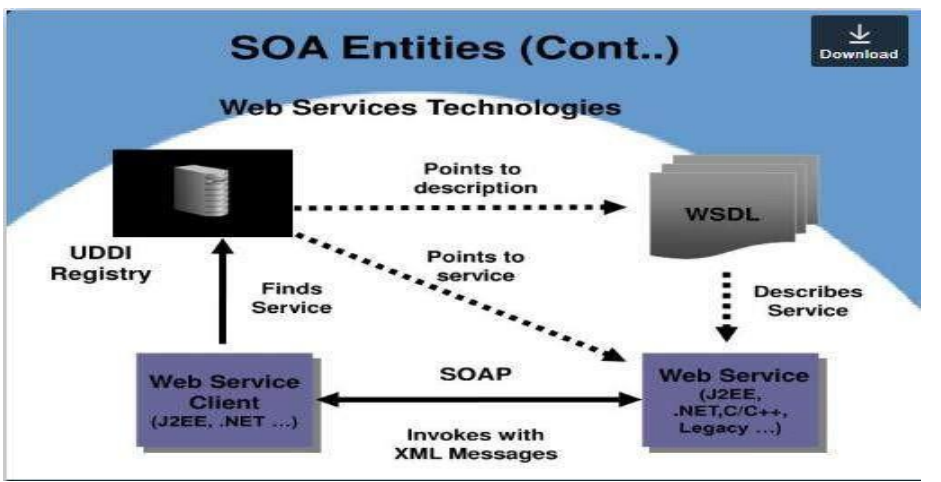
The World Wide Web Consortium (W3C) defines SOA as a form of distributed systems architecture characterized by the following properties:

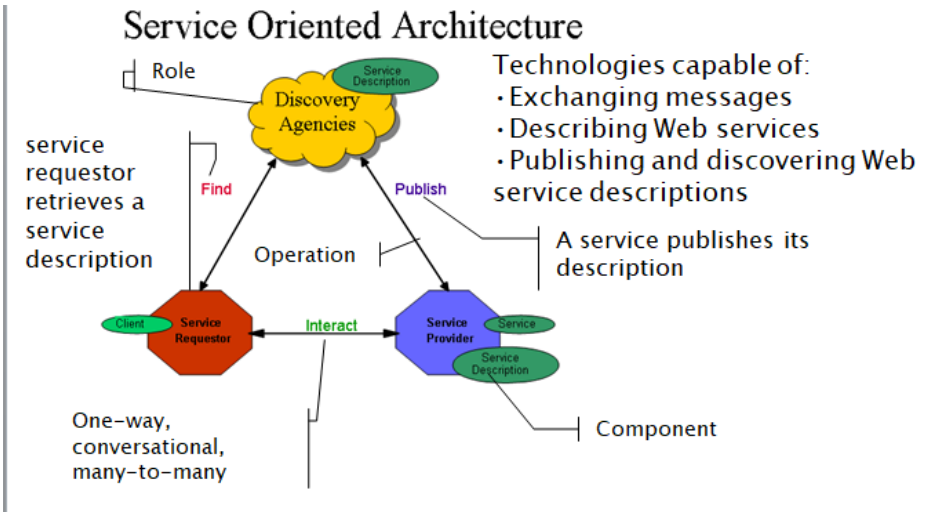
- **Logical view:** The SOA is an abstracted, logical view of actual programs, databases, business processes, and so on, defined in terms of what it does, typically carrying out a business-level operation. The service is formally defined in terms of the messages exchanged between provider agents and requester agents.
  - **Message orientation:** The internal structure of providers and requesters include the implementation language, process structure, and even database structure. These features are deliberately abstracted away in the SOA. A key benefit of this concerns legacy systems. By avoiding any knowledge of the internal structure of an agent, one can incorporate any software component or application to adhere to the formal service definition.
- **Description orientation:** A service is described by machine-executable metadata. The description supports the public nature of the SOA: Only those details that are exposed to the public and are important for the use of the service should be included in the description.
- The semantics of a service should be documented, either directly or indirectly, by its description.
  - □ Granularity Services tend to use a small number of operations with relatively large and complex messages.
  - □ Network orientation Services tend to be oriented toward use over a network, though this is not an absolute requirement.
  - □ Platform-neutral Messages are sent in a platform-neutral, standardized format delivered throughthe interfaces. XML is the most obvious format

that meets this constraint.

- Unlike the component-based model, which is based on design and development of tightly coupled components for processes within an enterprise, using different protocols and technologies such as CORBA and DCOM, SOA focuses on loosely coupled software applications running across different administrative domains, based on common protocols and technologies, such as HTTP and XML.
- SOA is related to early efforts on the architecture style of large-scale distributed systems, particularly Representational State Transfer (REST).
- REST still provides an alternative to the complexstandard-driven web servicestechnology and is used in many Web 2.0 services.

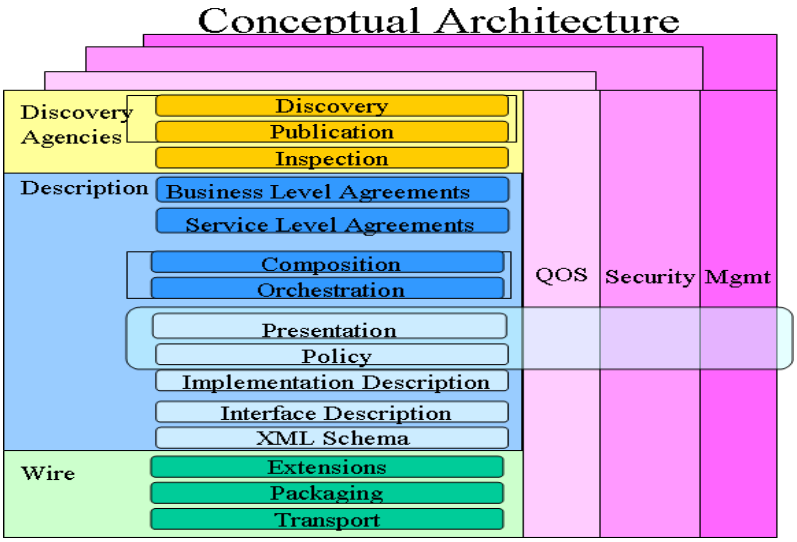
SOA is an architectural approach in which applications make use of services available in the network. In this architecture, services are provided to form applications, through a network call over the internet. It uses common communication standards to speed up and streamline the service integrations in applications.Service-oriented architecture (SOA) is a method of software development that uses software components called services to create business applications. Each service provides a business capability, and services can also communicate with each other across platforms and languages. SOA is used to improve healthcare delivery. Nowadays many apps are games and they use inbuilt functions to run. For example, an app might need GPS so it uses the inbuilt GPS functions of the device. This is SOA in mobile solution.

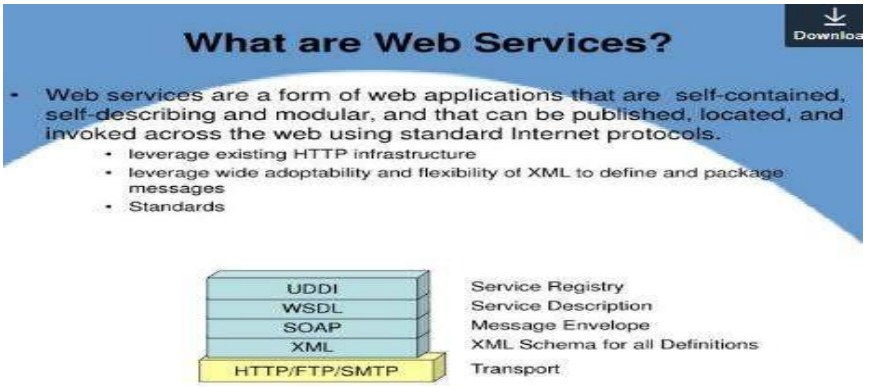




WEB SERVICES

A web service is a standardized method for propagating messages between client and server applications. A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems. Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer.





2.1.1 REST and Systems of Systems

- REST is a software architecture style for distributed systems, particularly distributed hypermedia systems, such as the World Wide Web. It hasrecently gained popularity among enterprises such as Google, Amazon, Yahoo!, and

  especially social networks such as Facebook and Twitter because of its simplicity, and its ease of being published and consumed by clients. REST, shown in Figure 2.1, was introduced and explained by Roy Thomas Fielding, one of the principal authors of the HTTP specification, in his doctoral dissertation in 2000 and was developed in parallel with the HTTP/1.1 protocol.

- The REST architectural style is based on four principles:

  - **Resource Identification through URIs:** The RESTful web service exposes a set of resources which identify targets of interaction with its clients. The key abstraction of information in REST is a resource. Any information that can be named can be a resource, such as a document or image or a temporal service. A resource is a conceptual mapping toa set of entities. Each particular resource is identified by a unique name, or more precisely, a Uniform Resource Identifier (URI) which is of type URL, providing a global addressing space for resources involved in an interaction between components as well as facilitating service discovery. The URIs can be bookmarked or exchanged via hyperlinks, providing more readability and the potential for advertisement.

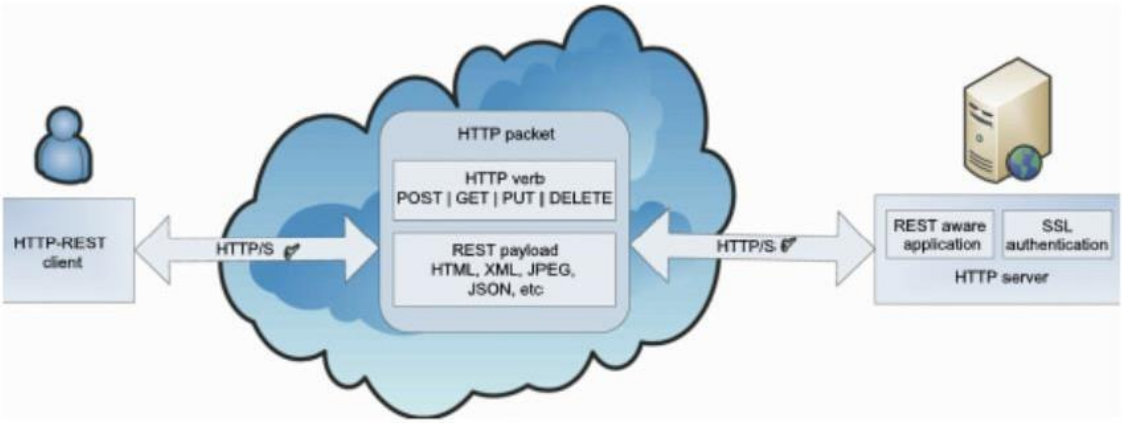


FIGURE: A simple REST interaction between user and server in HTTP specification.

- **Uniform, Constrained Interface:** Interaction with RESTful web services is done via the HTTP standard, client/server cacheable protocol. Resources are manipulated using a fixed set of four CRUD (create, read, update, delete) verbs or operations: PUT, GET, POST, and DELETE. PUT creates a new resource, which can then be destroyed by

  using DELETE. GET retrieves the current state of a resource. POST transfers a new state onto a resource.

- **Self- Descriptive Message:** A REST message includes enough information to describe how to process the message. This enables intermediaries to do more with the message without parsing themessage contents. In REST, resources are decoupled from their representation so that their content can be accessed in a variety of standard formats (e.g., HTML, XML, MIME, plain text, PDF, JPEG, JSON, etc.). REST provides multiple/alternate representations of each resource. Metadata about the resource is available and can be used for various purposes, such as cache control, transmission error detection, authentication or authorization, and access control.

- **Stateless Interactions:** The REST interactions are "stateless" in the sense that the meaning of a message does not depend on the state of the conversation. Stateless communications improve visibility, since a monitoring system does not have to look beyond a single request data field in order to determine the full nature of the request reliability as it

facilitates the task of recovering from partial failures, and increases scalability as discarding state between requests allows the server component to quickly free resources. However, stateless interactions may decrease network performance by increasing the repetitive data (per-interaction overhead). Stateful interactions are based on the concept of explicit state transfer. Several techniques exist to exchange state, such as URI rewriting, cookies, and hidden form fields. State can be embedded in response messages to point to valid future states of the interaction.

- Such lightweight infrastructure, where services can be built with minimal development tools, is inexpensive and easy to adopt. The effort required to build a client to interact with a RESTful service is rather small asdevelopers can begin testing such services from an ordinary web browser, without having to develop custom client-side software.
- From an operational point of view, a stateless RESTful web service is scalable to serve a very large number of clients, as a result of REST support for caching, clustering, and load balancing.
- RESTful web services can be considered an alternative to SOAP stack or "big web services," described in the next section, because of their simplicity, lightweight nature, and integration with HTTP. With the help of URIs and hyperlinks, REST has shown that it is possible to discover web resources without an approach based on registration to a centralized repository.
- Recently, the web Application Description Language (WADL) has been proposed as an XML vocabulary to describe RESTful web services, enabling them to be discovered and accessed immediately by potential clients. However, there are not a variety of toolkits for developing RESTful applications.
- Also, restrictions on GET length, which does not allow encoding of more than 4 KB of data in the resource URI, can create problems because the server would reject such malformed URIs, or may even be subject to crashes. REST is not a standard. It is a design and architectural style for largescale distributed systems.

REST Architectural Elements

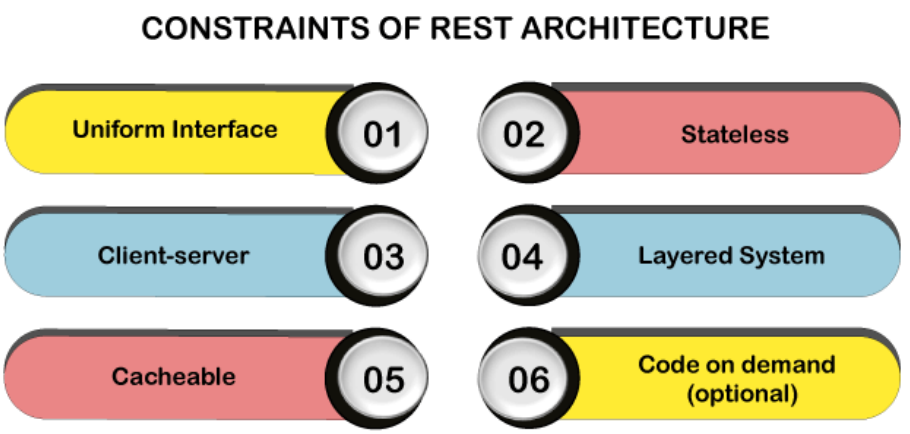
REST Elements	Elements	Example
Data elements	Resource	The intended conceptual target of a hypertext reference
	Resource identifier	URL
	Representation	HTML document, JPEG image, XML, etc.
	Representation metadata	Media type, last-modified time
	Resource metadata	Source link, alternates, vary
	Control data	If-modified-since, cache-control
Connectors	Client	libwww, libwww-perl
	Server	libwww, Apache API, NSAPI
	Cache	Browser cache, Akamai cache network
	Resolver	Bind (DNS lookup library)
	Tunnel	SSL after HTTP CONNECT
Components	Origin server	Apache httpd, Microsoft IIS
	Gateway	Squid, CGI, Reverse Proxy
	Proxy	CERN Proxy, Netscape Proxy, Gauntlet
	User agent	Netscape Navigator, Lynx, MOMspider

- Table lists the REST architectural elements. Several Java frameworks have emerged to help with building RESTful web services. Restlet, a lightweight framework, implements REST architectural elements such as resources, representation, connector, and media type for any kind of RESTful system, including web services.
- In the Restlet framework, both the client and the server are components. Components communicate with each other via connectors.
- JSR-311 (JAX-RS), a specification provided by Sun Microsystems, defines a set of Java APIs for the development of RESTful web services. The specification provides a set of annotations with associated classes and interfaces that can be used to expose Java objects as web resources. JSR- 311 provides clear mappings between the URI and corresponding resources, and mappings between HTTP methods with the methods in Java objects, by using

annotations.

- The API supports a wide range of HTTP entity content types including HTML, XML, JSON, GIF, JPG, and so on Jersey is a reference implementation of the JSR-311 specification for building RESTful web services. It also provides an API for developers to extend Jersey based on their needs.

REpresentational State Transfer (REST) is a software architectural style that defines the constraints to create web services. The web services that follows the **REST** architectural style is called **RESTful Web Services**. It differentiates between the computer system and web services. The REST architectural style describes the **six** barriers



1. Uniform Interface

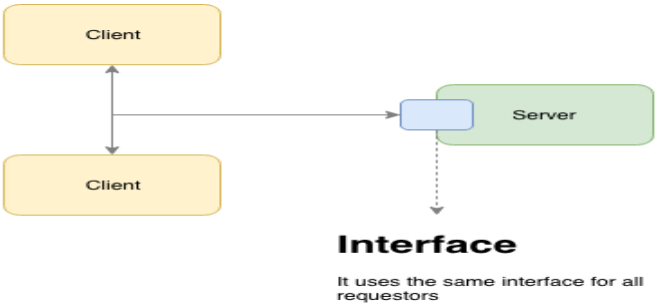
The Uniform Interface defines the interface between **client** and **server**. It simplifies and decomposes the architecture which enables every part to be developed. The Uniform Interface has four guiding principles:

Resource-based: Individual resources are identified using the URI as a resource identifier. The resources themselves are different from the representations returned to the customer. For example, the server cannot send the database but represents some database records expressed to **HTML, XML or JSON** depending on the server request and the implementation details.

Manipulation of resources by representation: When a client represents a resource associated with metadata, there is information on the server to modify or delete it.

Uniform interface is a constraint that describes a contract between clients and

servers.



- **Resource-based**
- The URI is what defines a resource as unique. This representation is what will be returned for clients. If you decided to perform GET to the offer URI, the resource that returns should be a resource representing an order containing the ID order, creation date, and so on. The representation should be in JSON or XML.

Here is a JSON example:

```
{
   id : 1234,
   creation-date : "1937-01-01T12:00:27.87+00:20",
   any-other-json-fields...
}
<order>
   <id>1234</id>
   <creation-date>1937-01-01T12:00:27.87+00:20</creation-date>
   any-other-xml-fields
</order>
```

- **The manipulation of resources using representations**

  Following the happy path, when the client makes a request to the server, the server responds with a resource that represents the current state of its resource. This resource can be manipulated by the client. The client can request what kind it desires for the representation such as JSON, XML, or plain text.

GET https://<HOST>/orders/12345

Accept: text/plain

The next one is in JSON format:

GET https://<HOST>/orders/12345

Accept: application/json

- **Self-descriptive messages**

  In general, the information provided by the RESTful service contains all the information about the resource that the client should be aware of. There is also a possibility of including more information than the resource itself.

Extension	Document Type	MIME type
.aac	AAC audio file	audio/aac
.arc	Archive document	application/octet-stream
.avi	Audio Video Interleave (AVI)	video/x-msvideo
.css	Cascading Style Sheets (CSS)	text/css
.csv	Comma-separated values (CSV)	text/csv
.doc	Microsoft Word	application/msword
.epub	Electronic publication (EPUB)	application/epub+zi

- **Hypermedia as the Engine of Application State (HATEOAS)**
- HATEOAS is a way that the client can interact with the response by navigating within it through the hierarchy in order to get complementary information.

For example, here the client makes a GET call to the order URI :

GET https://<HOST>/orders/1234

- **Self-Descriptive Message:** Each message contains enough information to describe how the message is processed. For example, the parser can be specified by the Internet media type (known as the **MIME** type).
- **As the engine of Hypermedia Application State (HATEOAS):** Customers provide states by query-string parameters, body content, request headers, and

requested URIs. The services provide customers with the state by **response codes, response headers** and **body content**. It is called hypermedia (**hyperlink within hypertext**).

- In addition to the above description, HATEOS also means that, where necessary, the object or itself is contained in the linked body (or header) to supply the URI for retrieving the related objects.
- The same interface that any **REST services** provide is fundamental to the design

2. Client-server

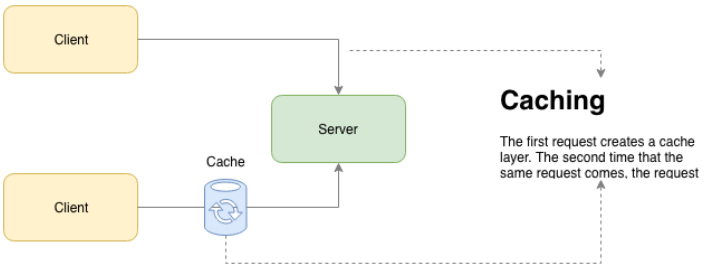
- A client-server interface separates the client from the server. For Example, the separation of concerns not having an internal relationship with internal storage for each server to improve the portability of customer's data codes. Servers are not connected with the user interface or user status to make the server simpler and scalable. Servers and clients are independently replaced and developed until the interface is changed.

3. Stateless

Stateless means the state of the service doesn't persist between subsequent requests and response. It means that the request itself contains the state required to handle the request. It can be a query-string parameter, entity, or header as a part of the **URI**. The URI identifies the resource and state (or state change) of that resource in the unit. After the server performs the appropriate state or status piece (s) that matters are sent back to the client through the header, status, and response body.

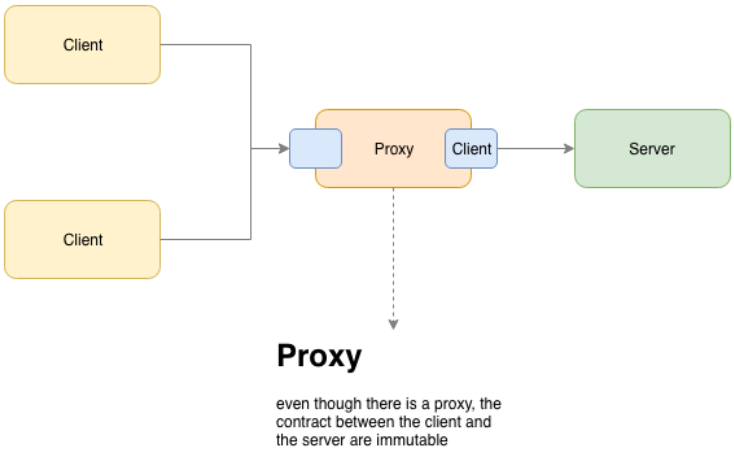
Cacheable

The aim of caching is to never have to generate the same response more than once. The key benefits of using this strategy are an increase in speed and a reduction in server processing.



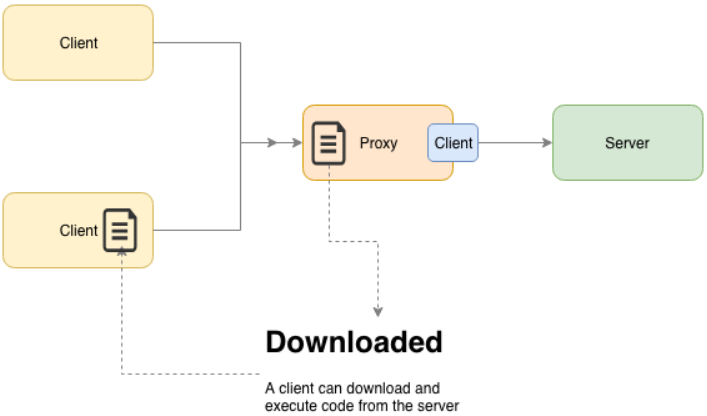
A layered system

Each layer must work independently and interact only with the layers directly connected to it. This strategy allows passing the request without bypassing other layers. For instance, when scaling a service is desired, you might use a proxy working as a load balancer—that way, the incoming requests are deliverable to the appropriate server instance.



Code on demand

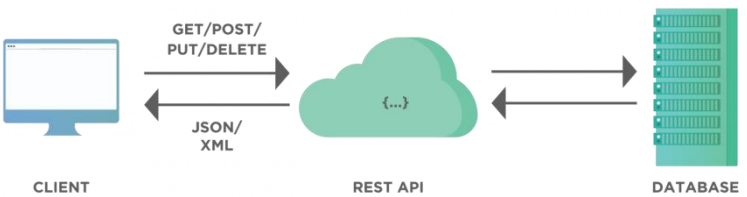
In summary, this optional pattern allows the client to download and execute code from the server on the client side. The constraint says that this strategy improves scalability since the code can execute independently of the server on the client side:



Creating RESTFul Webservice

In next chapters, we'll create a webservice say user management with following functionalities –

Sr.No.	URI	HTTP Method	POST body	Result
1	/UserService/users	GET	empty	Show list of all the users.
2	/UserService/addUser	POST	JSON String	Add details of new user.
3	/UserService/getUser/:id	GET	empty	Show details of a user.



Example RESTful Web Service in Amazon S3 Interface

- A good example of RESTful web service application in high-performance computing systems is the Amazon Simple Storage Service (S3) interface. Amazon S3 is data storage for Internet applications. It provides simple web services to store and retrieve data from anywhere at any time via the web. S3 keeps fundamental entities, "objects," which are named pieces of data accompanied by some metadata to be stored in containers called "buckets," each identified by a unique key.
- Buckets serve several purposes: They organize the Amazon S3 namespace at the highest level, identify the account responsible for storage and data

transfer charges, play a role in access control, and serve as the unit of aggregation for usage reporting.

- Amazon S3 provides three types of resources: a list of user buckets, a particular bucket, and a particular S3 object, accessible through https:// s3.amazonaws.com/{name-of-bucket}/{name-of-object}.

Table: Sample REST Request-Response for Creating an S3 Bucket

REST Request	REST Response
PUT/[bucket-name] HTTP/1.0	HTTP/1.1 200 OK
Date: Wed, 15 Mar 2011 14:45:15 GMT	x-amz-id-2: VjzdTviQorQtSjcgLshzCZSzN+7CnewvHA
Authorization:AWS [aws-access-key-id]:	+6sNxR3VRcUPyO5fmSmo8bWnIS52qa
[header-signature]	x-amz-request-id: 91A8CC60F9FC49E7
Host: s3.amazonaws.com	Date: Wed, 15 Mar 2010 14:45:20 GMT
	Location: /[bucket-name]
	Content-Length: 0
	Connection: keep-alive

- These resources are retrieved, created, or manipulated by basic HTTP standard operations: GET, HEAD, PUT, and DELETE. GET can be used to list buckets created by the user, objects kept inside a bucket, or an object's value and its related metadata. PUT can be used for creating a bucket or setting an object's value or metadata, DELETE for removing a particular bucket or object, and HEAD for getting a specific object's metadata.
- The Amazon S3 API supports the ability to find buckets, objects, and their related metadata; create new buckets; upload objects; and delete existing buckets and objects for the aforementioned operations. Table 2.2 shows some sample REST request-response message syntax for creating an S3 bucket.
- Amazon S3 REST operations are HTTP requests tocreate, fetch, and delete buckets and objects.
- A typical REST operation consists of sending a single HTTP request to Amazon S3, followed by waiting for an HTTP response.
- Like any HTTP request, a request to Amazon S3 contains a request method, a URI, request headers which contain basic information about the request,

and sometimes a query string and request body. The response contains a status code, response headers, and sometimes a response body.

- The request consists of a PUT command followed by the bucket name created on S3. The Amazon S3 REST API uses the standard HTTP header to pass authentication information.
- The authorization header consists of an AWS Access Key ID and AWS SecretAccess Key, issued by the developers when they register to S3 Web Services, followed by a signature.
- To authenticate, the AWSAccessKeyId element identifies the secret key to compute the signature upon request from the developer. If the request signature matches the signature included, the requester is authorized and subsequently, the request is processed.
- The composition of RESTful web services has been the main focus of composite Web 2.0 applications, such as mashups. A mashup application combines capabilities from existing web-based applications. A good example of a mashup is taking images from an online repository such as Flickr and overlaying them on Google Maps.
- Mashups differ from all-in-one software products in that instead of developing a new feature into an existing tool, they combine the existing tool with another tool that already has the desired feature. All tools work independently, but create a uniquely customized experience when used together in harmony.

2.1.2 Services and Web Services

- In an SOA paradigm, software capabilities are delivered and consumed via loosely coupled, reusable, coarse-grained, discoverable, and self-contained services interacting via a message-based communication model.
- The web has becomes a medium for connecting remote clients with applications for years, and more recently, integrating applications across the Internet has gained in popularity.
- The term “web service” is often referred to a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web.
- Once a web service is deployed, other applications and other web services can discover and invoke the deployed service (Figure).

- In fact, a web service is one of the most common instances of an SOA implementation. The W3C working group defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network.
- According to this definition, a web service has an interface described in a machine-executable format (specifically Web Services Description Language or WSDL). Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards. The technologies that make up the core of today's web services are as follows:

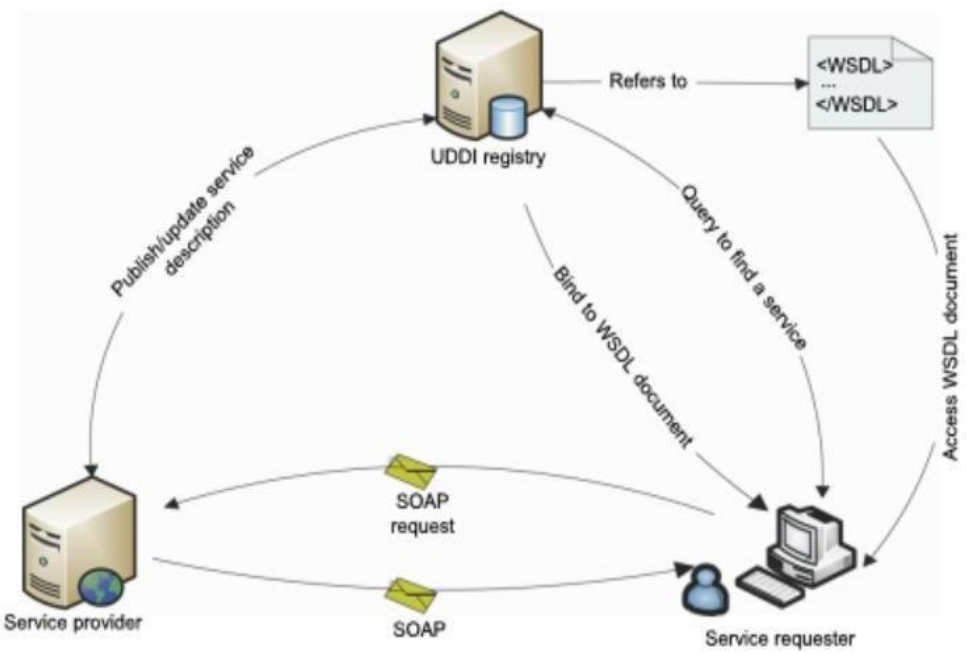


FIGURE: A simple web service interaction among provider, user, and the UDDI registry.

- Simple Object Access Protocol (SOAP) SOAP provides a standard packaging structure for transmission of XML documents over various Internet protocols, such as SMTP, HTTP, and FTP. By having such a standard message format, heterogeneous middleware systems can achieve interoperability.
- A SOAP message consists of a root element called envelope, which contains

a header: a container that can be extended by intermediaries with additional application-level elements such as routing information, authentication, transaction management, message parsing instructions, and Quality of Service (QoS) configurations, as well as a body element that carries the payload of the message.

- The content of the payload will be marshaled by the sender's SOAP engine and unmarshaled at the receiver side, based on the XML schema that describes the structure of the SOAP message.
- Web Services Description Language (WSDL) WSDL describes the interface, a set of operations supported by a web service in a standard format.
- It standardizes the representation of input and output parameters of its operations as well as the service's protocol binding, the way in which the messages will be transferred on the wire. Using WSDL enables disparate clients to automatically understand how to interact with a web service.
- Universal Description, Discovery, and Integration (UDDI) UDDI provides a global registry for advertising and discovery of web services, by searching for names, identifiers, categories, or the specification implemented by the web service.
- SOAP is an extension, and an evolved version of XML-RPC, a simple and effective remote procedure call protocol which uses XML for encoding its calls and HTTP as a transport mechanism, introduced in 1999.
- According to its conventions, a procedure executed on the server and the value it returns was a formatted in XML. However, XML-RPC was not fully aligned with the latest XML standardization.
- Moreover, it did not allow developers to extend the request or response format of an XML-RPC call. As the XML schema became a W3C recommendation in 2001, SOAP mainly describes the protocols between interacting parties and leaves the data format of exchanging messages to XML schema to handle.
- The major difference between web service technology and other technologies such as J2EE, CORBA, and CGI scripting is its standardization, since it is based on standardized XML, providing a language-neutral representation of data.
- Most web services transmit messages over HTTP, making them available as

Internet-scale applications. In addition, unlike CORBA and J2EE, using HTTP as the tunneling protocol by web services enables remote communication through firewalls and proxies.

- SOAP-based web services are also referred to as "big web services". RESTful services can also be considered a web service, in an HTTP context.
- SOAP-based web services interaction can be either synchronous or asynchronous, making them suitable for both request-response and one-way exchange patterns, thus increasing web service availability in case of failure.

2.1.2.1 WS-I Protocol Stack

- Unlike RESTful web services that do not cover QoS and contractual properties, several optional specifications have been proposed for SOAP-based web services to define non functional requirements and to guarantee a certain level of quality in message communication as well as reliable transactional policies, such as WS-Security, WS- Agreement, WS-Reliable Messaging, WS-Transaction, and WS-Coordination as shown in Figure.

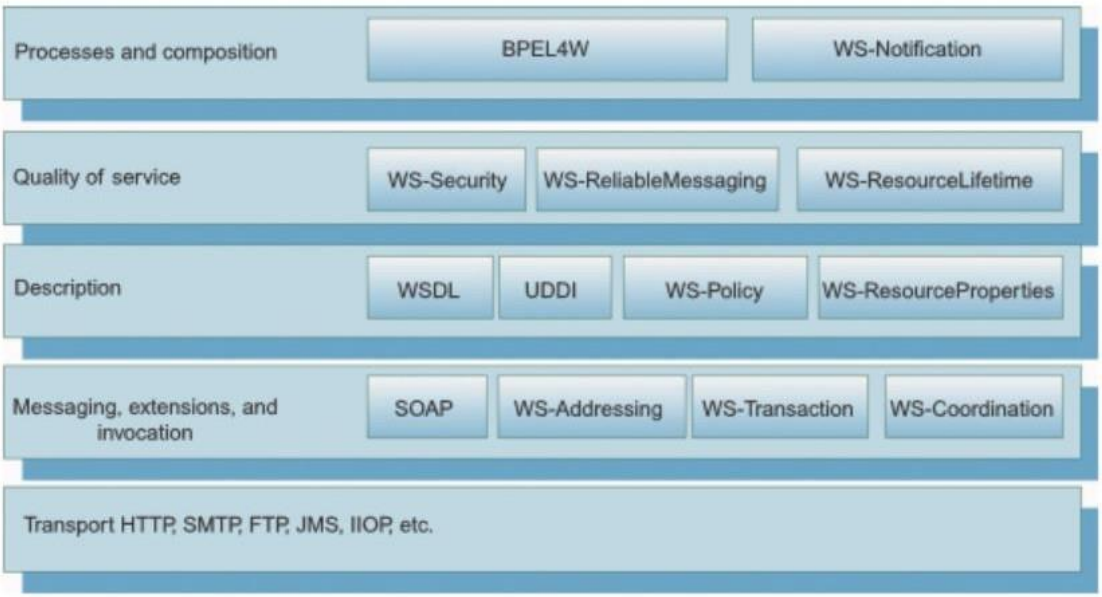


FIGURE: WS-I protocol stack and its related specifications

- SOAP messages are encoded using XML, which requires that all self-described data be sent as ASCII strings.

- The description takes the form of start and end tags which often constitute half or more of the message's bytes. Transmitting data using XML leads to a considerable transmission overhead, increasing the amount of transferred data by a factor 4 to 10.
- Moreover, XML processing is compute and memory intensive and grows with both the total size of the data and the number of data fields, making web services inappropriate for use by limited-profile devices, such as handheld PDAs and mobile phones.
- Web services provide on-the-fly software composition, through the use of loosely coupled, reusable software components.
- By using Business Process Execution Language for Web Services (BPEL4WS), a standard executable language for specifying interactions between web services recommended by OASIS, web services can be composed together to make more complex web services and workflows. BPEL4WS is an XML-based language, built on top of web service specifications, which is used to define and manage long-lived service orchestrations or processes.
- In BPEL, a business process is a large-grained stateful service, which executes steps to complete a business goal.
- That goal can be the completion of a business transaction, or fulfillment of the job of a service.
- The steps in the BPEL process execute activities (represented by BPEL language elements) to accomplish work.
- Those activities are centered on invoking partner services to perform tasks (their job) and return results back to the process. BPEL enables organizations to automate their business processes by orchestrating services.
- Workflow in a grid context is defined as "The automation of the processes, which involves the orchestration of a set of Grid services, agents and actors that must be combined together to solve a problem or to define a new service."
- The JBPM Project, built for the JBoss open source middleware platform, is an example of a workflow management and business process execution system.
- Another workflow system, Taverna, has been extensively used in life science

applications. There are a variety of tools for developing and deploying web services in different languages, among them SOAP engines such as Apache Axis for Java, gSOAP for C++, the Zolera Soap Infrastructure (ZSI) for Python, and Axis2/Java and Axis2/C.

- These toolkits, consisting of a SOAP engine and WSDL tools for generating client stubs, considerably hide the complexity of web service application development and integration. As there is no standard SOAP mapping for any of the aforementioned languages, two different implementations of SOAP may produce different encodings for the same objects.
- Since SOAP can combine the strengths of XML and HTTP, as a standard transmission protocol for data, it is an attractive technology for heterogeneous distributed computing environments, such as grids and clouds, to ensure interoperability.
- Open Grid Services Architecture (OGSA) grid services are extensions of web services and in new grid middleware, such as Globus Toolkit 4 and its latest released version GT5, pure standard web services. Amazon S3 as a cloud-based persistent storage service is accessible through both a SOAP and a REST interface.
- However, REST is the preferred mechanism for communicating with S3 due to the difficulties of processing large binary objects in the SOAP API, and in particular, the limitation that SOAP puts on the object size to be managed and processed. Table 2.3 depicts a sample SOAP request response to get a user object from S3.
- A SOAP message consists of an envelope used by the applications to enclose information that need to be sent. An envelope contains a header and a body block. The EncodingStyle element refers to the URI address of an XML schema for encoding elements of the message.
- Each element of a SOAP message may have a different encoding, but unless specified, the encoding of the whole message is as defined in the XML schema of the root element. The header is an optional part of a SOAP message that may contain auxiliary information as mentioned earlier, which does not exist in this example.

Table : Sample SOAP Request-Response for Creating an S3 Bucket

SOAP Request	SOAP Response
<soap:Envelope	<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/ soapenvelope"	xmlns:soap= "http://www.w3.org/2003/05/ soapenvelope"
soap:encodingStyle= "http://www.w3.org/2001/12/soa p- encoding">	soap:encodingStyle= "http://www.w3.org/2001/12/soa p- encoding">
<soap:Body>	<soap:Body>
<CreateBucket xmlns="http:// doc.s3.amazonaws.com/2010-03- 15">	<CreateBucket xmlns="http:// doc.s3.amazonaws.com/2010-03- 15">
<Bucket>SampleBucket</Bucket>	<Bucket>SampleBucket</Bucket>
<AWSAccessKeyId> 1B9FVRAYCP1VJEXAMPLE= </AWSAccessKeyId>	<AWSAccessKeyId> 1B9FVRAYCP1VJEXAMPLE= </AWSAccessKeyId>
<Timestamp>2010-03-15T14:40:00.165Z	<Timestamp>2010-03-15T14:40:00.165Z
</Timestamp>	</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqa EXAMPLE=</Signature>	<Signature>Iuyz3d3P0aTou39dzbqa EXAMPLE=</Signature>
</CreateBucket>	</CreateBucket>
</soap:Body>	</soap:Body>
</soap:Envelope>	</soap:Envelope>

Table: The 10 Areas Covered by the Core WS-* Specifications

WS-* Specification Area	Examples
1. Core Service Model	XML, WSDL, SOAP
2. Service Internet	WS-Addressing, WS-MessageDelivery, Reliable WSRM, Efficient MOTM
3. Notification	WS-Notification, WS-Eventing (Publish-Subscribe)

4. Workflow and Transactions	BPEL, WS-Choreography, WS-Coordination
5. Security	WS-Security, WS-Trust, WS-Federation, SAML, WS-SecureConversation
6. Service Discovery	UDDI, WS-Discovery
7. System Metadata and State	WSRF, WS-MetadataExchange, WS-Context
8. Management	WSDM, WS-Management, WS-Transfer
9. Policy and Agreements	WS-Policy, WS-Agreement
10. Portals and User Interfaces	WSRP (Remote Portlets)

- The body of a SOAP request-response message contains the main information of the conversation, formatted in one or more XML blocks.
- In this example, the client is calling CreateBucket of the Amazon S3 web service interface. In case of an error in service invocation, a SOAP message including a Fault element in the body will be forwarded to the service client as a response, as an indicator of a protocol-level error.

2.1.2.2 WS-* Core SOAP Header Standards

- Table 2.4 summarizes some of the many (around 100) core SOAP header specifications. There are many categories and several overlapping standards in each category.
- The number and complexity of the WS-* standards have contributed to the growing trend of using REST and not web services.
- It was a brilliant idea to achieve interoperability through self-describing messages, but experience showed that it was too hard to build the required tooling with the required performance and short implementation time.

Example WS-RM or WS-Reliable Messaging

- WS-RM is one of the best developed of the so-called WS-* core web service specifications. WS-RM uses message instance counts to allow destination services to recognize message delivery faults (either missing or out-of-order messages).
- WS-RM somewhat duplicates the capabilities of TCP-IP in this regard, but

operates at a different level—namely at the level of user messages and not TCP packets, and from source to destination independent of TCP routing in between.

2.2 PUBLISH-SUBSCRIBE MODEL AND NOTIFICATION

- An important concept here is "publish-subscribe" which describes a particular model for linking source and destination for a message bus. Here the producer of the message (publisher) labels the message in some fashion; often this is done by associating one or more topic names from a (controlled) vocabulary. Then the receivers of the message (subscriber) will specify the topics for which they wish to receive associated messages. Alternatively, one can use content-based delivery systems where the content is queried in some format such as SQL.

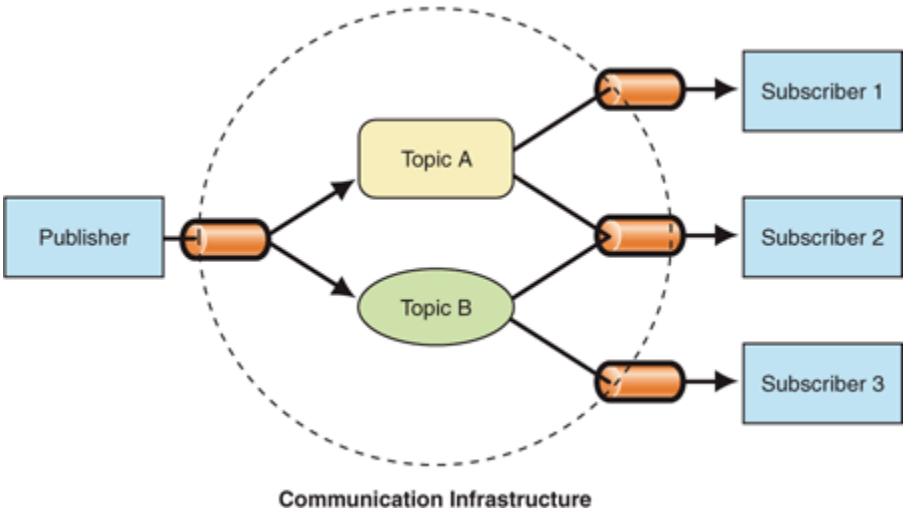
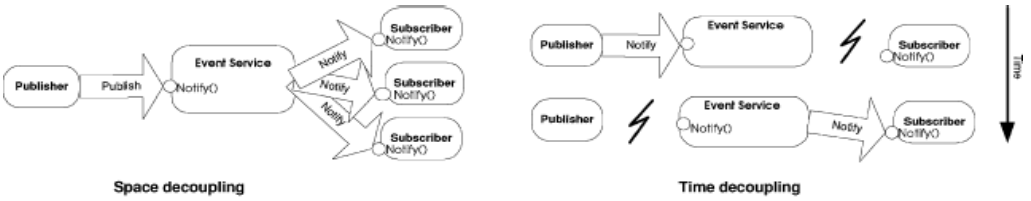


Fig. 1. A simple object-based publish/subscribe system.

- The producer, or publisher, writes messages to an input channel and sends them to a topic, unaware which application will receive them.
- The consumer, or subscriber, subscribes to the topic and receives messages from an output channel, unaware which service produced these messages.
- Publish/Subscribe (Pub/Sub) messaging provides instant event notifications for these distributed applications. The Publish Subscribe model enables event-driven architectures and asynchronous parallel processing, while improving performance, reliability and scalability.
- ***Topic:*** A named resource (which acts as an alias) to which the publishers send messages.
- ***Publisher:*** It is an application that creates and sends a message to a Pub/Sub topic. It is additionally referred to as Producer.
- ***Subscriber:*** It is an application that registers itself to a specified topic of interest in order to get appropriate messages.
- ***Subscription:*** It represents a stream of messages from a single, specific topic, which is to be delivered to the subscribing application.
- ***Message:*** It is a combination of data and other attributes which is sent by a publisher to a topic and is delivered to the subscribers at the end.
- ***Message attribute:*** A key-value pair used by a publisher to define a message.



- **Centralized Broker model**

Consists of multiple publishers and multiple subscribers and centralized broker/brokers (an overlay network of brokers interacting with each other).

Subscribers/Publishers will contact 1 broker, and does not need to have knowledge about others.

E.g. CORBA event services, JMS, JEDI etc…

Peer-to-Peer model

- Each node can be publisher, subscriber or broker.
- Subscribers subscribe to publishers directly and publishers notify subscribers directly. Therefore they must maintain knowledge of each other.
- Complex in nature, mechanisms such as DHT and

CHORD are employed to locate nodes in the network.

- E.g. Java distributed event service

Event filtering (event selection)

- The process which selects the set of subscribers that have shown interest in a given event. Subscriptions are stored in memory and searched when a publisher publishes a new event.

Event routing (event delivery)

- The process of routing the published events from the publisher to all interested subscribers
- After filtering the events, the broker/brokers must route the events to the corresponding subscribers.
- Can be done in the following ways:
- Unicast
- Multicast
- Server push/ client pull

Topic based

- Generally also known as topic based, group based or channel based event filtering.
- Each event is published to one of these channels by its publisher.
- Subscribers subscribes to a particular channel and will receive ALL events published to the subscribed channel.

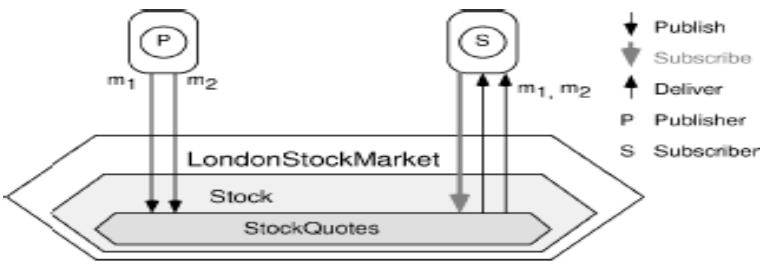


Fig. 12. Topic-based publish/subscribe interactions.

- Simple processfor matching an event to subscriptions. However, limited expressiveness.
- Event filtering is easy, event routing is difficult (Heavy load on the network). The challenge is to multicast event effectively to subscribers.

Content based

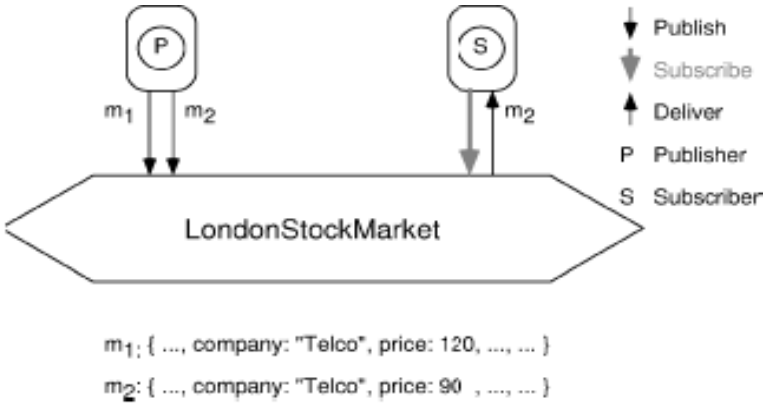
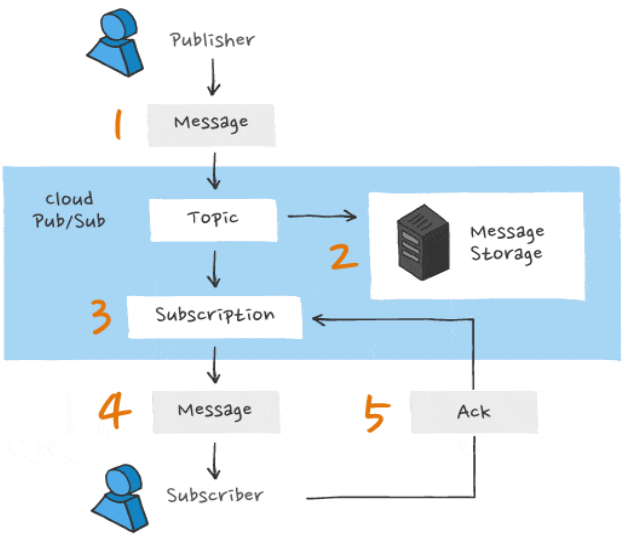


Fig. 14. Content-based publish/subscribe interactions.

- Added complexity in matching an event to subscriptions.
  (Implementation: Subscription arranged in a matching tree, where each node is a partial condition.
- However, more precision is provided and event routing is easier



1. A publisher application creates a topic in the Publisher/Subscriber service and sends messages to the topic. The message comprises a payload and optional attributes describing the payload content.
2. The service makes sure that published messages are retained on behalf of the subscriptions. The message published is retained for a subscription till it is acknowledged by a subscriber consuming messages from that particular

subscription.

3. Pub/Sub forwards messages individually from a topic to all of its subscriptions.
4. A subscriber receives messages either by Pub/Sub pushing them to the subscriber's chosen endpoint or by the subscriber pulling them from the service.
5. An acknowledgment is sent by the subscriber to the Pub/Sub service for each received message.
6. The service then removes acknowledged messages from the subscription's message queue.

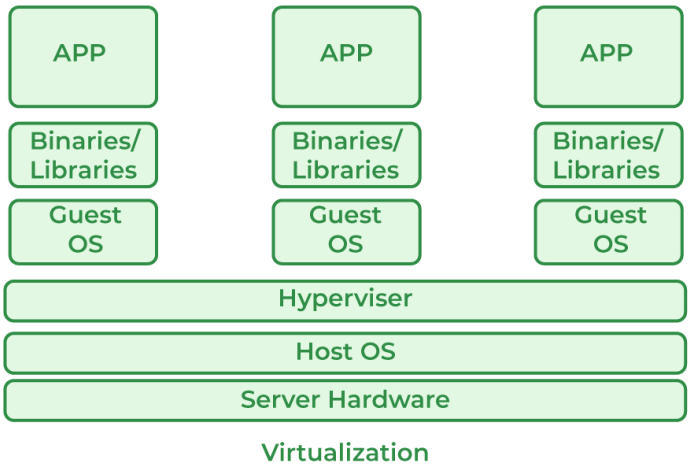
Message Filtering

- In the publish-subscribe model, subscribers typically receive only a subset of the total messages published. The process of selecting messages for reception and processing is called filtering. There are two common forms of filtering: topic-based and content-based.
- In a topic-based system, messages are published to "topics" or named logical channels. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe. The publisher is responsible for defining the topics to which subscribers can subscribe.
- In a content-based system, messages are only delivered to a subscriber if the attributes or content of those messages matches constraints defined by the subscriber. The subscriber is responsible for classifying the messages.
- The use of topic or content-based message selection is termed message filtering. Note that in each of these cases, we find a many-to-many relationship between publishers and subscribers. Publish-subscribe messaging middleware allows straightforward implementation of notification or event-based programming models. The messages could, for example, be labeled by the desired notifying topic (e.g., an error or completion code) and contain content elaborating thenotification.

2.3 BASICS OF VIRTUALIZATION

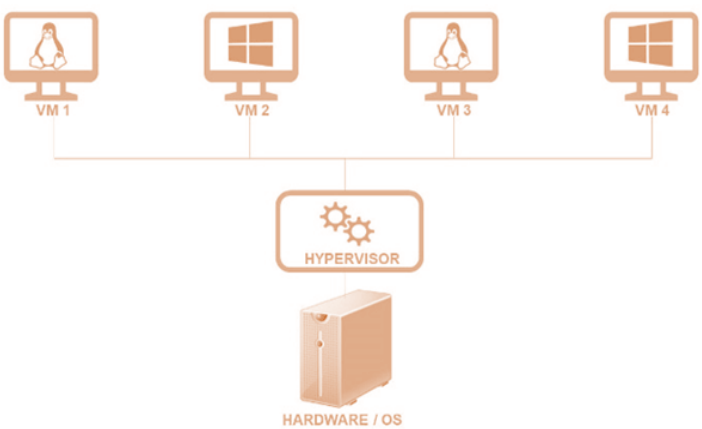
- *Virtualization* is a method of running multiple independent virtual operating systems on a single physical computer. A virtual machine (sometimes called a pseudo-machine). The creation and management of virtual machines has often been called *platform virtualization.*
- Platform virtualization is performed on a given computer (hardware platform) by software called a control program. The control program creates a simulated environment, a virtual computer, which enables the device to use hosted software specific to the virtual environment, sometimes called guest software.
- The guest software, which is often itself a complete operating system, runs just as if it were installed on a stand-alone computer. Frequently, more than one virtual machine is able to be simulated on a single physical computer, their number being limited only by the host device's physical hardware resources.
- Because the guest software often requires access to specific peripheral devices in order to function, the virtualized platform must support guest interfaces to those devices.
- Examples of such devices are the hard disk drive, CD-ROM, DVD, and network interface card.
- Virtualization technology is a way of reducing the majority of hardware acquisition and maintenance costs, which can result in significant savings forany company.

Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines. Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine



Virtualization

- Hypervisor is a program that allows multiple Operating Systems to share a single physical hardware. Each operating system will share the host's processor, memory, file storage, and other resources.
- The hypervisor controls the host processor and resources, allocating what is needed to each operating system. This ensures that he the guest operating systems (called virtual machines) cannot interrupt each other.
- Since Hypervisors help create and manage virtual machines (VMs), they are also known as Virtual Machine Monitors or VMMs.



hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.

A hypervisor is software that creates and runs virtual machines (VMs). A hypervisor, sometimes called a virtual machine monitor (VMM), isolates the

hypervisor operating system and resources from the virtual machines and enables the creation and management of those VMs

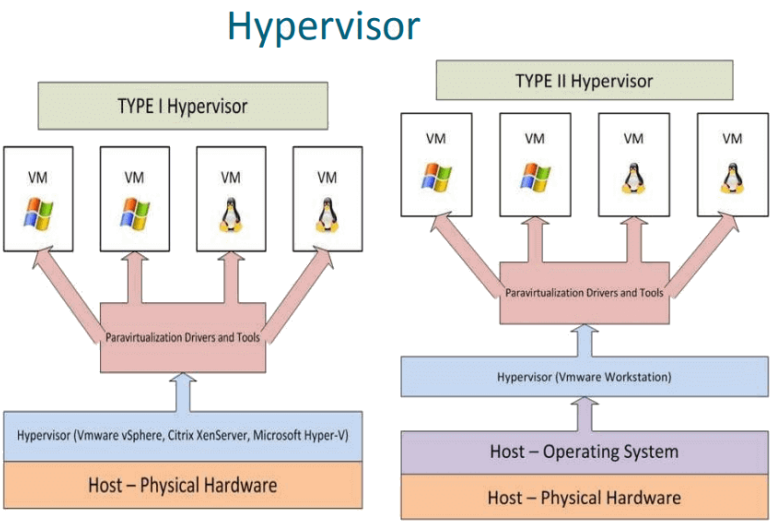
Types of Hypervisor –

TYPE-1Hypervisor:

The hypervisor runs directly on the underlying host system. It is also known as a "Native Hypervisor" or "Bare metal hypervisor". It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer, and Microsoft Hyper-V hypervisor.

TYPE-2 Hypervisor:

A Host operating system runs on the underlying host system. It is also known as 'Hosted Hypervisor". Such kind of hypervisors doesn't run directly over the underlying hardware rather they run as an application in a Host system(physical machine).



2.3.1 Parallel Processing

- Parallel processing is performed by the simultaneous execution of program instructions that have been allocated across multiple processors with the objective of running a program in less time.
- The next advancement in parallel processing was multiprogramming. In a multiprogramming system, multiple programs submitted by users are each allowed to use the processor for a short time, each taking turns and having exclusive time with the processor in order to execute instructions. This approach is known as "round-robin scheduling" (RR scheduling).
- In RR scheduling, a small unit of time called a time slice (or quantum) is defined. All executable processes are held in a circular queue. The time slice is defined based on the number of executable processes that are in the queue. For example, if there are five user processes held in the queue and the time slice allocated for the queue to execute in total is 1 second, each user process is allocated 200 milliseconds of process execution time on the CPU before the scheduler begins moving to the next process in the queue.
- The CPU scheduler manages this queue, allocating the CPU to each process for a time interval of one time slice. New processes are always added to the end of the queue. The CPU scheduler picks the first process from the queue, sets its timer to interrupt the process after the expiration of the timer, and then dispatches the next process in the queue. The process whose time has expired is placed at the end of the queue.
- If a process is still running at the end of a time slice, the CPU is interrupted and the process goes to the end of the queue. If the process finishes before the end of the time-slice, it releases the CPU voluntarily.
- In either case, the CPU scheduler assigns the CPU to the next process in the queue. Every time a process is granted the CPU, a context switch occurs, which adds overhead to the process execution time. To users it appears that all of the programs are executing at the same time.
- Resource contention problems often arose in these early systems. Explicit requests for resources led to a condition known as deadlock. Competition for resources on machines with no tie-breaking instructions led to the critical section routine. Contention occurs when several processes request access to the same resource.

- In order to detect deadlock situations, a counter for each processor keeps track of the number of consecutive requests from a process that have been rejected. Once that number reaches a predetermined threshold, a state machine that inhibits other processes from making requests to the main store is initiated until the deadlocked process is successful in gaining access to the resource.

2.3.2 Vector Processing

- The next step in the evolution of parallel processing was the introduction of multiprocessing. Here, two or more processors share a common workload.
- The earliest versions of multiprocessing were designed as a master/slave model, where one processor (the master) was responsible for all of the tasks to be performed and it only off-loaded tasks to the other processor (the slave) when the master processor determined, based on a predetermined threshold, that work could be shifted to increase performance.
- Vector processing was developed to increase processing performance by operating in a multitasking manner. Matrix operations were added to computers to allow a single instruction to manipulate two arrays of numbers performing arithmetic operations.
- This was valuable in certain types of applications in which data occurred in the form of vectors or matrices. In applications with less well-formed data, vector processing was less valuable.

2.3.3 Symmetric Multiprocessing Systems

- The next advancement was the development of symmetric multiprocessing systems (SMP) to address the problem of resource management in master/ slave models.
- In SMP systems, each processor is equally capable and responsible for managing the workflow as it passes through the system.
- The primary goal is to achieve *sequential consistency,* in other words, to make SMP systems appear to be exactly the same as a single-processor, multiprogramming platform. Engineers discovered that system performance could be increased nearly 10–20% by executing some instructions out of order.
- However, programmers had to deal with the increased complexity and cope

with a situation where two or more programs might read and write the same operands simultaneously.

- This difficulty, however, is limited to a very few programmers, because it only occurs in rare circumstances. To this day, the question of how SMP machines should behave when accessing shared data remains unresolved.
- Data propagation time increases in proportion to the number of processors added to SMP systems.
- After a certain number (usually somewhere around 40 to 50 processors), performance benefits gained by using even more processors do not justify the additional expense of adding such processors.
- To solve the problem of long data propagation times, message passing systems were created. In these systems, programs that share data send messages to each other to announce that particular operands have been assigned a new value. Instead of a global message announcing an operand's new value, the message is communicated only to those areas that need to know the change.
- There is a network designed to support the transfer of messages between applications. This allows a great number processors (as many as several thousand) to work in tandem in a system. These systems are highly scalable and are called massively parallel processing (MPP) systems.

2.3.4 Massively Parallel Processing Systems

- *Massive parallel processing* is used in computer architecture circles to refer to a computer system with many independent arithmetic units or entire microprocessors, which run in parallel. "Massive" connotes hundreds if not thousands of such units. In this form of computing, all the processing elements are interconnected to act as one very large computer.
- This approach is in contrast toa distributed computing model, where massive numbers of separate computers are used to solve a single problem (such as in the SETI project, mentioned previously).
- Early examples of MPP systems were the Distributed Array Processor, the Goodyear MPP, the Connection Machine, and the Ultracomputer. In data mining, there is a need to perform multiple searches of a static database.
- The earliest massively parallel processing systems all used serial computers

as individual processing units in order to maximize the number of units available for a given size and cost.

- Single-chip implementations of massively parallel processor arrays are becoming ever more cost effective due to the advancements in integrated circuit technology.
- An example of the use of MPP can be found in the field of artificial intelligence.
- For example, a chess application must analyze the outcomes of many possible alternatives and formulate the best course of action to take.
- Another example can be found in scientific environments, where certain simulations (such as molecular modeling) and complex mathematical problems can be split apart and each part processed simultaneously. Parallel data query (PDQ) is a technique used in business.
- This technique divides very large data stores into pieces based on various algorithms. Rather than searching sequentially through an entire database to resolve a query, 26 CPUs might be used simultaneously to perform a sequential search, each CPU individually evaluating a letter of the alphabet.
- MPP machines are not easy to program, but for certain applications, such as data mining, they are the best solution.

2.4 TYPES OF VIRTUALIZATION

Virtualization is a method of running multiple independent virtual operating systems on a single physical computer

Types of virtualization

1. Hardware Virtualization
    - Full Virtualization
    - Emulation Virtualization
    - Para-virtualization
2. Software Virtualization
3. OS Virtualization

4. Server Virtualization
5. Storage Virtualization

1. Hardware Virtualization

This is one of the most common types of Virtualization as it is related to the application uptime and utilization of hardware. The primary motive behind this technology brings together all the physical servers into one large physical server. Thus making the processor work more efficiently and effectively.

The operating system of the physical server gets transformed into a well to do OS that runs on a virtual system. It consists of a hypervisor that solely manages the memory, processor and other elements by allowing multiple OS to run on the same machine without the help of any source code.

Hardware virtualization is of three kinds.

These are:

- Full Virtualization: Here the hardware architecture is completely simulated. Guest software doesn't need any modification to run any applications.
- Emulation Virtualization: Here the virtual machine simulates the hardware & is independent. Furthermore, the guest OS doesn't require any modification.
- Para-Virtualization: Here, the hardware is not simulated; instead the guest software runs its isolated system.

2. Software Virtualization

It is also called application virtualization is the practice of running software from a remote server. Software virtualization is similar to that of virtualization except that it is capable to abstract the software installation procedure and create virtual software installation.

3. OS level virtualization

It is also called OS-level virtualization is a type of virtualization technology which work on OS layer. Here the kernel of an OS allows more than one isolated user-space instances to exist. Such instances are called containers/software containers or virtualization engines.

4. Server virtualization

It is the division of physical server into several virtual servers and this division is

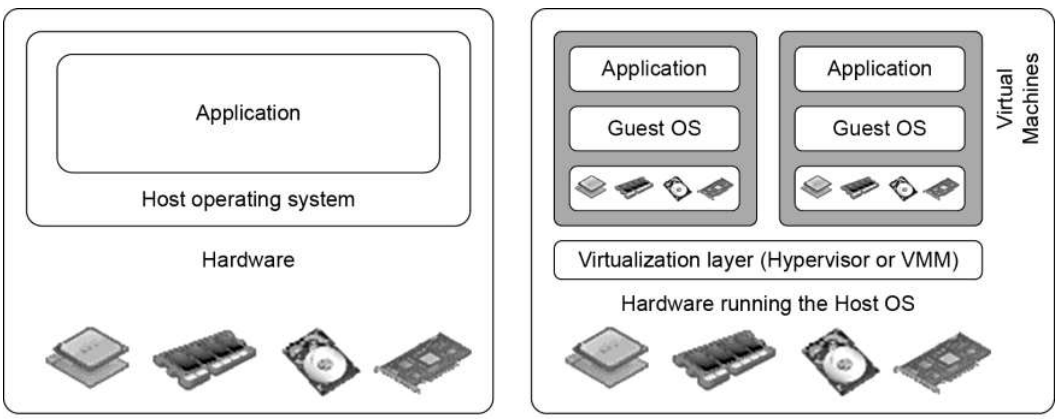
mainly done to improvise the utility of server resource. In other word it is the masking of resources that are located in server which includes the number & identity of processors, physical servers & the operating system.

5. **Storage Virtualization**

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 also has many advantages with this as it has the capacity to reduce downtime, speed, performance and load balancing.

2.5 IMPLEMENTATION LEVELS OF VIRTUALIZATION

- A traditional computer runs with a host operating system specially tailored for its hardware architecture.
- After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional software, called a virtualization layer. This virtualization layer is known as hypervisor or virtual machine monitor (VMM).
- The VMs are shown in the upper boxes, where applications run with their own guest OS over the virtualized CPU, memory, and I/O resources.



(a) Traditional computer (b) After virtualization

The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively. This can be implemented at various operational levels, as we will discuss shortly.

The Virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system.

Common virtualization layers include

- Instruction set architecture (ISA) level,
- hardware level,
- operating system level,
- library support level, and
- application level

Five abstraction levels

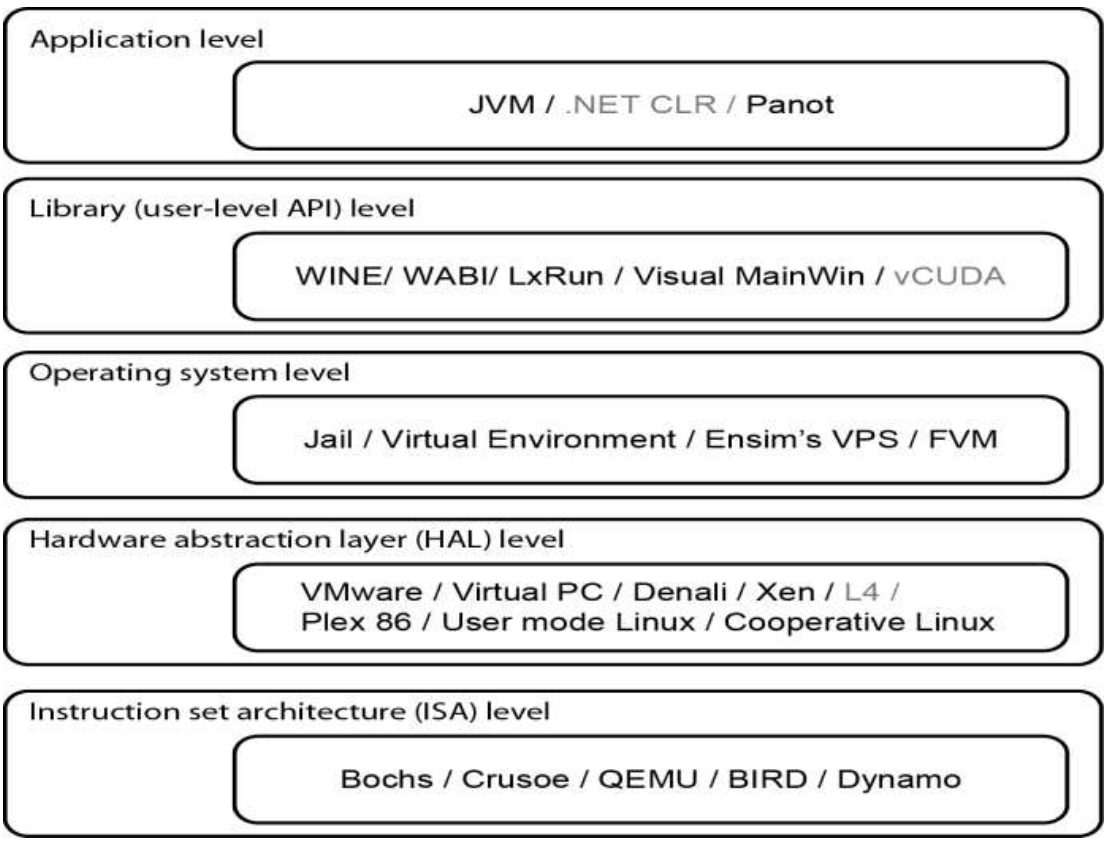


Figure: Virtualization ranging from hardware to applications in five abstraction levels

1. Instruction Set Architecture Level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.
- The basic emulation method is through code interpretation. An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function.
- This approach translates basic blocks of dynamic source instructions to target instructions. The basic blocks can also be extended to program traces or super blocks to increase translation efficiency. Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

2. Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware. On the one hand, this approach generates a virtual hardware environment for a VM. On the other hand, the process manages the underlying hardware through virtualization.
- The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices. The intention is to upgrade the hardware utilization rate by multiple users concurrently. The idea was implemented in the IBM VM/370 in the 1960s. More recently, the Xen hypervisor has been applied to virtualize x86-based machines to run Linux or other guest OS.

3. Operating System Level

This refers to an abstraction layer between traditional OS and user applications. OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users.

Advantages of OS Extension for Virtualization

1. VMs at OS level has minimum startup/shutdown costs

2. OS-level VM can easily synchronize with its environment

Disadvantage of OS Extension for Virtualization

- All VMs in the same OS container must have the same or similar guest OS, which restrict application flexibility of different VMs on the same physical machine.

4. Library Support Level

- Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS. Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization.
- Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks. The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.
- Another example is the vCUDA which allows applications executing within VMs to leverage GPU hardware acceleration.

5. User-Application Level

Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization.

- The most popular approach is to deploy high level language (HLL) VMs. In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.
- Any program written in the HLL and compiled for this VM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM.

2.5.1.1 Relative Merits of Different Approaches

Relative Merits of Virtualization at Various Levels (More "X"'s Means Higher Merit, with a Maximum of 5 X's)

- The column headings correspond to four technical merits.

- "Higher Performance" and "Application Flexibility" are self-explanatory. "Implementation Complexity" implies the cost to implement that particular virtualization level. "Application Isolation" refers to the effort required to isolate resources committed to different VMs.

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application Level	XX	XX	XXXXX	XXXXX

VMM Design Requirements

- There are three requirements for a VMM.
  - □ First, a VMM should provide an environment for programs which is essentially identical to the original machine.
  - □ Second, programs run in this environment should show, at worst, only minor decreases in speed.
  - □ Third, a VMM should be in complete control of the system resources.

Comparison of Four VMM and Hypervisor Software Packages

Provider and References	Host CPU	Host OS	Guest OS	Architecture
VMware Workstation	x86, x86-64	Windows, Linux	Windows, Linux, Solaris, FreeBSD, Netware, OS/2, SCO, BeOS, Darwin	Full Virtualization
VMware ESX Server	x86, x86-64	No host OS	The same as VMware Workstation	Para-Virtualization
Xen	x86, x86-64, IA-64	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP and 2003 Server	Hypervisor
KVM	x86, x86-64,IA-64, S390, PowerPC	Linux	Linux, Windows, FreeBSD, Solaris	Para-Virtualization

Virtualization Support at the OS Level

Cloud computing has at least two challenges.

- The first is the ability to use a variable number of physical machines and VM instances depending on the needs of a problem. For example, a task may need only a single CPU during some phases of execution but may need hundreds of CPUs at other times.
- The second challenge concerns the slow operation of instantiating new VMs. Currently, new VMs originate either as fresh boots or as replicates of a template VM, unaware of the current application state. Therefore, to better support cloud computing, a large amount of research and development should be done.

Why OS-Level Virtualization?

- To reduce the performance overhead of hardware-level virtualization, even hardware modification is needed. OS-level virtualization provides a feasible solution for these hardware-level virtualization issues.
- Operating system virtualization inserts a virtualization layer inside an operating

system to partition a machine's physical resources. It enables multiple isolated VMs within a single operating system kernel.

- This kind of VM is often called a virtual execution environment (VE), Virtual Private System (VPS), or simply container. From the user's point of view, VEs look like real servers. This means a VE has its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings.
- Although VEs can be customized for different people, they share the same operating system kernel. Therefore, OS-level virtualization is also called single-OS image virtualization. Figure illustrates operating system virtualization from the point of view of a machine stack.

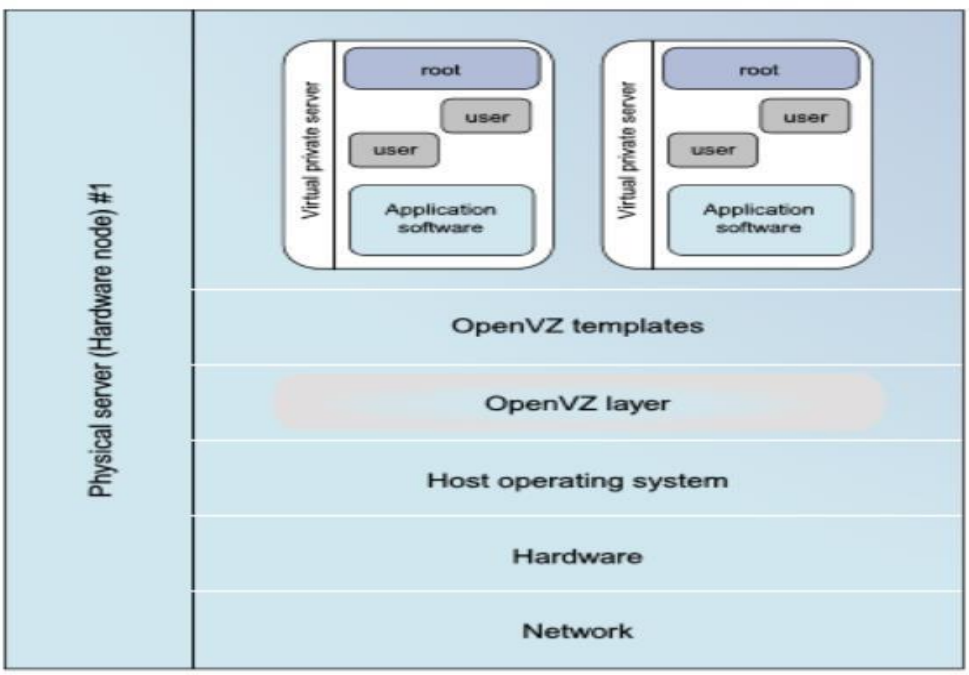


FIGURE The OpenVZ virtualization layer inside the host OS, which provides some OS images to create VMs quickly.

Advantages of OS Extension for Virtualization

1. VMs at OS level has minimum startup/shutdown costs
2. OS-level VM can easily synchronize with its environment

Disadvantage of OS Extension for Virtualization

- All VMs in the same OS container must have the same or similar guest OS, which restrict application flexibility of different VMs on the same physical machine.

Virtualization Support for Linux and Windows NT Platforms

Virtualization Support and Source of Information	Brief Introduction on Functionality and Application Platforms
Linux vServer for Linux platforms (http://linuxvserver.org/)	Extends Linux kernels to implement a security mechanism to help build VMs by setting resource limits and file attributes and changing the root environment for VM isolation
OpenVZ for Linux platforms [65]; http://ftp.openvz .org/doc/OpenVZ-Users-Guide.pdf)	Supports virtualization by creating virtual private servers (VPSes); the VPS has its own files, users, process tree, and virtual devices, which can be isolated from other VPSes, and checkpointing and live migration are supported
FVM (Feather- Weight Virtual Machines) for virtualizing the Windows NT platforms)	Uses system call interfaces to create VMs at the NY kernel space; multiple VMs are supported by virtualized namespace and copy- on-write

Middleware Support for Virtualization

Library-level virtualization is also known as user-level Application Binary Interface (ABI) or API emulation. This type of virtualization can create execution environments for running alien programs on a platform rather than creating a VM to run the entire operating system. API call interception and remapping are the key functions performed.

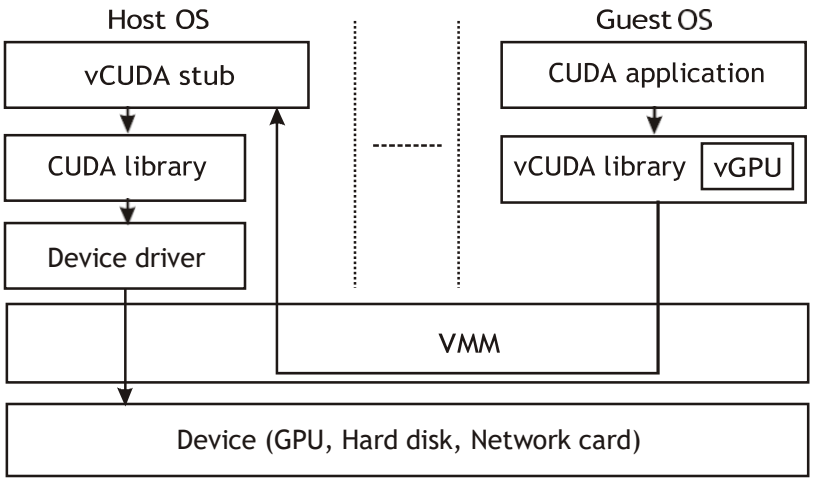
The library-level virtualization systems: namely the Windows Application Binary Interface (WABI), lxrun, WINE, Visual MainWin, and vCUDA, which are summarized in Table.

Middleware and Library Support for Virtualization

Middleware or Runtime Library and References or Web Link	Brief Introduction and Application Platforms.
WABI (http://docs.sun.com/app/ docs/doc/802-6306)	Middleware that converts Windows system calls running on x86 PCs to Solaris system calls running on SPARC workstations
Lxrun (Linux Run) (http:// www.ugcs.caltech.edu/~steven/ lxrun/)	A system call emulator that enables Linuxapplications written for x86 hosts to run on UNIX systems such as the SCO OpenServer
WINE (http://www.winehq.org/)	A library support system for virtualizing x86processors to run Windows applications underLinux, FreeBSD, and Solaris
Visual MainWin (http:// www.mainsoft.com/)	A compiler support system to develop Windows applications using Visual Studio to run on Solaris, Linux, and AIX hosts
vCUDA (Example 2.4) (IEEE IPDPS 2009)	Virt ualization support for using general-purpose GPUs to run data- intensive applications under a special guest OS

- The WABI offers middleware to convert Windows system calls to Solaris system calls. Lxrun is really a system call emulator that enables Linux applications written for x86 hosts to run on UNIX systems. Similarly, Wine offers library support for virtualizing x86 processors to run Windows applications on UNIX hosts.
- Visual MainWin offers a compiler support system to develop Windows applications using Visual Studio to run on some UNIX hosts. The vCUDA is explained in Example with a graphical illustration in Figure.

Basic concept of the vCUDA architecture.



Example The vCUDA for Virtualization of General-Purpose GPUs

- CUDA is a programming model and library for general-purpose GPUs. It leverages the high performance of GPUs to run compute-intensive applications on host operating systems. However, it is difficult to run CUDA applications on hardware-level VMs directly. vCUDA virtualizes the CUDA library and can be installed on guest OSes.
- When CUDA applications run on a guest OS and issue a call to the CUDA API, Vcuda intercepts the call and redirects it to the CUDA API running on the host OS. Figure 2.7 shows the basic concept of the vCUDA architecture.
- The vCUDA employs a client- server model to implement CUDA virtualization. It consists of three user space components: the vCUDA library, a virtual GPU in the guest OS (which acts as a client), and the vCUDA stub in the host OS (which acts as a server).
- The vCUDA library resides in the guest OS as a substitute for the standard CUDA library. It is responsible for intercepting and redirecting API calls from the client to the stub. Besides these tasks, vCUDA also creates vGPUs and manages them.
- The functionality of a vGPU is threefold: It abstracts the GPU structure and gives applications a uniform view of the underlying hardware; when a CUDA application in the guest OS allocates a device's memory the vGPU can return a local virtual address to the application and notify the remote stub to allocate the real device memory, and the vGPU is responsible for storing the CUDA API flow.
- The vCUDA stub receives and interprets remote requests and creates a corresponding execution context for the API calls from the guest OS, then

2.6 VIRTUALIZATION STRUCTURES/TOOLS AND MECHANISMS

Figure showed the architectures of a machine before and after virtualization.

- Before virtualization, the operating system manages the hardware.
- After virtualization, a virtualization layer is inserted between the hardware and the operating system. In such a case, the virtualization layer is responsible for converting portions of the real hardware into virtual hardware.
- Therefore, different operating systems such as Linux and Windows can run on the same physical machine, simultaneously.
- Depending on the position of the virtualization layer, there are several classes of VM architectures, namely the hypervisor architecture, para-virtualization, and host-based virtualization.
- The hypervisor is also known as the VMM (Virtual Machine Monitor). They both perform the same virtualization operations.

2.6.1 Hypervisor

- A hypervisor is a hardware virtualization technique allowing multiple operating systems, called guests to run on a host machine. This is also called the Virtual Machine Monitor (VMM).

Type 1: bare metal hypervisor

- sits on the bare metal computer hardware like the CPU, memory, etc.
- All guest operating systems are a layer above the hypervisor.
- The original CP/CMS hypervisor developed by IBM was of this kind.

Type 2: hosted hypervisor

- Run over a host operating system.
- Hypervisor is the second layer over the hardware.
- Guest operating systems run a layer over the hypervisor.
- The os is usually unware of the virtualization

2.6.2 The XEN Architecture

- Xen is an open source hypervisor program developed by Cambridge University. Xen is a micro- kernel hypervisor, which separates the policy from the mechanism.

- Xen does not include any device drivers natively . I t just provides a mechanism by which a guest OS can have direct access to the physical devices.
- As a result, the size of the Xen hypervisor is kept rather small. Xen provides a Virtual environment located between the hardware and OS.

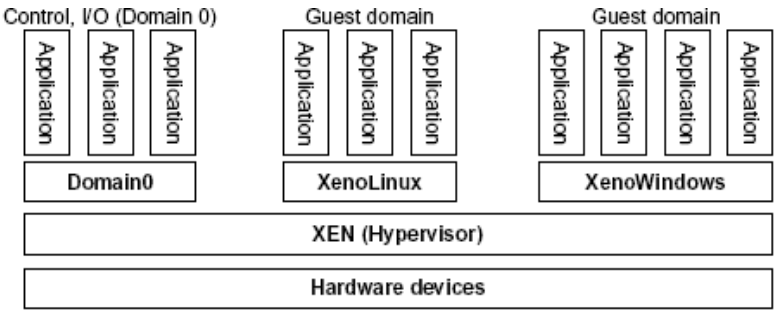


FIGURE 3.5

The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.

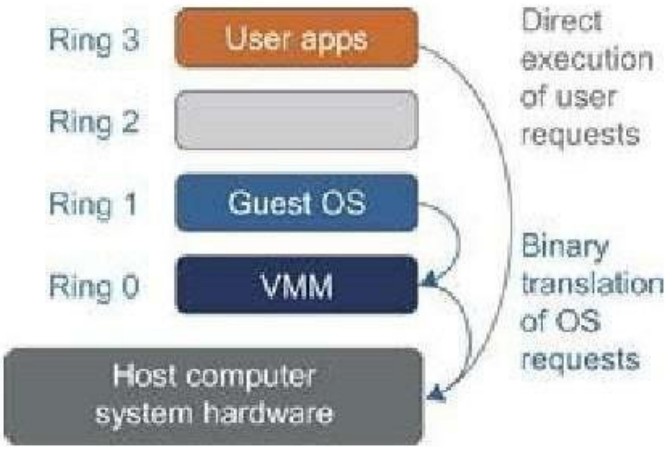
2.6.3 Binary Translation with Full Virtualization

- Depending on implementation technologies, hardware virtualization can be classified into two categories: full virtualization and host-based virtualization.
- Full virtualization does not need to modify the host OS. I t relies on binary translation to trap and to virtualizes the execution of certain sensitive, non virtualizable instructions. The guest OSes and their applications consist of noncritical and critical instructions.
- I n a host-based system, both a host OS and a guest OS are used. A virtualization software layer is built between the host OS and guest OS.
- These two classes of VM architecture are introduced next.

Binary Translation of Guest OS Requests Using a VMM

- This approach was implemented by VMware and many other software companies.
- VMware puts the VMM at Ring 0 and the guest OS at Ring 1. The VMM scans the instruction stream and identified the privileged, control- and behavior sensitive instructions.
- When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions.

- The method used in this emulation is called binary translation. Therefore, full virtulation combines binary translation and direct execution.

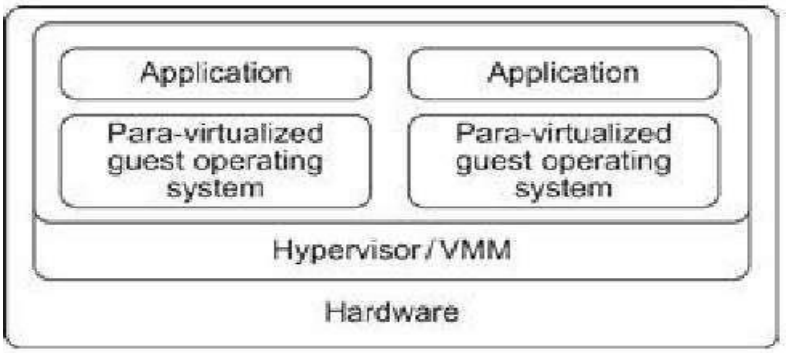


2.6.4 Host-Based Virtualization

- An alternative VM architecture is to install a virtualization layer on top of the host OS. This host OS is still responsible for managing the hardware.
- 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. This will simplify the VM design and ease its deployment.
- Second, the host-based approach appeals to many host machine configurations. Compared to the hypervisor/VMM architecture, the performance of the host-based architecture may also be low.

2.6.5 Para –virtualization

- Para -virtualization needs to modify the guest operating systems. A para-virtualized VM
- provides special API s requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system.



The use of para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls

2.6.5.1 Para-Virtualization Architecture

- When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS. According to the x86 ring definition, the virtualization layer should also be installed at Ring 0. Different instructions at Ring 0 may cause some problems.
- In Figure, we show that para-virtualization replaces nonvirtualizable instructions with hypercalls that communicate directly with the hypervisor or VMM. However, when the guest OS kernel is modified for virtualization, it can no longer run on the hardware directly.

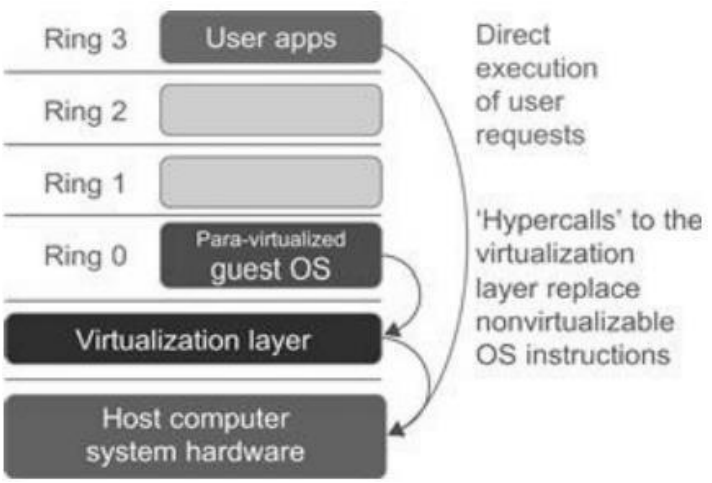


FIGURE The use of a para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls

- Compared with full virtualization, para-virtualization is relatively easy and more practical. The main problem in full virtualization is its low performance in binary translation. To speed up binary translation is difficult. Therefore, many virtualization products employ the para-virtualization architecture. The popular Xen, KVM, and VMware ESX are good examples.

2.6.5.2 KVM (Kernel-Based VM)

- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel. Memory management and scheduling activities are carried out by the existing Linux kernel.
- The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine. KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSessuch as Windows, Linux, Solaris, and other UNIX variants.

2.6.5.3 Para-Virtualization with Compiler Support

Unlike the full virtualization architecture which intercepts and emulates privileged and sensitive instructions at runtime, para-virtualization handles these instructions at compile time. The guest OS kernel is modified to replace the privileged and sensitive instructions with hypercalls to the hypervisor or VMM. Xen assumes such a para-virtualization architecture.

The guest OS running in a guest domain may run at Ring 1 instead of at Ring 0. This implies that the guest OS may not be able to execute some privileged and sensitive instructions. The privileged instructions are implemented by hypercalls to the hypervisor. After replacing the instructions with hypercalls, the modified guest OS emulates the behavior of the original guest OS. On an UNIX system, a system call involves an interrupt or service routine. The hypercalls apply a dedicated service routine in Xen.

Example VMware ESX Server for Para-Virtualization

VMware pioneered the software market for virtualization. The company has developed virtualization tools for desktop systems and servers as well as virtual infrastructure for large data centers. ESX is a VMM or a hypervisor for bare-metal x86 symmetric multiprocessing (SMP) servers.

It accesses hardware resources such as I/O directly and has complete resource management control. An ESX-enabled server consists of four components

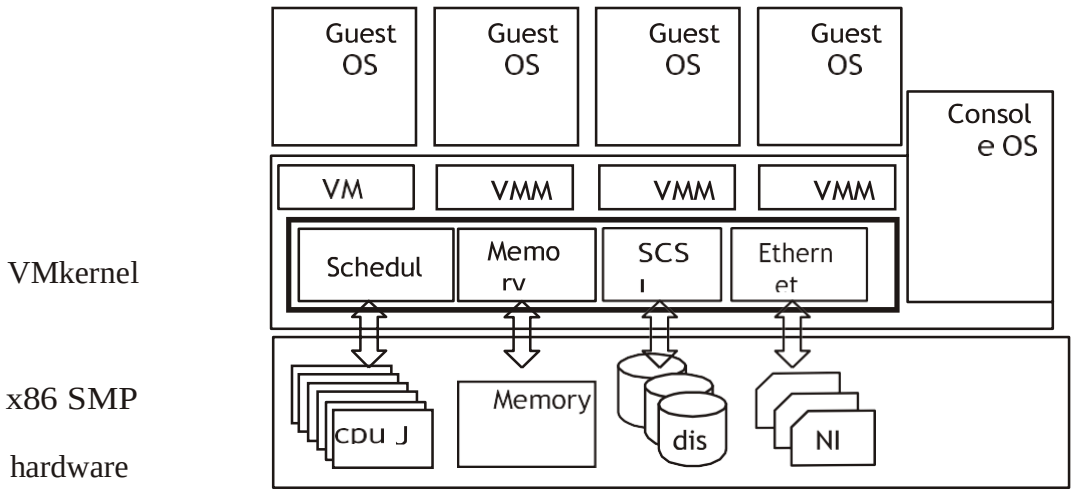


FIGURE The VMware ESX server architecture using para-virtualization.

Full virtualization vs. Para virtualization

Full virtualization

- Does not need to modify guest OS, and critical instructions are emulated by software through the use of binary translation.
- VMware Workstation applies full virtualization, which uses binary translation to automatically modify x86 software on-the-fly to replace critical instructions.

Advantage: no need to modify OS.

Disadvantage: binary translation slows down the performance.

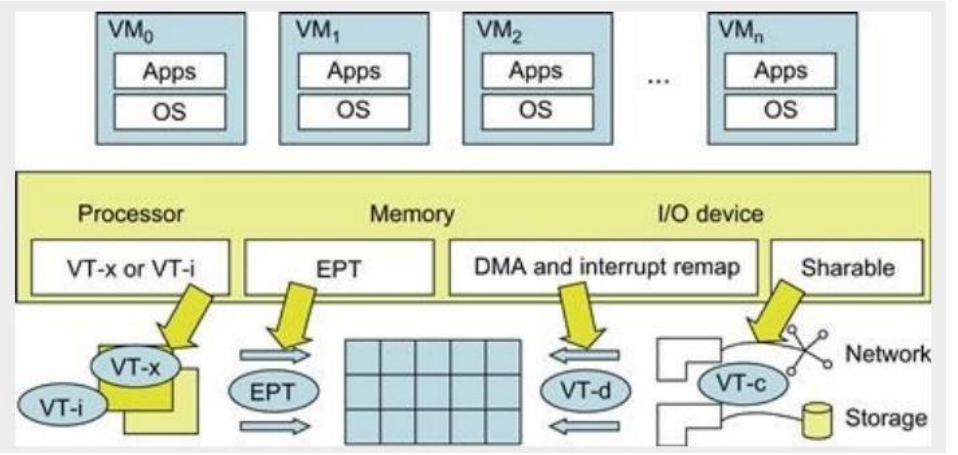
Para virtualization

- Reduces the overhead, but cost of maintaining a paravirtualized OS is high.
- The improvement depends on the workload.
- Para virtualization must modify guest OS, non-virtualizableinstructions are replaced by hyper calls that communicate directly with the hypervisor or VMM.

2.7 VIRTUALIZATION OF CPU, MEMORY, AND I/O DEVICES

2.7.1 Hardware Support for Virtualization

Modern operating systems and processors permit multiple processes to run simultaneously. If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash. Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware. Instructions running in supervisor mode are called privileged instructions. Other instructions are unprivileged instructions. The VMware Workstation is a VM software suite for x86 and x86-64 computers. This software suite allows users to set up multiple x86 and x86-64 virtual computers and to use one or more of these VMs simultaneously with the host operating system. The VMware Workstation assumes the host-based virtualization. Xen is a hypervisor for use in IA-32, x86-64, Itanium, and PowerPC 970 hosts.



2.7.2 CPU Virtualization

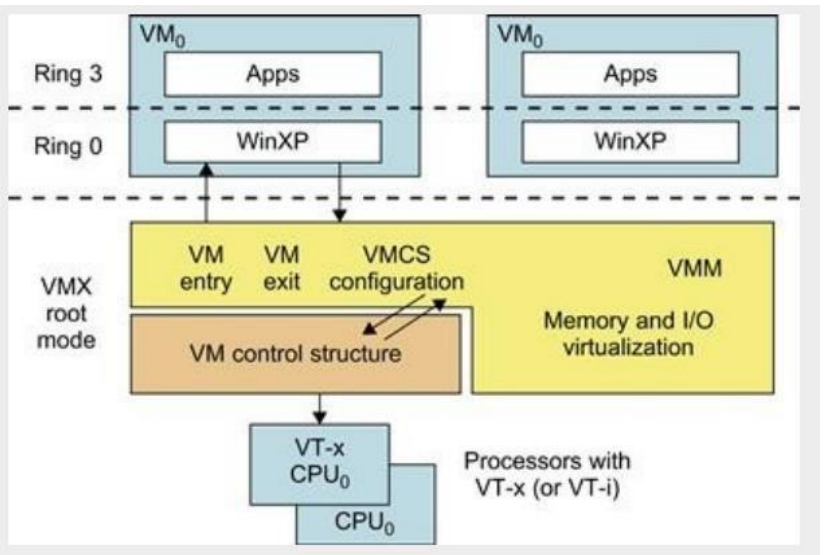
A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode. Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency. Other critical instructions should be handled carefully for correctness and stability. The critical instructions are divided into three categories: privileged instructions, control-sensitive instructions, and behavior-sensitive instructions. Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode. Control-sensitive instructions attempt to change the configuration of resources used. Behavior-sensitive

instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

The VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system.

2.7.2.1. Hardware-Assisted CPU Virtualization

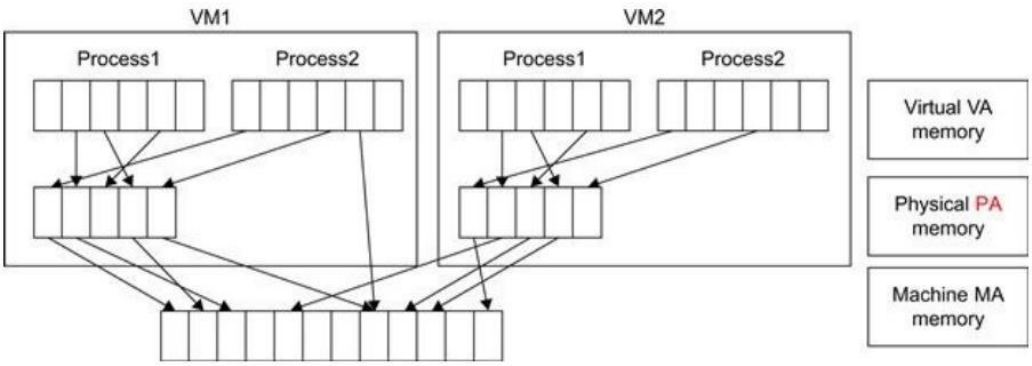
This technique attempts to simplify virtualization because full or paravirtualization is complicated. Intel and AMD add an additional mode called privilege mode level to x86 processors.



2.7.3 Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. A two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory.

Two level memory mapping process

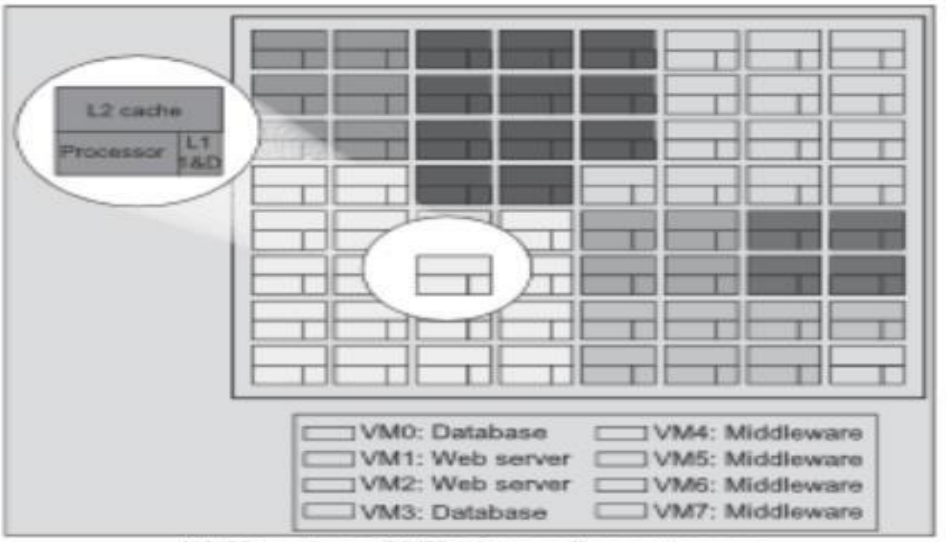


The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. The MMU already handles virtual-to-physical translations as defined by the OS. Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor. Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access. When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.

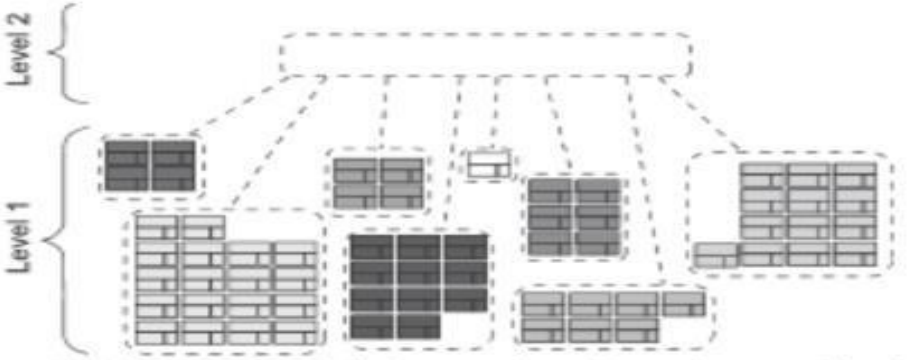
Example: Extended Page Table by Intel for Memory Virtualization

- In Figure, the page tables of the guest OS and EPT are all four-level.
- When a virtual adderss needs to be translated, the CPU will first look for the L4 page table pointed to by Guest CR3. Since the address in Guest CR3 is a physical address in the guest OS, the CPU needs to convert the Guest CR3 GPA to the host physical address (HPA) using EPT.
- In this procedure, the CPU will check the EPT TLB to see if the translation is there. If there is no required translation in the EPT TLB, the CPU will look for it in the EPT. If the CPU cannot find the translation in the EPT, an EPT violation exception will be raised.
- When the GPA of the L4 page table is obtained, the CPU will calculate the GPA of the L3 page table by using the GVA and the content of the L4 page table. If the entry corresponding to the GVA in the L4 page table is a page fault, the CPU will generate a page fault interrupt and will let the guest OS kernel handle the interrupt.
- When the PGA of the L3 page table is obtained, the CPU will look for the EPT to get the HPA of the L3 page table, as described earlier. To get the HPA corresponding to a

GVA, the CPU needs to look for the EPT five times, and each time, the memory needs to be accessed four times. Therefore, there are 20 memory accesses in the worst case, which is still very slow. To overcome this shortcoming, Intel increased the size of the EPT TLB to decrease the number of memory accesses.



(a) Mapping of VMs into adjacent cores



(b) Multiple virtual clusters assigned to various workloads

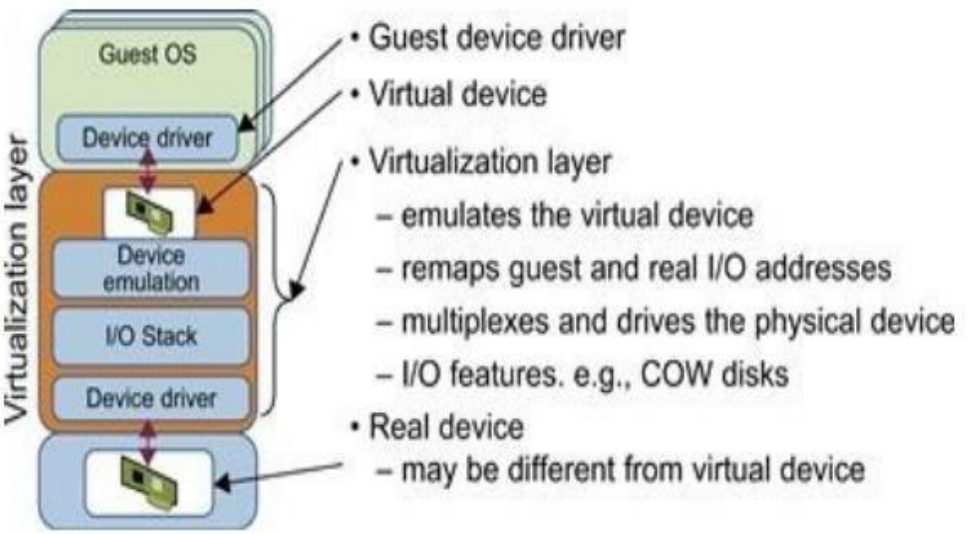
2.7.4 I/O Virtualization

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware. At the time of this writing, there are **three ways to implement I/O virtualization: full device emulation, para-virtualization, and direct I/O.**

Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well-known, real-world devices. All the functions of a device or bus

infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.

Full device emulation



Device emulation for I/O virtualization implemented inside the middle layer that maps real I/O devices into the virtual devices for the guest device driver to use.

The para-virtualization method of I/O virtualization is typically used in Xen. It is also known as the split driver model consisting of a frontend driver and a backend driver. The frontend driver is running in **Domain U** and the backend driver is running in **Domain 0**.They interact with each other via a block of shared memory. The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs.

Direct I/O virtualization lets the VM access devices directly. It can achieve close-to-native performance without high CPU cost.

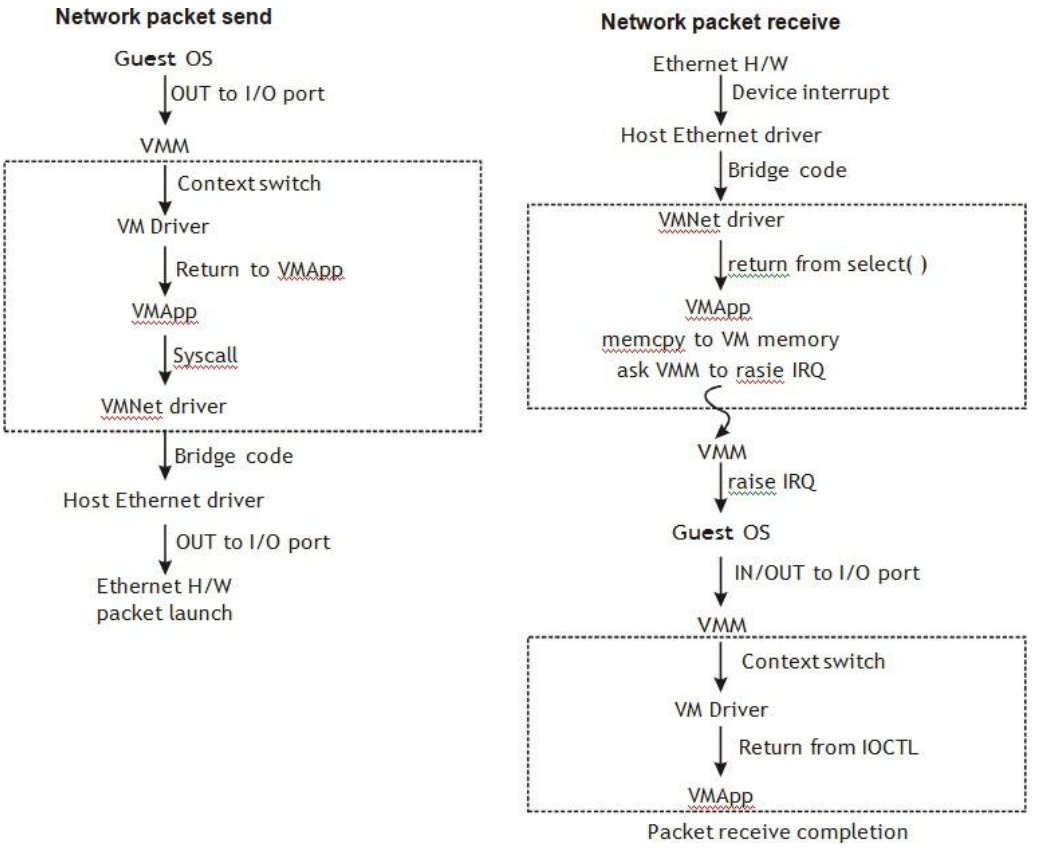
- For example, when a physical device is reclaimed (required by workload migration) for later reassignment, it may have been set to an arbitrary state (e.g., DMA to some arbitrary memory locations) that can function incorrectly or even crash the whole

system. Since software-based I/O virtualization requires a very high overhead of device emulation, hardware-assisted I/O virtualization is critical.

- Intel VT-d supports the remapping of I/O DMA transfers and device- generated interrupts. The architecture of VT-d provides the flexibility to support multiple usage models that may run unmodified, special-purpose, or "virtualization-aware" guest OSes.
- Another way to help I/O virtualization is via self-virtualized I/O (SV-IO). The key idea of SV-IO is to harness the rich resources of a multicore processor. All tasks associated with virtualizing an I/O device are encapsulated in SV-IO. It provides virtual devices and an associated access API to VMs and a management API to the VMM.
- SV-IO defines one virtual interface (VIF) for every kind of virtualized I/O device, such as virtual network interfaces, virtual block devices (disk), virtual camera devices, and others. The guest OS interacts with the VIFs via VIF device drivers. Each VIF consists of two message queues. One is for outgoing messages to the devices and the other is for incoming messages from the devices. In addition, each VIF has a unique ID for identifying it in SV-IO.

VMware Workstation for I/O Virtualization

- The VMware Workstation runs as an application. It leverages the I/O device support in guest OSes, host OSes, and VMM to implement I/O virtualization. The application portion (VMApp) uses a driver loaded into the host operating system (VMDriver) to establish the privileged VMM, which runs directly on the hardware.
- A given physical processor is executed in either the host world or the VMM world, with the VMDriver facilitating the transfer of control between the two worlds. The VMware Workstation employs full device emulation to implement I/O virtualization. Figure shows the functional blocks used in sending and receiving packets via the emulated virtual NIC.



- The virtual NIC models an AMD Lance Am79C970A controller. The device driver for a Lance controller In the guest OS initiates packet transmissions by reading and writing a sequence of virtual I/O ports; each read or write switches back to the VMApp to emulate the Lance port accesses.
- When the last OUT instruction of the sequence is encountered, the Lance emulator calls a normal write() to the VMNet driver. The VMNet driver then passes the packet onto the network via a host NIC and then the VMApp switches back to the VMM. The switch raises a virtual interrupt to notify the guest device driver that the packet was sent. Packet receives occur in reverse.

2.7.5 Virtualization in Multi-Core Processors

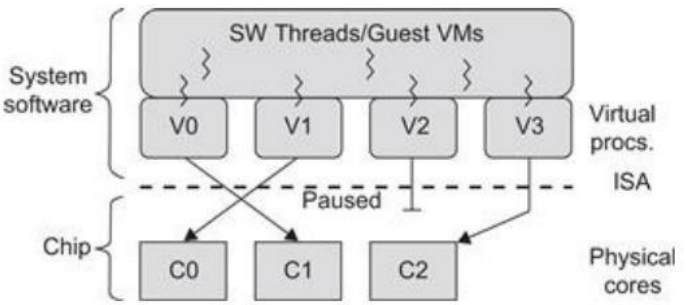
Virtualizing a multi-core processor is relatively more complicated than virtualizing a unicore processor. Though multicore processors are claimed to have higher performance by integrating multiple processor cores in a single chip, muti-core virtualization has raised

some new challenges to computer architects, compiler constructors, system designers, and application programmers. There are mainly two difficulties: Application programs must be parallelized to use all cores fully, and software must explicitly assign tasks to the cores, which is a very complex problem.

Physical versus Virtual Processor Cores

This technique alleviates the burden and inefficiency of managing hardware resources by software. It is located under the ISA and remains unmodified by the operating system or VMM (hypervisor).

Multicore virtualization method that exposes four VCPUs to the software, when only three cores are actually present.



Virtual Hierarchy

The emerging many-core chip multiprocessors (CMPs) provides a new computing landscape. To optimize for space-shared workloads, they propose using virtual hierarchies to overlay a coherence and caching hierarchy onto a physical processor. Unlike a fixed physical hierarchy, a virtual hierarchy can adapt to fit how the work is space shared for improved performance and performance isolation.

- Figure illustrates a logical view of such a virtual cluster hierarchy in two levels. Each VM operates in a isolated fashion at the first level. This will minimize both miss access time and performance interference with other workloads or VMs.
- Moreover, the shared resources of cache capacity, inter-connect links, and miss handling are mostly isolated between VMs. The second level maintains a globally shared memory. This facilitates dynamically repartitioning resources without

costly cache flushes. Furthermore, maintaining globally shared memory minimizes changes to existing system software and allows virtualization features such as content-based page sharing.

- A virtual hierarchy adapts to space-shared workloads like multiprogramming and server consolidation. Figure shows a case study focused on consolidated server workloads in a tiled architecture. This many-core mapping scheme can also optimize for space-shared multiprogrammed workloads in a single-OS environment.

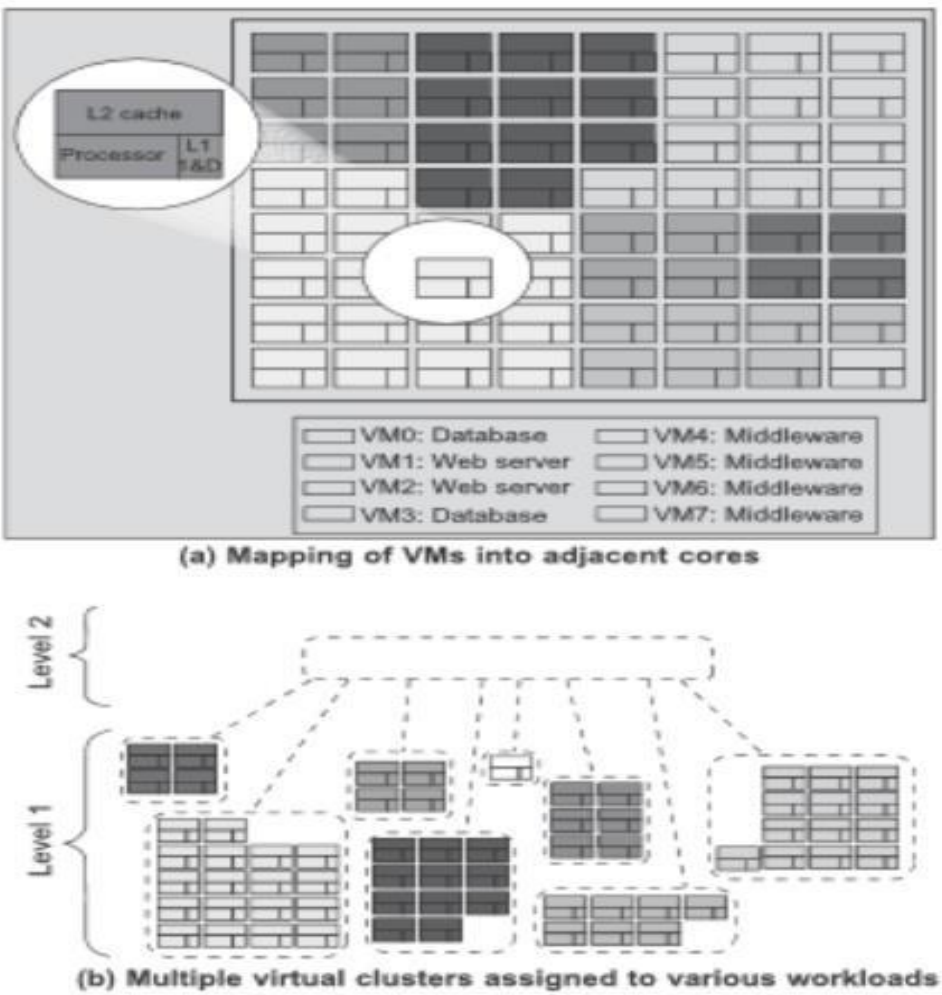
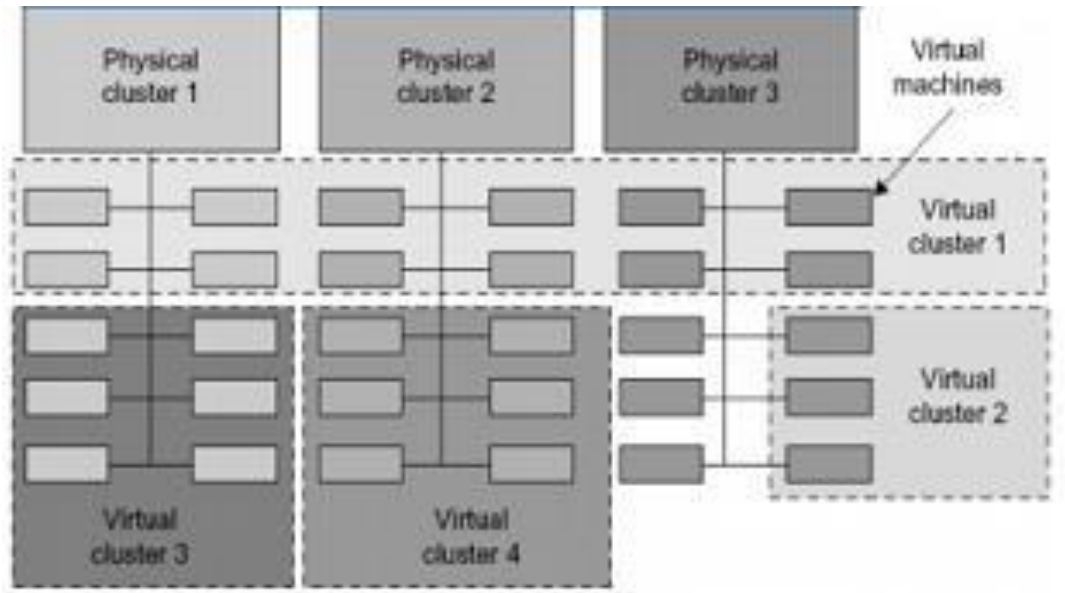


FIGURE: CMP server consolidation by space-sharing of VMs into many cores forming multiple virtual clusters to execute various workloads.

Virtual clusters and resource management?

A physical cluster is a collection of servers (physical machines) interconnected by a physical network such as a LAN.

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. Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters. The virtual cluster boundaries are shown as distinct boundaries.



The provisioning of VMs to a virtual cluster is done dynamically to have the following Interesting properties:

• The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OSes can be deployed on the same physical node.

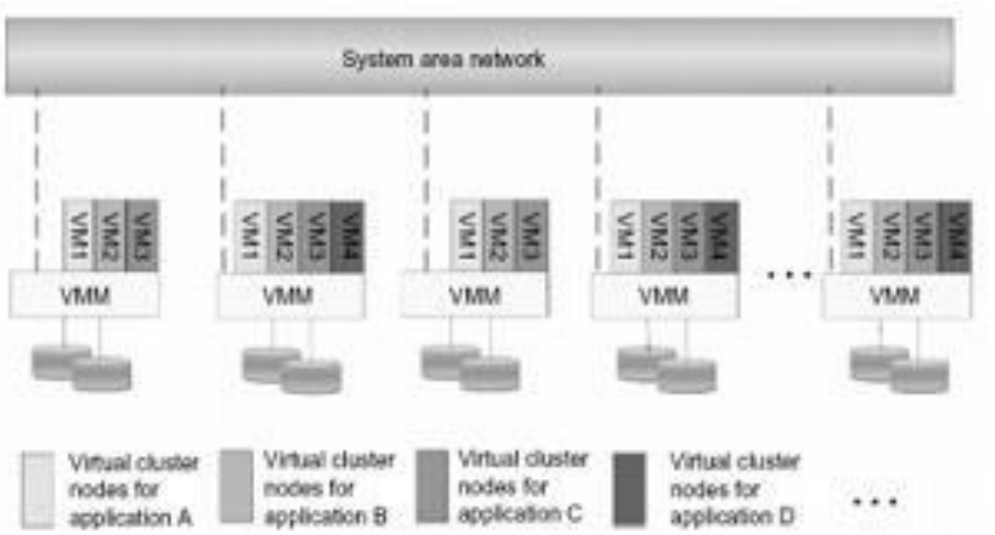
• A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.

• The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance server utilization and application flexibility.

• VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.

• The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similar to the way an overlay network varies in size in a peer-to-peer (P2P) network.

• The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.



A Virtual Clusters based on Application Partitioning:

Parallax Providing Virtual Disks to Clients VMs from a Large Common Shared Physical Disk.

2.8 VIRTUALIZATION SUPPORT AND DISASTER RECOVERY

- One very distinguishing feature of cloud computing infrastructure is the use of system virtualization and the modification to provisioning tools. Virtualization of servers on a shared cluster can consolidate web services.
- As the VMs are the containers of cloud services, the provisioning tools will first find the corresponding physical machines and deploy the VMs to those nodes before scheduling the service to run on the virtual nodes.
- In addition, in cloud computing, virtualization also means the resources and fundamental infrastructure are virtualized. The user will not care about the computing resources that are used for providing the services.
- Cloud users do not need to know and have no way to discover physical resources that are involved while processing a service request. Also,

application developers do not care about some infrastructure issues such as scalability and fault tolerance (i.e., they are virtualized).

- Application developers focus on service logic. Figure 2.21 shows the infrastructure needed to virtualize the servers in a data center for implementing specific cloud applications.

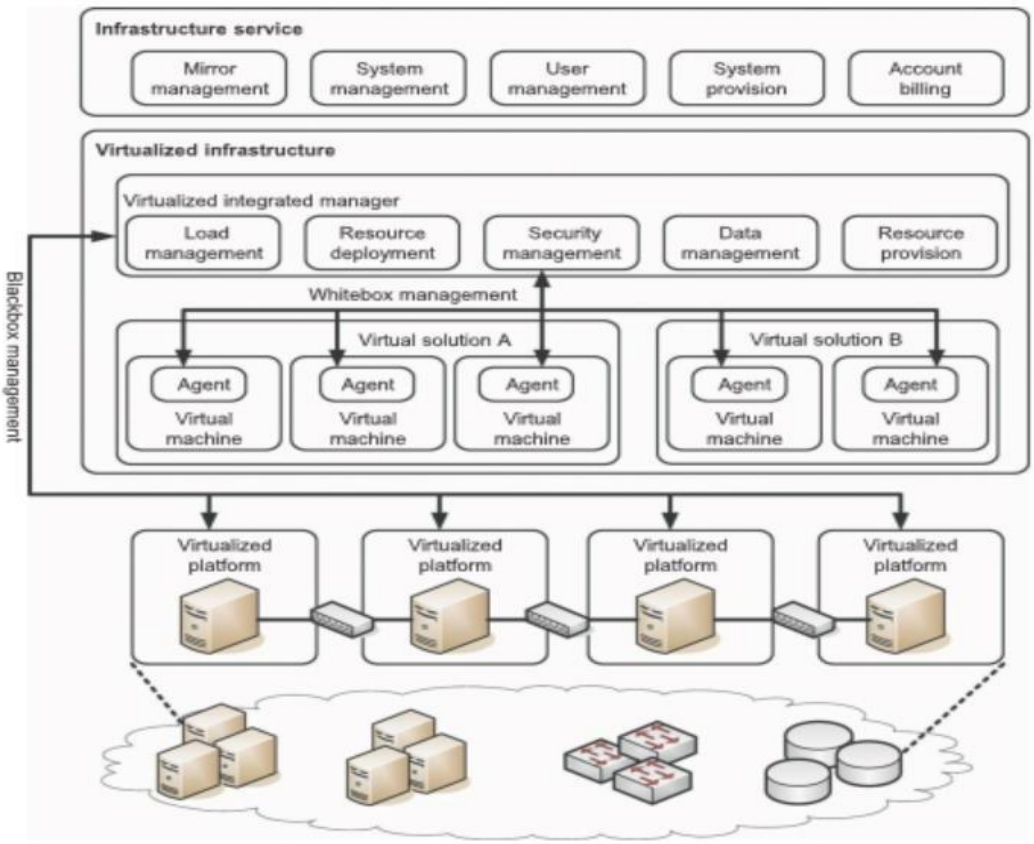


FIGURE: Virtualized servers, storage, and network for cloud platform construction.

2.8.1 Hardware Virtualization

- In many cloud computing systems, virtualization software is used to virtualize the hardware. System virtualization software is a special kind of software which simulates the execution of hardware and runs even unmodified operating systems.
- Cloud computing systems use virtualization software as the running environment for legacy software such as old operating systems and unusual

applications.

- Virtualization software is also used as the platform for developing newcloud applications that enable developers to use any operating systems and programming environments they like.
- The development environment and deployment environment can now be the same, which eliminates some runtime problems.
- Some cloud computing providers have used virtualization technology to provide this service for developers. As mentioned before, system virtualization software is considered the hardware analog mechanism to run an unmodified operating system, usually on bare hardware directly, on top of software.
- Table lists some of the system virtualization software in wide use at the time of this writing. Currently, the VMs installed on a cloud computing platform are mainly used for hosting third-party programs. VMs provide flexible runtime services to free users from worrying about the system environment.

Table Virtualized Resources in Compute, Storage, and Network Clouds

Provider	AWS	Microsoft Azure	GAE
Compute cloud with virtual cluster of servers	x86 instruction set, Xen VMs, resource elasticity allows scalability through virtual cluster, or a third party such as RightScale must provide the cluster.	Common language runtime VMs provisioned by declarative descriptions.	Predefined application framework handlers written in Python, automatic scaling up and down, server failover nconsistent with the web applications
Storage cloud with virtual storage	Models for block store (EBS) and augmented key/blob store (SimpleDB), utomatic scaling varies from EBS to fully automatic (SimpleDB, S3).	SQL Data Services (restricted view of SQL Server), Azure storage service.	MegaStore/BigTable

Network cloud services	Declarative IP-level topology; placement details hidden, security groups restricting communication, avai-lability zones isolate network failure, elasticIP applied.	Automatic with user's declarative descriptions or roles of app. components.	Fixed topology to accommodate three-tier web app. structure, scaling up and down is automatic and programmer-invisible

- Traditional sharing of cluster resources depends on the user and group mechanism on a system. Such sharing is not flexible. Users cannot customize the system for their special purposes. Operating systems cannot be changed. The separation is not complete.

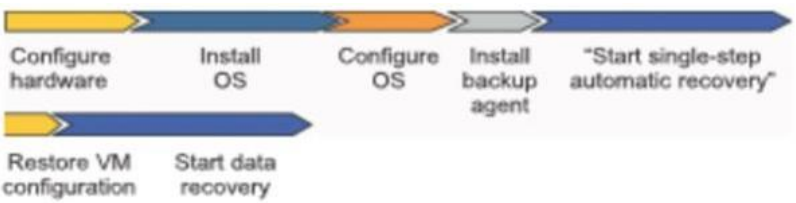


FIGURE Recovery overhead of a conventional disaster recovery scheme, compared with that required to recover from live migration of VMs.

2.8.1 VM Cloning for Disaster Recovery

- VM technology requires an advanced disaster recoveryscheme. One scheme is to recover one physical machine by another physical machine.
- The second scheme is to recover one VM by another VM. As shown in the top timeline of Figure, traditional disaster recovery from one physical machine to another is rather slow, complex, and expensive.
- Total recovery time is attributed to the hardware configuration, installing and configuring the OS, installing the backup agents, and the long time to restart the physical machine.
- To recover a VM platform, the installation and configuration times for the OS and backup agents are eliminated. Therefore, we end up with a much shorter disaster recovery time, about 40 percent of that to recover the physical machines. Virtualization aids in fast disaster recovery by VM encapsulation

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

UNIT 2

PART A

1. List out the four principles of REST architectural style.

- Resource Identification through URIs
- Uniform, Constrained Interface
- Self- Descriptive Message
- Stateless Interactions

2. Define Self-Descriptive Message.

- A REST message includes enough information to describe how to process the message. This enables intermediaries to do more with the message without parsing themessage contents.
- In REST, resources are decoupled from their representation so that their content

can be accessed in a variety of standard formats

3. **Give an example for RESTful web service.**

- A good example of RESTful web service application in high-performance computing systems is the Amazon Simple Storage Service (S3) interface.
- Amazon S3 is data storage for Internet applications. It provides simple web services to store and retrieve data from anywhere at any time via the web.
- S3 keeps fundamental entities, "objects," which are named pieces of data accompanied by some metadata to be stored in containers called "buckets," each identified by a unique key.

4. **What are the resources provided by Amazon S3?**

Amazon S3 provides three types of resources: a list of user buckets, a particular bucket, and a particular S3 object, accessible through https:// s3.amazonaws.com/{name-of-bucket}/{name-of-object}.

5. **List out the basic HTTP standard operations.**

These resources are retrieved, created, or manipulated by basic HTTP standard operations: GET, HEAD, PUT, and DELETE.

- GET can be used to list buckets created by the user, objects kept inside a bucket, or an object's value and its related metadata.
- PUT can be used for creating a bucket or setting an object's value or metadata
- DELETE for removing a particular bucket or object, and
- HEAD for getting a specific object's metadata.

6. **What is web services?**

- The term "web service" is often referred to a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web.
- A web service has an interface described in a machine-executable format

(specifically Web Services Description Language or WSDL).

7. Define SOAP protocol.

- Simple Object Access Protocol (SOAP) SOAP provides a standard packaging structure for transmission of XML documents over various Internet protocols, such as SMTP, HTTP, and FTP.
- By having such a standard message format, heterogeneous middleware systems can achieve interoperability.

8. What is WSDL?

- Web Services Description Language (WSDL) WSDL describes the interface, a set of operations supported by a web service in a standard format.
- It standardizes the representation of input and output parameters of its operations as well as the service's protocol binding, the way in which the messages will be transferred on the wire.
- Using WSDL enables disparate clients to automatically understand how to interact with a web service.

9. Define UDDI and its usage in web service?

Universal Description, Discovery, and Integration (UDDI) UDDI provides a global registry for advertising and discovery of web services, by searching for names, identifiers, categories, or the specification implemented by the web service.

10. Discuss massive parallel processing.

- *Massive parallel processing* is used in computer architecture circles to refer to a computer system with many independent arithmetic units or entire microprocessors, which run in parallel.
- "Massive" connotes hundreds if not thousands of such units. In this form of computing, all the processing elements are interconnected to act as one very large computer.

11. What is meant by virtualization?

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.
- The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

12. What is the use of Hardware Virtualization technology?

- This is one of the most common types of Virtualization as it is related to the application uptime and utilization of hardware. The primary motive behind this technology brings together all the physical servers into one large physical server. Thus making the processor work more efficiently and effectively.
- The operating system of the physical server gets transformed into a well to do OS that runs on a virtual system. It consists of a hypervisor that solely manages the memory, processor and other elements by allowing multiple OS to run on the same machine without the help of any source code.

13. List out the types of Hardware Virtualization

- Full Virtualization
- Para Virtualization
- Partial Virtualization

14. What is Software Virtualization?.

Software Virtualization comprises of the ability to the primary system to create and run more virtual environment. It enables a computer system to allow a guest OS to run. For example, Linux could run as a guest to run a Microsoft Windows OS.

15. What is Storage Virtualization and its advantages?

- 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 also has many advantages with this as it has the capacity to reduce downtime, speed, performance and load balancing.

16. What are the levels of virtualization?

- Hardware Virtualization
- Virtual Machine
- Storage Virtualization
- Desktop Virtualization
- Network Virtualization

17. List out the three requirements for a VMM.

- First, a VMM should provide an environment for programs which is essentially identical to the original machine.
- Second, programs run in this environment should show, at worst, only minor decreases in speed.
- Third, a VMM should be in complete control of the system resources.

18. What is the need for OS-Level Virtualization?

- It is slow to initialize a hardware-level VM because each VM creates its own image from scratch. In a cloud computing environment, perhaps thousands

of VMs need to be initialized simultaneously.

- Besides slow operation, storing the VM images also becomes an issue. As a matter of fact, there is considerable repeated content among VM images.
- CUDA is a programming model and library for general-purpose GPUs. It leverages the high performance of GPUs to run compute-intensive applications on host operating systems.

19. What is meant by XEN hypervisor?

- Xen is an open source hypervisor program developed by CambridgeUniversity.
- Xen is a micro-kernel hypervisor, which separates the policy from the mechanism. The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain 0.

20. What is the feature of Full Hypervisor ?

- The Noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software.
- Both the hypervisor and VMM approaches are considered full virtualization.

21. What is meant by Para-virtualization ?

- Para-virtualization needs to modify the guest operating systems.
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system.

22. Give short notes on KVM.

- The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine.
- KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

23. What are the categories of instructions?

1. privileged instructions,
2. control-sensitive instructions,
3. and behavior-sensitive instructions.

24. Differentiate Physical versus Virtual Clusters?

- 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.
- Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters.

25. What are the four steps to deploy a group of VMs onto a target cluster?

- Preparing the disk image,
- configuring the VMs,
- choosing the destination nodes,
- and executing the VM deployment command on every host.

26. List out the states of VM?

- An inactive state is defined by the virtualization platform, under which the VM is not enabled.
- An active state refers to a VM that has been instantiated at the virtualization platform to perform a real task.
- A paused state corresponds to a VM that has been instantiated but disabled to process a task or paused in a waiting state.
- A VM enters the suspended state if its machine file and virtual resources are stored back to the disk.

27. What are the side effects of server virtualization?

- Consolidation enhances hardware utilization
- This approach enables more agile provisioning and deployment of resources.
- The total cost of ownership is reduced.
- This approach improves availability and business continuity.
- It becomes easier to transfer a VM from one server to another, because virtual servers are unaware of the underlying hardware.

PART B

1. Explain REST Architectural Elements and its interaction in detail.
2. Discuss about services and web services and WS-I protocol stack
3. Explain Publish-Subscribe Model with example.
4. What is virtualization? Explain different types of virtualization in detail.
5. Discuss the implementation levels of virtualization with neat block diagram.
6. Explain OS-level virtualization and discuss its advantages and disadvantages.

7. Discuss the concepts of vCUDA architecture in detail.
8. Explain how the Xen hypervisor provides virtual environment in cloud?
9. What is Binary Translation? Explain in detail with examples.
10. Discuss para-virtualization and how it is implemented in VMware ESX server with neat diagram.
11. Explain how the virtualization implemented in CPU, Memory and I/O devices.

UNIT III

CLOUD ARCHITECTURE, SERVICES AND STORAGE

Layered Cloud Architecture Design - NIST Cloud Computing Reference Architecture - Public, Private and Hybrid Clouds - laaS - PaaS - SaaS - Architectural Design Challenges - Cloud Storage - Storage-as-a-Service - Advantages of Cloud Storage - Cloud Storage Providers - S3.

3.1 LAYERED CLOUD ARCHITECTURE DESIGN

The architecture of a cloud is developed at three layers:

- Infrastructure,
- Platform
- Application

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

- The platform layer is for general-purpose and repeated usage of the collection of software resources. This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance. The platform should be able to assure users that they have scalability, dependability, and security protection.
- In a way, the virtualized cloud platform serves as a "system middleware" between the infrastructure and application layers of the cloud.

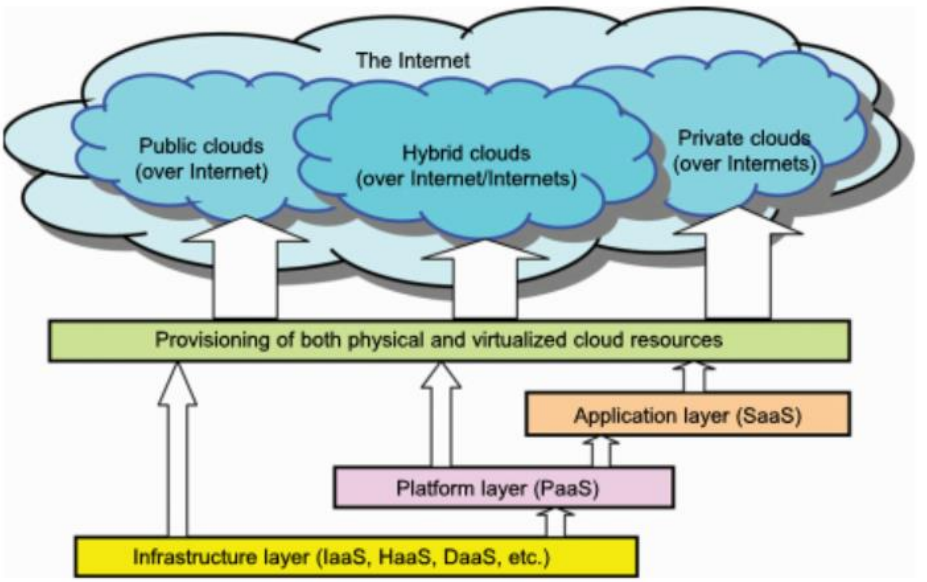


FIGURE: Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

- The application layer is formed with a collection of all needed software modules for SaaS applications. Service applications in this layer include daily office management work, such as information retrieval, document processing, and calendar and authentication services.
- The application layer is also heavily used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transactions, and supply chain management.

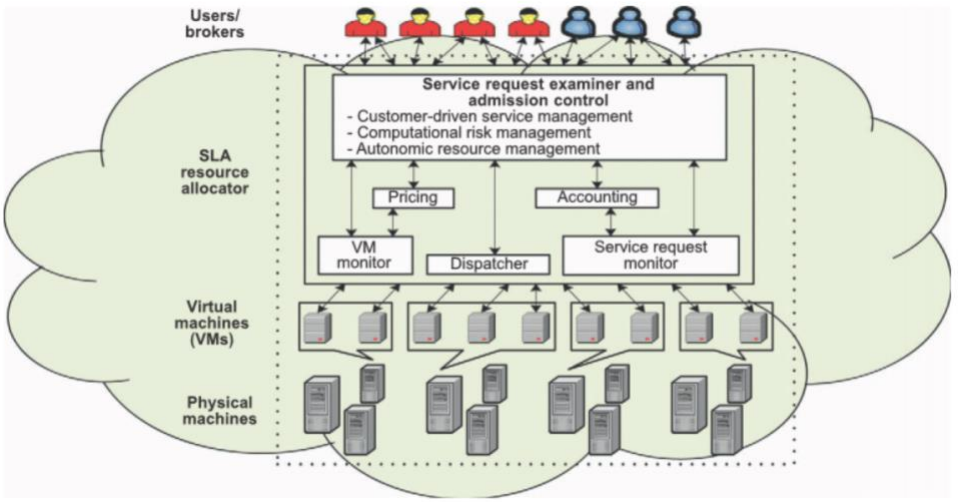
- In general, SaaS demands the most work from the provider, PaaS is in the middle, and IaaS demands the least.
- For example, Amazon EC2 provides not only virtualized CPU resources to users, but also management of these provisioned resources. Services at the application layer demand more work from providers. The best example of this is the Salesforce.com CRM service, in which the provider supplies not only the hardware at the bottom layer and the software at the top layer, but also the platform and software tools for user application development and monitoring.

3.1.1 Market-Oriented Cloud Architecture

- Cloud providers consider and meet the different QoS parameters of each individual consumer as negotiated in specific SLAs. To achieve this, the providers cannot deploy traditional system-centric resource management architecture. Instead, market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand.
- The designer needs to provide feedback on economic incentives for both consumers and providers. The purpose is to promote QoS-based resource allocation mechanisms. In addition, clients can benefit from the potential cost reduction of providers, which could lead to a more competitive market, and thus lower prices.
- Figure shows the high-level architecture for supporting market-oriented resource allocation in a cloud computing environment.
- This cloud is basically built with the following entities: Users or brokers acting on user's behalf submit service requests from anywhere in the world to the data center and cloud to be processed. The SLA resource allocator acts as the interface between the data center/cloud service provider and external users/brokers. It requires the interaction of the following mechanisms to support SLA-oriented resource management. When a service request is first submitted the service request examiner interprets the submitted request for QoS requirements before determining

whether to accept or reject the request.

- The request examiner ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources. It also needs the latest status information regarding resource availability (from the VM Monitor mechanism) and workload processing (from the Service Request Monitor mechanism) in order to make resource allocation decisions effectively.
- Then it assigns requests to VMs and determines resource entitlements for allocated VMs. The Pricing mechanism decides how service requests are charged.
- For instance, requests can be charged based on submission time (peak/off-peak), pricing rates (fixed/changing), or availability of resources (supply/ demand). Pricing serves as a basis for managing the supply and demand of computing resources within the data center and facilitates in prioritizing resource allocations effectively.



F IGURE: Market-oriented cloud architecture to expand/shrink leasing of resources with variation in QoS/demand from users.

- The Accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to users. In

addition, the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.

- The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements. The Dispatcher mechanism starts the execution of accepted service requests on allocated VMs. The Service Request Monitor mechanism keeps track of the execution progress of service requests.
- Multiple VMs can be started and stopped on demand on a single physical machine to meet accepted service requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests.
- In addition, multiple VMs can concurrently run applications based on different operating system environments on a single physical machine since the VMs are isolated from one another on the same physical machine.

3.1.2 Quality of Service Factors

- The data center comprises multiple computing servers that provide resources to meet service demands. In the case of a cloud as a commercial offering to enable crucial business operations of companies, there are critical QoS parameters to consider in a service request, such as time, cost, reliability, and trust/security.
- In short, there should be greater importance on customers since they pay to access services in clouds. In addition, the state of the art in cloud computing has no or limited support for dynamic negotiation of SLAs between participants and mechanisms for automatic allocation of resources to multiple competing requests. Negotiation mechanisms are needed to respond to alternate offers protocol for establishing SLAs.
- Commercial cloud offerings must be able to support customer-driven service management based on customer profiles and requested service requirements. Commercial clouds define computational risk management

tactics to identify, assess, and manage risks involved in the execution of applications with regard to service requirements and customer needs.

- The cloud also derives appropriate market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA-oriented resource allocation.
- The system incorporates autonomic resource management models that effectively self-manage changes in service requirements to satisfy both new service demands and existing service obligations, and leverage VM technology to dynamically assign resource shares according to service requirements.

3.2 NIST CLOUD COMPUTING REFERENCE ARCHITECTURE

3.2.1 The Conceptual Reference Model

- ☐ The National Institute of Standards and Technologies (NIST) is releasing its first guidelines for agencies that want to use cloud computing in the second half of 2009 National Institute of Standards and Technologies
- ☐ Figure presents an overview of the NIST cloud computing reference architecture, which identifies the major actors, their activities and functions in cloud computing.
- ☐ The diagram depicts a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

The Conceptual Reference Model Figure 1 presents an overview of the NIST cloud computing reference architecture, which identifies the major actors, their activities and functions in cloud computing. The diagram depicts a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

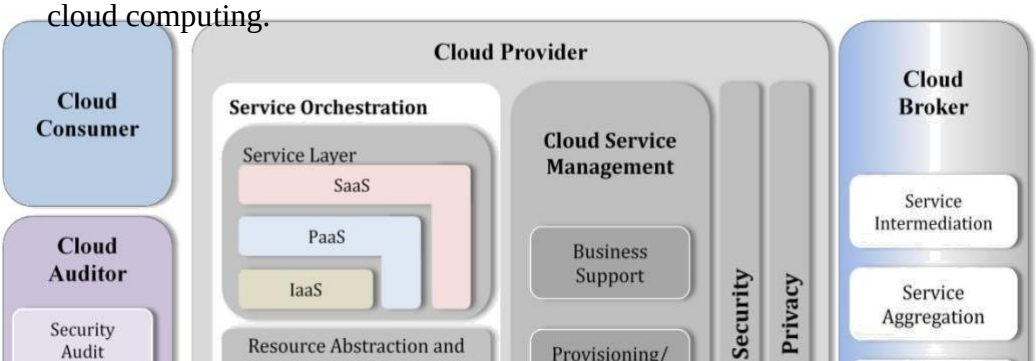


Figure: The Conceptual Reference Model

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

- Table briefly lists the actors defined in the NIST cloud computing reference architecture.

Actors in Cloud Computing

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, *Cloud Providers*.
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between *Cloud Providers* and *Cloud Consumers.*
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from *Cloud Providers* to *Cloud Consumers*.

- Figure illustrates the interactions among the actors. A cloud consumer may request cloud services from a cloud provider directly or via a cloud broker.
- A cloud auditor conducts independent audits and may contact the others to collect necessary information.

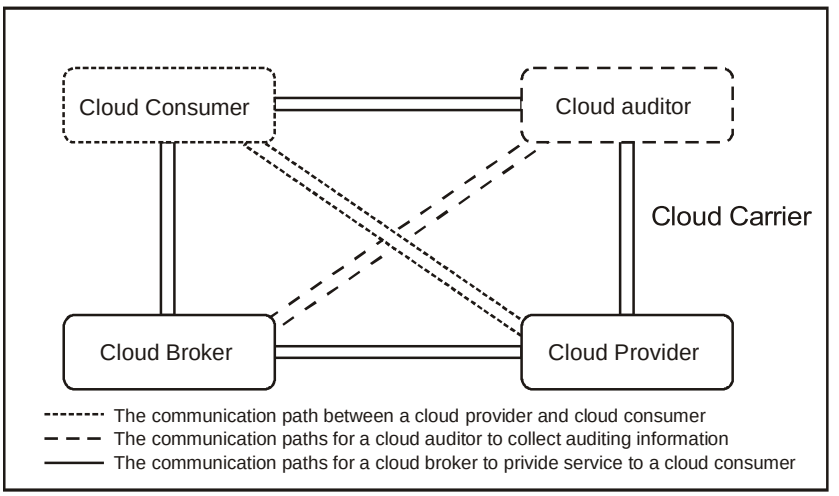


Figure : Interactions between the Actors in Cloud Computing

Example Usage Scenario 1:

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

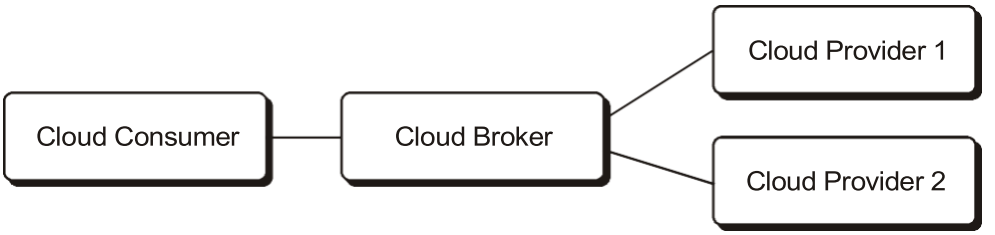


Figure: Usage Scenario for Cloud Brokers

Example Usage Scenario 2:

Cloud carriers provide the connectivity and transport of cloud services from cloud providers to cloud consumers. As illustrated in Figure 3.6, a cloud provider participates in and arranges for two unique service level agreements (SLAs), one with a cloud carrier (e.g. SLA2) and one with a cloud consumer (e.g. SLA1). A cloud provider arranges service level agreements (SLAs) with a cloud carrier and may request dedicated and encrypted connections to ensure the cloud services are consumed at a consistent level according to the contractual obligations with the cloud consumers. In this case, the provider may specify its requirements on capability, flexibility and functionality in SLA2 in order to provide essential requirements in SLA1.

Figure : Usage Scenario for Cloud Carriers

Example Usage Scenario 3:

For a cloud service, a cloud auditor conducts independent assessments of the operation and security of the cloud service implementation. The audit may involve interactions with both the Cloud Consumer and the Cloud Provider.

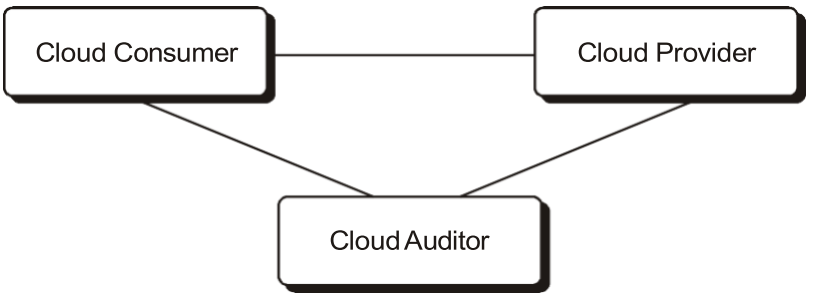


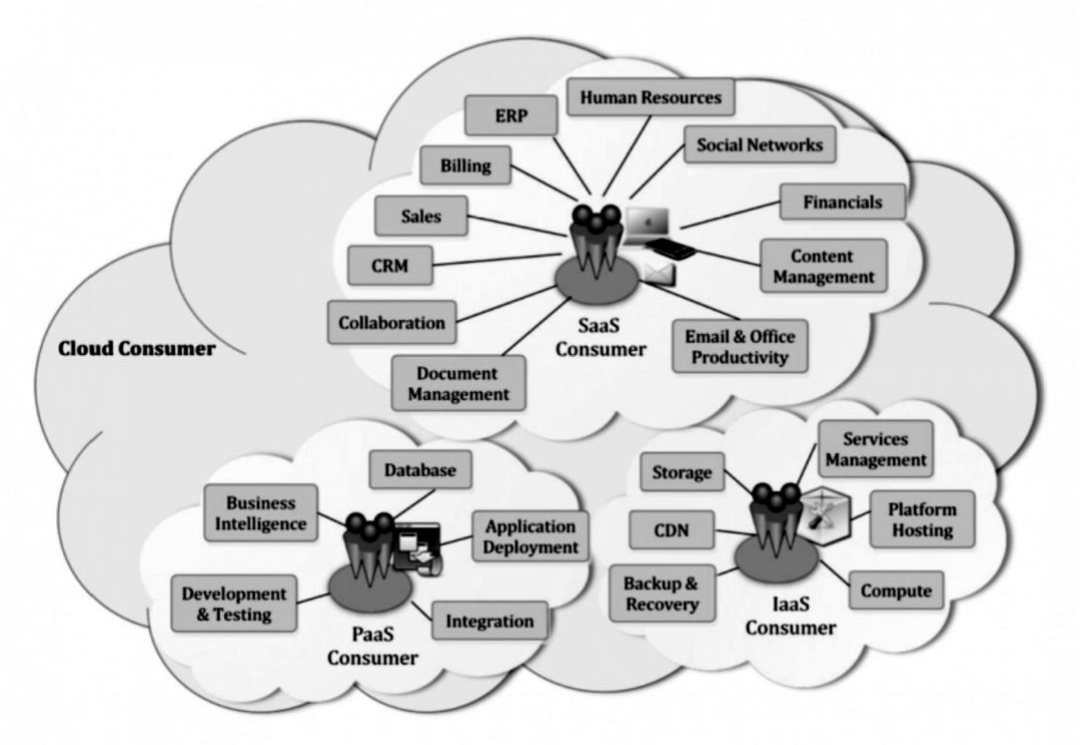
Figure: Usage Scenario for Cloud Auditors

3.2.2 Cloud Consumer

- The cloud consumer is the principal stakeholder for the cloud computing service. A cloud consumer represents a person or organization that maintains a business relationship with, and uses the service from a cloud provider.
- A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider, and uses the service. The cloud consumer may be billed for the service provisioned, and needs to arrange payments accordingly.
- Cloud consumers need SLAs to specify the technical performance requirements fulfilled by a cloud provider. SLAs can cover terms regarding the quality of service, security, remedies for performance failures. A cloud provider may also list in the SLAs a set of promises explicitly not made to consumers, i.e. limitations, and obligations that cloud consumers must accept.

- A cloud consumer can freely choose a cloud provider with better pricing and more favorable terms. Typically a cloud provider s pricing policy and SLAs are non-negotiable, unless the customer expects heavy usage and might be able to negotiate for better contracts.
- Depending on the services requested, the activities and usage scenarios can be different among cloud consumers.
- Figure presents some example cloud services available to a cloud consumer SaaS applications in the cloud and made accessible via a network to the SaaS consumers.
- The consumers of SaaS can be organizations that provide their members with access to software applications, end users who directly use software applications, or software application administrators who configure applications for end users. SaaS consumers can be billed based on the number of end users, the time of use, the network bandwidth consumed, the amount of data stored or duration of stored data.

Figure: Example Services Available to a Cloud Consumer



- Cloud consumers of PaaS can employ the tools and execution resources provided by cloud providers to develop, test, deploy and manage the applications hosted in a cloud environment.
- PaaS consumers can be application developers who design and implement application software, application testers who run and test applications in cloud-based environments, application deployers who publish applications into the cloud, and application administrators who configure and monitor application performance on a platform.
- PaaS consumers can be billed according to, processing, database storage and network resources consumed by the PaaS application, and the duration of the platform usage.
- Consumers of IaaS have access to virtual computers, network-accessible storage, network infrastructure components, and other fundamental computing resources on which they can deploy and run arbitrary software.
- The consumers of IaaS can be system developers, system administrators and IT managers who are interested in creating, installing, managing and monitoring services for IT infrastructure operations. IaaS consumers are provisioned with the capabilities to access these computing resources, and are billed according to the amount or duration of the resources consumed, such as CPU hours used by virtual computers, volume and duration of data stored, network bandwidth consumed, number of IP addresses used for certain intervals.

3.2.3 Cloud Provider

- A cloud provider is a person, an organization; it is the entity responsible for making a service available to interested parties.
- A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access.
- For Software as a Service, the cloud provider deploys, configures,

maintains and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers.

- The provider of SaaS assumes most of the responsibilities in managing and controlling the applications and the infrastructure, while the cloud consumers have limited administrative control of the applications.
- For PaaS, the Cloud Provider manages the computing infrastructure for the platform and runs the cloud software that provides the components of the platform, such as runtime software execution stack, databases, and other middleware components.
- The PaaS Cloud Provider typically also supports the development, deployment and management process of the PaaS Cloud Consumer by providing tools such as integrated development environments (IDEs), development version of cloud software, software development kits (SDKs), deployment and management tools.
- The PaaS Cloud Consumer has control over the applications and possibly some the hosting environment settings, but has no or limited access to the infrastructure underlying the platform such as network, servers, operating systems (OS), or storage.
- For IaaS, the Cloud Provider acquires the physical computing resources underlying the service, including the servers, networks, storage and hosting infrastructure.
- The Cloud Provider runs the cloud software necessary to makes computing resources available to the IaaS Cloud Consumer through a set of service interfaces and computing resource abstractions, such as virtual machines and virtual network interfaces.
- The IaaS Cloud Consumer in turn uses these computing resources, such as a virtual computer, for their fundamental computing needs Compared to SaaS and PaaS Cloud Consumers, an IaaS Cloud Consumer has access to more fundamental forms of computing resources and thus has more control over the more software components in an application stack,

including the OS and network.

- The IaaS Cloud Provider, on the other hand, has control over the physical hardware and cloud software that makes the provisioning of these infrastructure services possible, for example, the physical servers, network equipments, storage devices, host OS and hypervisors for virtualization.
- A Cloud Provider s activities can be described in five major areas, as shown in Figure 3.9, a cloud provider conducts its activities in the areas of *service deployment*, *service orchestration*, *cloud service management*, *security,* and *privacy*.

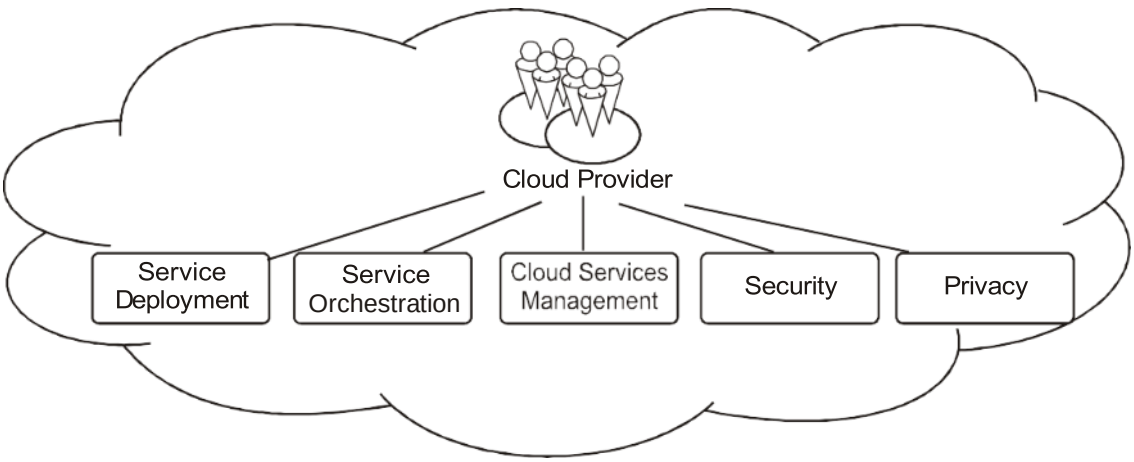


Figure: Cloud Provider - Major Activities

3.2.4 Cloud Auditor

- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon.
- Audits are performed to verify conformance to standards through review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc.
- Auditing is especially important for federal agencies as "agencies should include a contractual clause enabling third parties to assess security controls of cloud providers" (*by Vivek Kundra, Federal Cloud Computing*

Strategy, Feb. 2011.).

- Security controls are the management, operational, and technical safeguards or countermeasures employed within an organizational information system to protect the confidentiality, integrity, and availability of the system and its information.
- For security auditing, a cloud auditor can make an assessment of the security controls in the information system to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to the security requirements for the system.
- The security auditing should also include the verification of the compliance with regulation and security policy. For example, an auditor can be tasked with ensuring that the correct policies are applied to data retention according to relevant rules for the jurisdiction. The auditor may ensure that fixed content has not been modified and that the legal and business data archival requirements have been satisfied.
- A privacy impact audit can help Federal agencies comply with applicable privacy laws and regulations governing an individual s privacy, and to ensure confidentiality, integrity, and availability of an individual s personal information at every stage of development and operation.

3.2.5 Cloud Broker

- A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly. A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers.
- In general, a cloud broker can provide services in three categories:
  - ***Service Intermediation*:** A cloud broker enhances a given service by improving some specific capability and providing value-added services to cloud consumers. The improvement can be managing access to cloud services, identity management, performance reporting, enhanced security, etc.

- ***Service Aggregation*:** A cloud broker combines and integrates multiple services into one or more new services. The broker provides data integration and ensures the secure data movement between the cloud consumer and multiple cloud providers.
- ***Service Arbitrage*:** Service arbitrage is similar to service aggregation except that the services being aggregated are not fixed. Service arbitrage means a broker has the flexibility to choose services from multiple agencies. The cloud broker, for example, can use a credit-scoring service to measure and select an agency with the best score.

3.2.6 Cloud Carrier

- A cloud carrier acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers. Cloud carriers provide access to consumers through network, telecommunication and other access devices.
- For example, cloud consumers can obtain cloud services through network access devices, such as computers, laptops, mobile phones, mobile Internet devices (MIDs), etc.
- The distribution of cloud services is normally provided by network and telecommunication carriers or a *transport agent* , where a transport agent refers to a business organization that provides physical transport of storage media such as high-capacity hard drives.
- Note that a cloud provider will set up SLAs with a cloud carrier to provide services consistent with the level of SLAs offered to cloud consumers, and may require the cloud carrier to provide dedicated and secure connections between cloud consumers and cloud providers.

3.2.7 Scope of Control between Provider and Consumer

- The Cloud Provider and Cloud Consumer share the control of resources in a cloud system. As illustrated in Figure, different service models affect an

organization s control over the computational resources and thus what can be done in a cloud system.

- The figure shows these differences using a classic software stack notation comprised of the application, middleware, and OS layers.
- This analysis of delineation of controls over the application stack helps understand the responsibilities of parties involved in managing the cloud application.

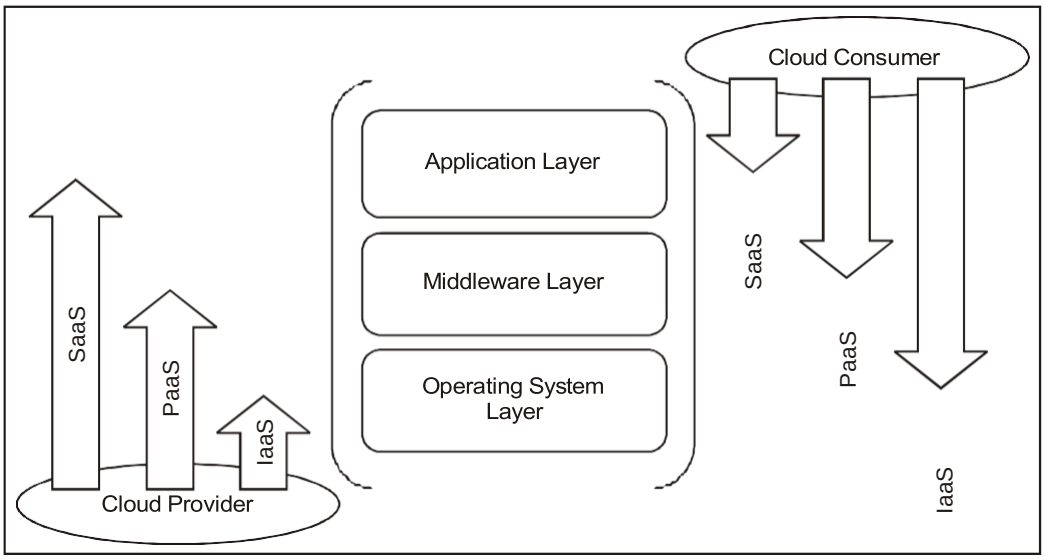


Figure: Scope of Controls between Provider and Consumer

- The application layer includes software applications targeted at end users or programs. The applications are used by SaaS consumers, or installed/ managed/ maintained by PaaS consumers, IaaS consumers, and SaaS providers.
- The middleware layer provides software building blocks (e.g., libraries, database, and Java virtual machine) for developing application software in the cloud. The middleware is used by PaaS consumers, installed/managed/ maintained by IaaS consumers or PaaS providers, and hidden from SaaS consumers.
- The OS layer includes operating system and drivers, and is hidden from

SaaS consumers and PaaS consumers.

- An IaaS cloud allows one or multiple guest OS s to run virtualized on a single physical host.
- Generally, consumers have broad freedom to choose which OS to be hosted among all the OS s that could be supported by the cloud provider.
- The IaaS consumers should assume full responsibility for the guest OS s, while the IaaS provider controls the host OS.

CLOUD COMPUTING REFERENCE ARCHITECTURE: ARCHITECTURAL COMPONENTS

3.3.1 Service Deployment

- A cloud infrastructure may be operated in one of the following deployment models: public cloud, private cloud, community cloud, or hybrid cloud. The differences are based on how exclusive the computing resources are made to a Cloud Consumer.

Public Cloud

- A *public cloud* is built over the Internet and can be accessed by any user who has paid for the service. Public clouds are owned by service providers and are accessible through a subscription.
- The callout box in top of Figure shows the architecture of a typical public cloud.

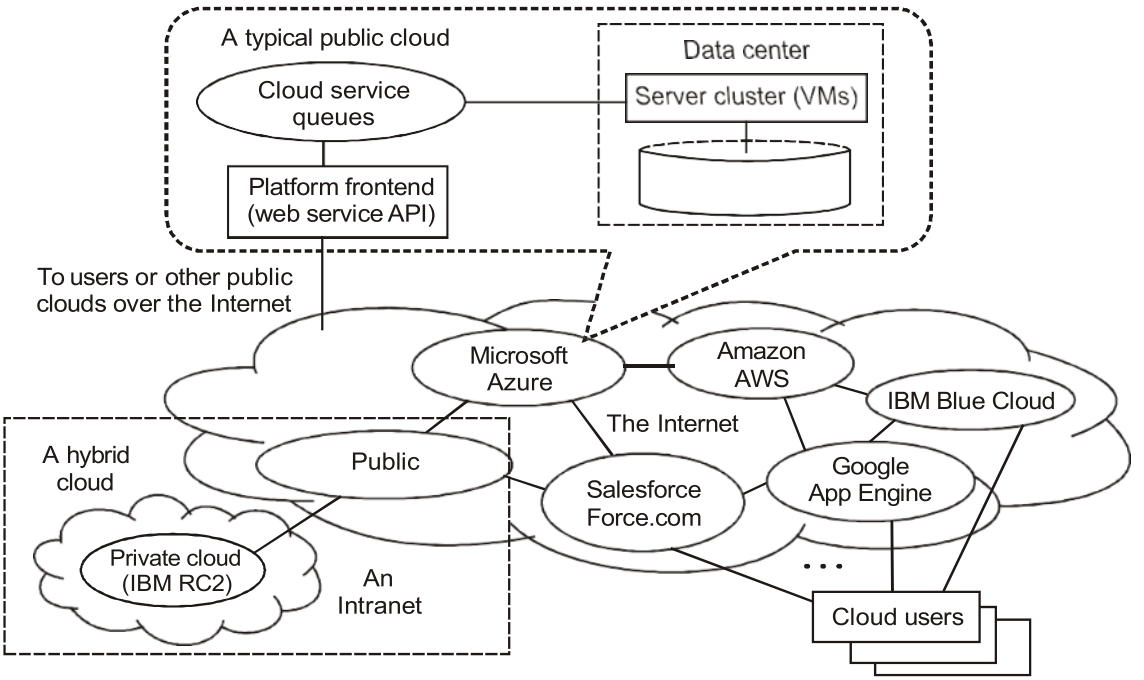


FIGURE Public, private, and hybrid clouds illustrated by functional architecture and connectivity of representative clouds available by 2011.

- Many public clouds are available, including Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com's Force.com. The providers of the aforementioned clouds are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.
- A public cloud delivers a selected set of business processes.
- The application and infrastructure services are offered on a flexible price-peruse basis.
- A public cloud is one in which the cloud infrastructure and computing resources are made available to the general public over a public network.

The NIST definition "Cloud computing"

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

- The NIST definition also identifies
    - 5 essential characteristics
    - 3 service models
    - 4 deployment models

4 deployment models / Types of cloud

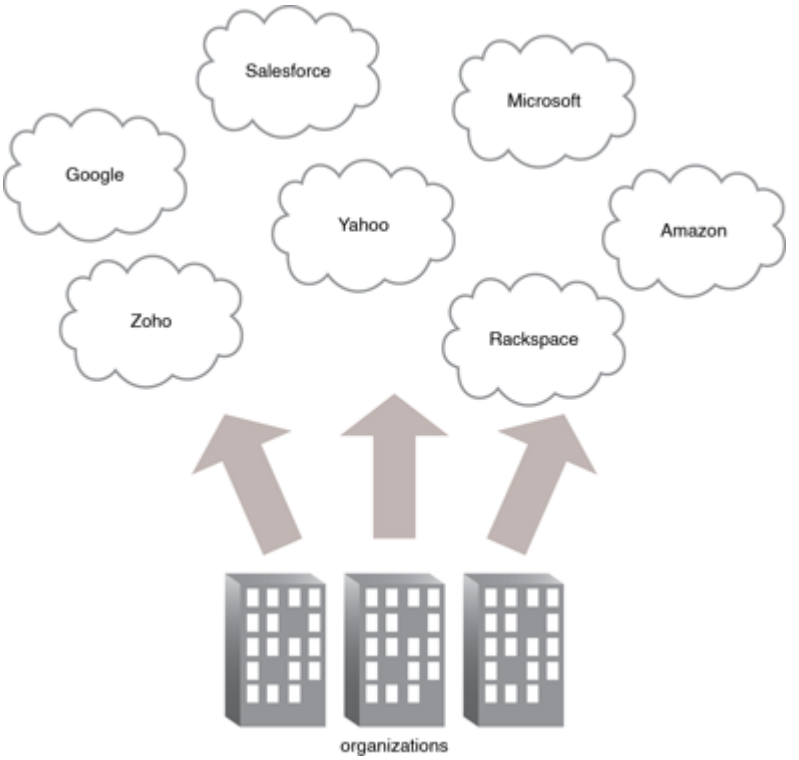
- **Public**: Accessible, via the Internet, to anyone who pays

– Owned by service providers; e.g., Google App Engine, Amazon Web Services, Force.com.

A *public cloud* is a publicly accessible cloud environment owned by a third-party cloud provider. The IT resources on public clouds are usually provisioned via the previously described cloud delivery models and are generally offered to cloud consumers at a cost or are commercialized via other avenues (such as advertisement).

The cloud provider is responsible for the creation and on-going maintenance of the public cloud and its IT resources. Many of the scenarios and architectures explored in upcoming chapters involve public clouds and the relationship between the providers and consumers of IT resources via public clouds.

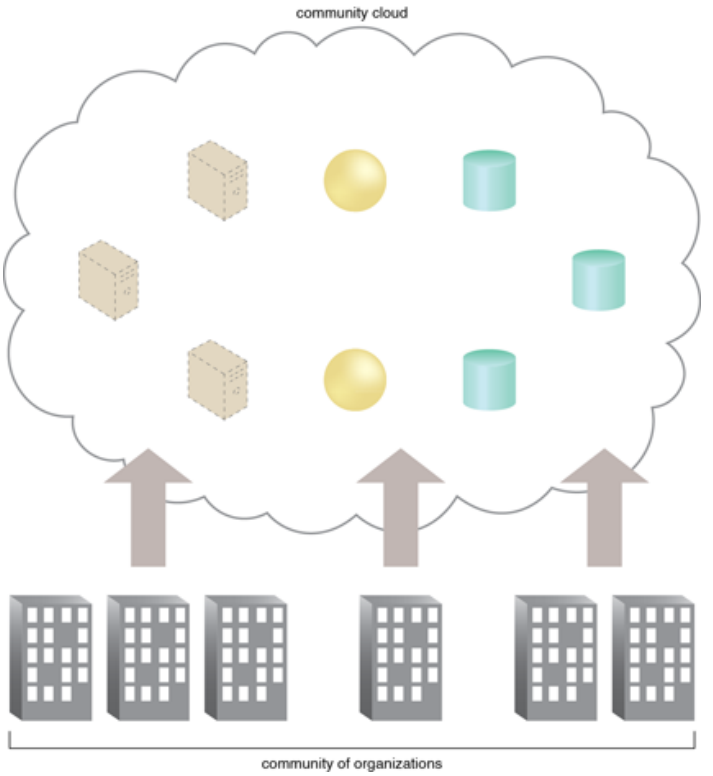
Figure 1 shows a partial view of the public cloud landscape, highlighting some of the primary vendors in the marketplace.



- **Community**: Shared by two or more organizations with joint interests, such as colleges within a university

  A community cloud is similar to a public cloud except that its access is limited to a specific community of cloud consumers. The community cloud may be jointly owned by the community members or by a third-party cloud provider that provisions a public cloud with limited access. The member cloud consumers of the community typically share the responsibility for defining and evolving the community cloud (Figure 1).

  Membership in the community does not necessarily guarantee access to or control of all the cloud's IT resources. Parties outside the community are generally not granted access unless allowed by the community.



An example of a "community" of organizations accessing IT resources from a community cloud.

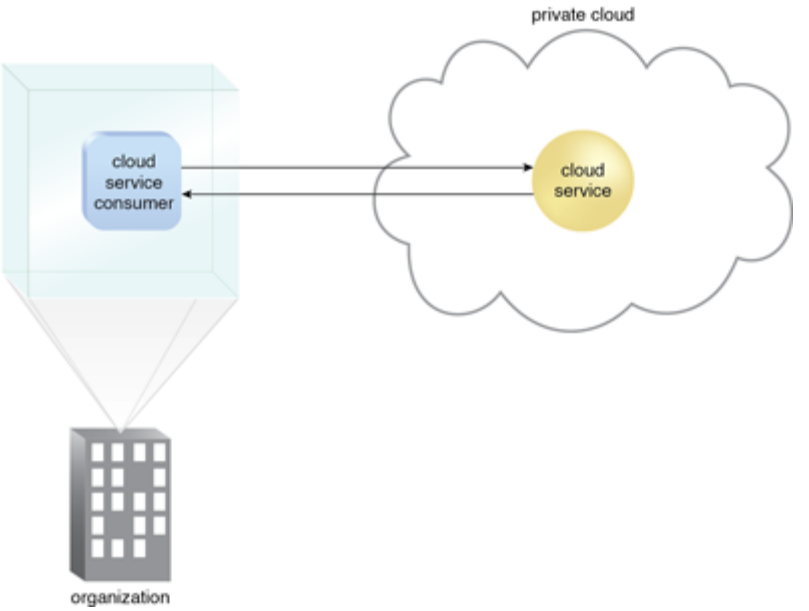
- **Private**: Accessible via an intranet to the members of the owning organization

– Can be built using open source software such as CloudStack or OpenStack

– Example of private cloud: NASA's cloud for climate modeling

A private cloud is owned by a single organization. Private clouds enable an organization to use cloud computing technology as a means of centralizing access to IT resources by different parts, locations, or departments of the organization. When a private cloud exists as a controlled environment, the problems described in the Risks and Challenges section do not tend to apply. The use of a private cloud can change how organizational and trust

boundaries are defined and applied. The actual administration of a private cloud environment may be carried out by internal or outsourced staff.



A cloud service consumer in the organization's on-premise environment accesses a cloud service hosted on the same organization's private cloud via a virtual private network.

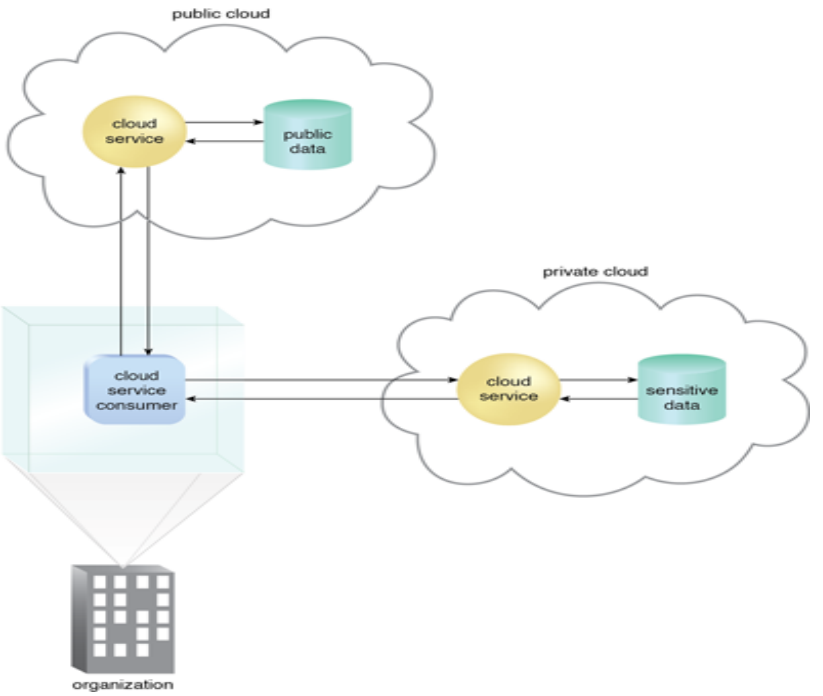
With a private cloud, the same organization is technically both the cloud consumer and cloud provider . In order to differentiate these roles:

- a separate organizational department typically assumes the responsibility for provisioning the cloud (and therefore assumes the cloud provider role)
- departments requiring access to the private cloud assume the cloud consumer role

Hybrid

A hybrid cloud is a cloud environment comprised of two or more different cloud deployment models. For example, a cloud consumer may choose to

deploy cloud services processing sensitive data to a private cloud and other, less sensitive cloud services to a public cloud. The result of this combination is a hybrid deployment model.



An organization using a hybrid cloud architecture that utilizes both a private and public cloud.

Hybrid deployment architectures can be complex and challenging to create and maintain due to the potential disparity in cloud environments and the fact that management responsibilities are typically split between the private cloud provider organization and the public cloud provider.

Three service models / Categories of cloud computing.

- Software as a Service (SaaS)
  - Use provider's applications over a network

- Platform as a Service (PaaS)
  - Deploy customer-created applications to a cloud
- Infrastructure as a Service (IaaS)

- *Software as a Service (SaaS).* The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of provider-defined user-specific application configuration settings.
- *Platform as a Service (PaaS).* The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.
- *Infrastructure as a Service (IaaS).* The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control

the underlying cloud physical infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components.

Comparison of cloud service models

Type	Consumer	Service Provided By Cloud	Service Level Coverage	Customization
SaaS	End user	• Finished application	• Application uptime • Application Performance	• Minimal to no customization • Capabilities dictated by market or provider
PaaS	Application owner	• Runtime environment for application code • Cloud storage • Other Cloud services such as integration	• Environment availability • Environment performance • No application coverage	• High degree of application level customization available within constraints of the service offered • Many applications will need to be rewritten
IaaS	Application owner or IT provides OS, middleware and application support	• Virtual server • Cloud storage	• Virtual server availability • Time to provision • No platform or application coverage	• Minimal constraints on applications installed on standardized virtual OS builds

3.3.2 Service Orchestration

- *Service Orchestration* refers to the composition of system components to support the Cloud Providers activities in arrangement, coordination and management of computing resources in order to provide cloud services to Cloud Consumers.

3.4 INFRASTRUCTURE-AS-A-SERVICE (IAAS)

- Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers.
- The services provided over the cloud can be generally categorized into three different models: namely IaaS, Platform as a Service (PaaS), and Software as a Service (SaaS). All three models allow users to access services over the Internet, relying entirely on the infrastructures of cloud service providers.
- These models are offered based on various SLAs between providers and users. In a broad sense, the SLA for cloud computing is addressed in terms of service availability, performance, and data protection and security.
- Figure illustrates three cloud models at different service levels of the cloud. SaaS is applied at the application end using special interfaces by users or clients.

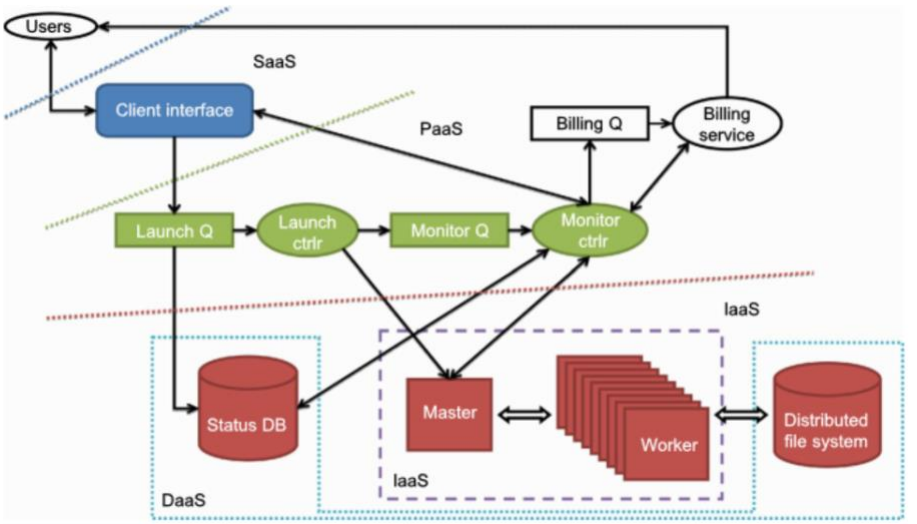


FIGURE: The IaaS, PaaS, and SaaS cloud service models at different service levels.

- At the PaaS layer, the cloud platform must perform billing services and handle job queuing, launching, and monitoring services. At the bottom layer of the IaaS services, databases, compute instances, the file system, and storage must be provisioned to satisfy user demands.

3.4.1 Infrastructure as a Service

- This model allows users to use virtualized IT resources for computing, storage, and networking. In short, the service is performed by rented cloud infrastructure. The user can deploy and run his applications over his chosen OS environment.
- The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components. This IaaS model encompasses *storage as a service*, compute *instances as a service*, and *communication as a service*.
- The *Virtual Private Cloud (VPC)* in Example shows how to provide Amazon EC2 clusters and S3 storage to multiple users. Many startup cloud providers have appeared in recent years. GoGrid, FlexiScale, and Aneka are good examples. Table summarizes the IaaS offerings by five public cloud providers. Interested readers can visit the companies' web sites for updated information.

Public Cloud Offerings of IaaS

Cloud Name	**VM Instance Capacity**	**API and Access Tools**	**Hypervisor, Guest OS**
Amazon EC2	Each instance has 1 –20 EC2 processors, 1.7–15 GB of memory, and 160-1.69 TB of storage.	CLI or web Service (WS) portal	Xen, Linux, Windows
GoGrid	Each instance has 1–6 CPUs, 0.5–8 GB of memory, and 30-480 GB of storage.	REST. Java. PHP, Python, Ruby	Xen, Linux, Windows

Rackspace Cloud	Each instance has a four-core CPU, 0.25–16 GB of memory, and 10-620 GB of storage.	REST, Python, PHP, Java, C#, .NET	Xen, Linux
Flexi Scale in the UK	Each instance has 1–4 CPUs, 0.5–16 GB of memory, and 20-270 GB of storage.	web console	Xen, Linux, Windows
Joyent Cloud	Each instance has up to eight CPUs, 0.25–32 GB of memory, and 30-480 GB of storage.	No specific API, SSH, Virtual/Min	OS-level virtualization, Open Solaris

Example

Amazon VPC for Multiple Tenants

- A user can use a private facility for basic computations. When he must meet a specific workload requirement, he can use the Amazon VPC to provide additional EC2 instances or more storage (S3) to handle urgent applications.
- Figure shows VPC which is essentially a private cloud designed to address the privacy concerns of public clouds that hamper their application when sensitive data and software are involved.

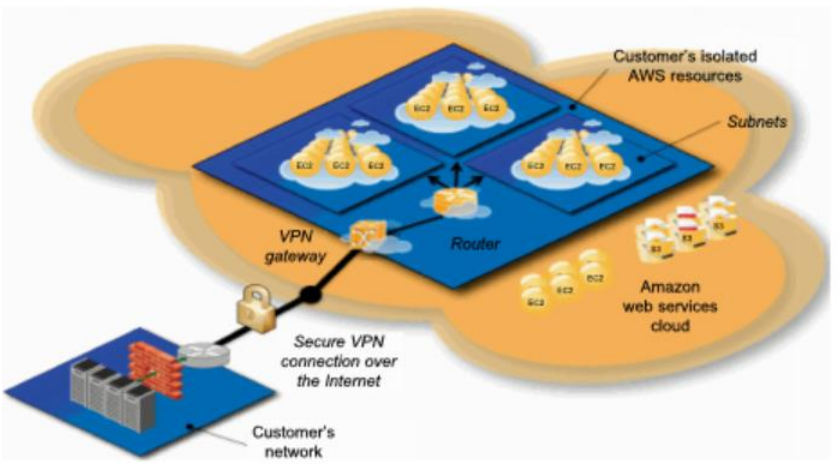


FIGURE Amazon VPC (virtual private cloud) Courtesy of VMWare, http://aws.amazon.com/vpc/

- Amazon EC2 provides the following services: resources from multiple data centers globally distributed, CL1, web services (SOAP and Query), web- based console user interfaces, access to VM instances via SSH and Windows,
  99.5 percent available agreements, per-hour pricing, Linux and Windows OSes, and automatic scaling and load balancing.

- VPC allows the user to isolate provisioned AWS processors, memory, and storage from interference by other users. Both *autoscaling* and *elastic load balancing* services can support related demands. Autoscaling enables users to automatically scale their VM instance capacity up or down. With auto- scaling, one can ensure that a sufficient number of Amazon EC2 instances are provisioned to meet desired performance. Or one can scale down the VM instance capacity to reduce costs, when the workload is reduced.

3.5 PLATFORM-AS-A-SERVICE (PAAS) AND SOFTWARE-AS- A-SERVICE (SAAS)

SaaS is often built on top of the PaaS, which is in turn built on top of the IaaS.

3.5.1 Platform as a Service (PaaS)

- To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment. Such a platform includes operating system and runtime library support.

- This has triggered the creation of the PaaS model to enable users to develop and deploy their user applications. Table highlights cloud platform services offered by five PaaS services.

Five Public Cloud Offerings of PaaS

Cloud Name	Languages and Developer Tools	Programming Models Supported by Provider	Target Applications and Storage Option
Google App Engine	Python, Java, and Eclipse-based IDE	MapReduce, web programming on demand	Web applications and BigTable storage
Salesforce.com's Force.com	Apex, Eclipse-based IDE, web-based Wizard	Workflow, Excel-like formula, Web programming on demand	Business applications such as CRM
Microsoft Azure	.NET, Azure tools for MS Visual Studio	Unrestricted model	Enterprise and web applications
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++	MapReduce	Data processing and e-commerce
Aneka	.NET, stand-alone SDK	Threads, task, MapReduce	.NET enterprise applications, HPC

- The platform cloud is an integrated computer system consisting of both hardware and software infrastructure. The user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET).
- The user does not manage the underlying cloud infrastructure. The cloud provider supports user application development and testing on a well-defined service platform. This PaaS model enables a collaborated software development platform for users from different parts of the world. This model also encourages third parties to provide software management, integration, and service monitoring solutions.

Example

Google App Engine for PaaS Applications

As web applications are running on Google's server clusters, they share the same capability with many other users. The applications have features such as automatic scaling and load balancing which are very convenient

while building web applications. The distributed scheduler mechanism can also schedule tasks for triggering events at specified times and regular intervals.

Figure shows the operational model for GAE. To develop applications using GAE, a development environment must be provided.

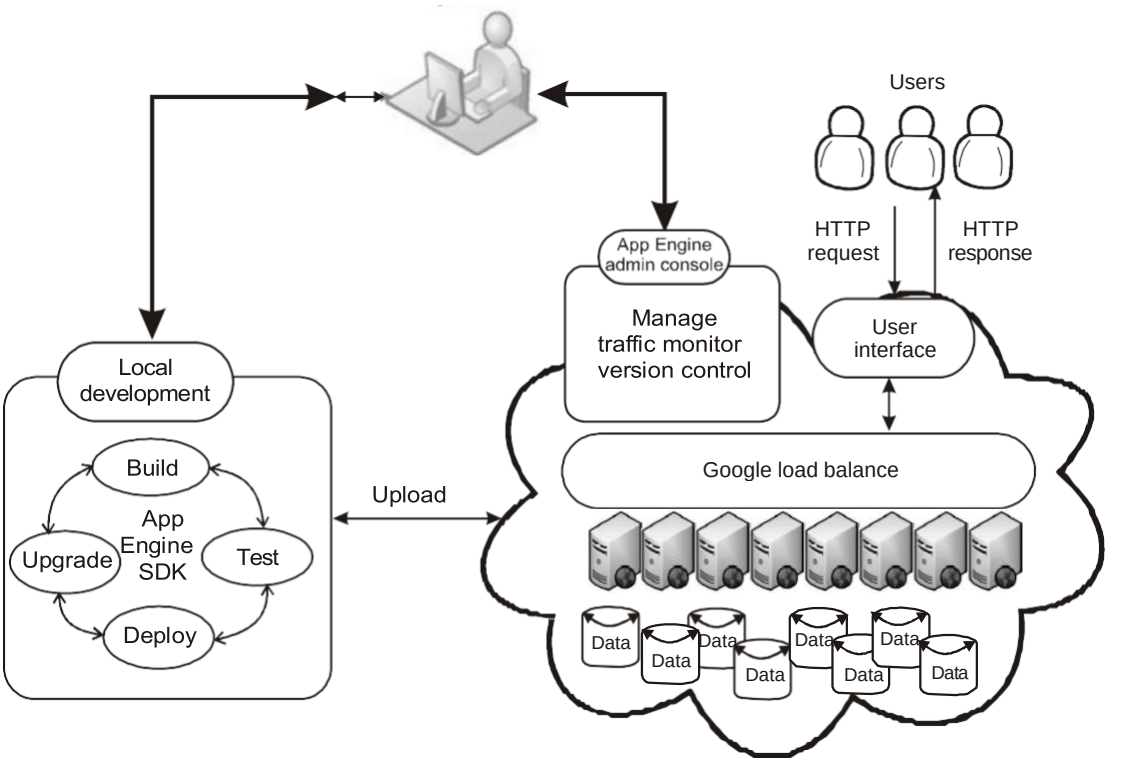


FIGURE Google App Engine platform for PaaS operations

Google provides a fully featured local development environment that simulates GAE on the developer's computer. All the functions and application logic can be implemented locally which is quite similar to traditional software development. The coding and debugging stages can be performed locally as well. After these steps are finished, the SDK provided provides a tool for uploading the user's application to Google's infrastructure where the applications are actually deployed. Many additional

third-party capabilities, including software management, integration, and service monitoring solutions, are also provided.

3.5.2 Software as a Service (SaaS)

- This refers to browser-initiated application software over thousands of cloud customers. Services and tools offered by PaaS are utilized in construction of applications and management of their deployment on resources offered by IaaS providers. The SaaS model provides software applications as a service.
- As a result, on the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are kept rather low, compared with conventional hosting of user applications.
- Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support PaaS and IaaS. The best examples of SaaS services include Google Gmail and docs, Microsoft SharePoint, and the CRM software from Salesforce.com. They are all very successful in promoting their own business or are used by thousands of small businesses in their dayto-day operations.
- Providers such as Google and Microsoft offer integrated IaaS and PaaS services, whereas others such as Amazon and GoGrid offer pure IaaS services and expect third-party PaaS providers such as Manjrasoft to offer application development and deployment services on top of their infrastructure services. To identify important cloud applications in enterprises, the success stories of three real-life cloud applications are presented in Example 3.6 for HTC, news media, and business transactions. The benefits of using cloud services are evident in these SaaS applications.

Example

Three Success Stories on SaaS Applications

1. To discover new drugs through DNA sequence analysis, Eli Lily Company has used Amazon's AWS platform with provisioned server and storage clusters to conduct high-performance biological sequence analysis

without using an expensive supercomputer. The benefit of this IaaS application is reduced drug deployment time with much lower costs.

2. The *New York Times* has applied Amazon's EC2 and S3 services to retrieve useful pictorial information quickly from millions of archival articles and newspapers. The *New York Times* has significantly reduced the time and cost in getting the job done.

3. Pitney Bowes, an e-commerce company, offers clients the opportunity to perform B2B transactions using the Microsoft Azure platform, along with .NET and SQL services. These offerings have significantly increased the company's client base.

3.5.3 Mashup of Cloud Services

- At the time of this writing, public clouds are in use by a growing number of users. Due to the lack of trust in leaking sensitive data in the business world, more and more enterprises, organizations, and communities are developing private clouds that demand deep customization.
- An enterprise cloud is used by multiple users within an organization. Each user may build some strategic applications on the cloud, and demands customized partitioning of the data, logic, and database in the metadata representation. More private clouds may appear in the future.
- Based on a 2010 Google search survey, interest in grid computing is declining rapidly. *Cloud mashups* have resulted from the need to use multiple clouds simultaneously or in sequence.
- For example, an industrial supply chain may involve the use of different cloud resources or services at different stages of the chain. Some public repository provides thousands of service APIs and mashups for web commerce services. Popular APIs are provided by Google Maps, Twitter, YouTube, Amazon eCommerce, Salesforce.com, etc.

3.6 ARCHITECTURAL DESIGN CHALLENGES

Six open challenges in cloud architecture development

1. Service Availability and Data Lock-in Problem
2. Data Privacy and Security Concerns
3. Unpredictable Performance and Bottlenecks
4. Distributed Storage and Widespread Software Bugs
5. Cloud Scalability, Interoperability, and Standardization
6. Software Licensing and Reputation Sharing

3.6.1 Challenge1—Service Availability and Data Lock-in Problem

- The management of a cloud service by a single company is often the source of single points of failure. To achieve HA, one can consider using multiple cloud providers. Even if a company has multiple data centers located in different geographic regions, it may have common software infrastructure and accounting systems. Therefore, using multiple cloud providers may provide more protection from failures.
- Another availability obstacle is distributed denial of service (DDoS) attacks. Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable. Some utility computing services offer SaaS providers the opportunity to defend against DDoS attacks by using quick scale-ups.
- Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary. Thus, customers cannot easily extract their data and programs from one site to run on another.
- The obvious solution is to standardize the APIs so that a SaaS developer can deploy services and data across multiple cloud providers. This will rescue the loss of all data due to the failure of a single company.
- In addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the same software infrastructure can be used in both public and private clouds. Such an option could enable "surge computing," in which the public cloud is used to capture the extra tasks that cannot

be easily run in the data center of a private cloud.

3.6.2 Challenge 2—Data Privacy and Security Concerns

- Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks. Many obstacles can be overcome immediately with well-understood technologies such as encrypted storage, virtual LANs, and network middleboxes (e.g., firewalls, packet filters).
- For example, you could encrypt your data before placing it in a cloud. Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries.
- Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms. In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits.
- Another type of attack is the man-in-the-middle attack for VM migrations. In general, passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

3.6.3 Challenge 3—Unpredictable Performance and Bottlenecks

- Multiple VMs can share CPUs and main memory in cloud computing, but I/ O sharing is problematic. For example, to run 75 EC2 instances with the STREAM benchmark requires a mean bandwidth of 1,355 MB/second. However, for each of the 75 EC2 instances to write 1 GB files to the local disk requires a mean disk write bandwidth of only 55 MB/second. This demonstrates the problem of I/O interference between VMs. One solution is to improve I/O architectures and operating systems to efficiently virtualize interrupts and I/O channels.
- Internet applications continue to become more data-intensive. If we assume applications to be “pulled apart” across the boundaries of clouds, this may complicate data placement and transport. Cloud users and

providers have to think about the implications of placement and traffic at every level of the system, if they want to minimize costs. This kind of reasoning can be seen in Amazon's development of its new CloudFront service. Therefore, data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.

3.6.4 Challenge 4—Distributed Storage and Widespread Software Bugs

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

3.6.5 Challenge 5—Cloud Scalability, Interoperability, and Standardization

- The pay-as-you-go model applies to storage and network bandwidth; both are counted in terms of the number of bytes used. Computation is different depending on virtualization level. GAE automatically scales in response to load increases and decreases; users are charged by the cycles used.
- AWS charges by the hour for the number of VM instances used, even if the machine is idle. The opportunity here is to scale quickly up and down in response to load variation, in order to save money, but without violating SLAs.
- Open Virtualization Format (OVF) describes an open, secure, portable,

efficient, and extensible format for the packaging and distribution of VMs. It also defines a format for distributing software to be deployed in VMs. This VM format does not rely on the use of a specific host platform,virtualization platform, or guest operating system. The approach is to address virtual platform-agnostic packaging with certification and integrity of packaged software. The package supports virtual appliances to span more than one VM.

- OVF also defines a transport mechanism for VM templates, and can apply to different virtualization platforms with different levels of virtualization. In terms of cloud standardization, we suggest the ability for virtual appliances to run on any virtual platform. We also need to enable VMs to run on heterogeneous hardware platform hypervisors. This requires hypervisor- agnostic VMs. We also need to realize cross-platform live migration between x86 Intel and AMD technologies and support legacy hardware for load balancing. All these issue are wide open for further research.

3.6.6 Challenge 6—Software Licensing and Reputation Sharing

- Many cloud computing providers originally relied on open source software because the licensing model for commercial software is not ideal for utility computing.
- The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing. One can consider using both pay-for-use and bulk-use licensing schemes to widen the business coverage.
- One customer's bad behavior can affect the reputation of the entire cloud. For instance, blacklisting of EC2 IP addresses by spam-prevention services may limit smooth VM installation.
- An opportunity would be to create reputation-guarding services similar to the "trusted e-mail" services currently offered (for a fee) to services hosted on smaller ISPs. Another legal issue concerns the transfer of legal liability. Cloud providers want legal liability to remain with the customer, and vice

versa. This problem must be solved at the SLA level.

3.7 CLOUD STORAGE OVERVIEW

- Cloud storage involves exactly what the name suggests—storing your data with a cloud service provider rather than on a local system. As with other cloud services, you access the data stored on the cloud via an Internet link.
- A cloud storage system just needs one data server connected to the Internet. A subscriber copies files to the server over the Internet, which then records the data. When a client wants to retrieve the data, he or she accesses the data server with a web-based interface, and the server then either sends the files back to the client or allows the client to access and manipulate the data itself.

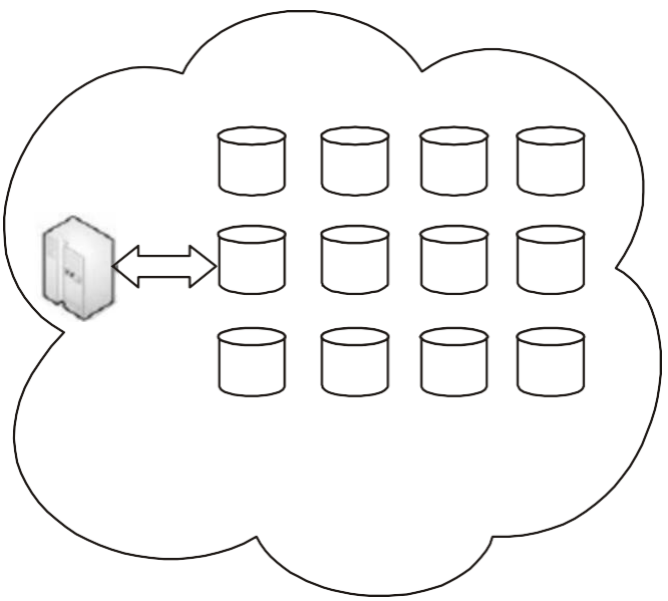


Figure A cloud service provider can simply add more commodity hard drives to increase the organization's capacity.

- More typically, however, cloud storage systems utilize dozens or hundreds of data servers. Because servers require maintenance or repair, it is necessary to store the saved data on multiple machines, providing

redundancy. Without that redundancy, cloud storage systems couldn't assure clients that they could access their information at any given time.

- Most systems store the same data on servers using different power supplies. That way, clients can still access their data even if a power supply fails.
- Many clients use cloud storage not because they've run out of room locally, but for safety. If something happens to their building, then they haven't lost all their data.

3.7.1 Storage as a Service

- The term Storage as a Service (another Software as a Service, or SaaS, acronym) means that a third-party provider rents space on their storage to end users who lack the budget or capital budget to pay for it on their own. It is also ideal when technical personnel are not available or have inadequate knowledge to implement and maintain that storage infrastructure.
- Storage service providers are nothing new, but given the complexity of current backup, replication, and disaster recovery needs, the service has become popular, especially among small and medium-sized businesses.
- The biggest advantage to SaaS is cost savings. Storage is rented from the provider using a cost-per-gigabyte-stored or cost-per-data-transferred model. The end user doesn't have to pay for infrastructure; they simply pay for how much they transfer and save on the provider's servers.

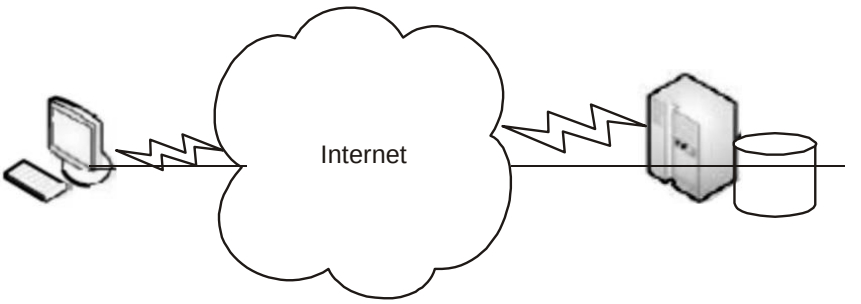


Figure Clients rent storage capacity from cloud storage vendors.

- A customer uses client software to specify the backup set and then

transfers data across a WAN. When data loss occurs, the customer can retrieve the lost data from the service provider.

3.7.2 Providers

- There are hundreds of cloud storage providers on the Web, and more seem to be added each day. Not only are there general-purpose storage providers, but there are some that are very specialized in what they store. Some examples of specialized cloud providers are:

□ Google Docs allows users to upload documents, spreadsheets, and presentations to Google's data servers. Those files can then be edited using a Google application.

□ Web email providers like Gmail, Hotmail, and Yahoo! Mail store email messages on their own servers. Users can access their email from computers and other devices connected to the Internet.

□ Flickr and Picasa host millions of digital photographs. Users can create their own online photo albums.

□ YouTube hosts millions of user-uploaded video files.

□ Hostmonster and GoDaddy store files and data for many client web sites.

□ Facebook and MySpace are social networking sites and allow members to post pictures and other content. That content is stored on the company's servers.

□ MediaMax and Strongspace offer storage space for any kind of digital data.

3.7.3 Security

To secure data, most systems use a combination of techniques:

□ **Encryption** A complex algorithm is used to encode information. To decode the encrypted files, a user needs the encryption key. While it's possible to crack encrypted information, it's very difficult and most hackers don't have access to the amount of computer processing power they would need to crack the code.

- **Authentication processes** This requires a user to create a name and password.
- **Authorization practices** The client lists the people who are authorized to access information stored on the cloud system. Many corporations have multiple levels of authorization. For example, a front-line employee might have limited access to data stored on the cloud and the head of the IT department might have complete and free access to everything.

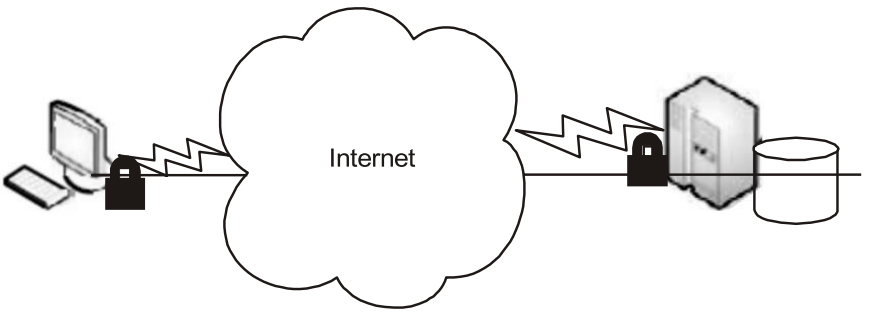


Figure Encryption and authentication are two security measures you can use to keep your data safe on a cloud storage provider.

- But even with these measures in place, there are still concerns that data stored on a remote system is vulnerable. There is always the concern that a hacker will find a way into the secure system and access the data.
- Also, a disgruntled employee could alter or destroy the data using his or her own access credentials.

3.7.4 Reliability

- The other concern is reliability. If a cloud storage system is unreliable, it becomes a liability.
- No one wants to save data on an unstable system, nor would they trust a company that is financially unstable.
- Most cloud storage providers try to address the reliability concern through redundancy, but the possibility still exists that the system could crash and leave clients with no way to access their saved data.
- Reputation is important to cloud storage providers. If there is a perception

that the provider is unreliable, they won't have many clients. And if they are unreliable, they won't be around long, as there are so many players in the market.

3.7.5 Advantages

- Cloud storage is becoming an increasingly attractive solution for organizations. That's because with cloud storage, data resides on the Web, located across storage systems rather than at a designated corporate hosting site. Cloud storage providers balance server loads and move data among various datacenters, ensuring that information is stored close—and thereby available quickly—to where it is used.
- Storing data on the cloud is advantageous, because it allows you to protect your data in case there's a disaster. You may have backup files of your critical information, but if there is a fire or a hurricane wipes out your organization, having the backups stored locally doesn't help. Having your data stored off- site can be the difference between closing your door for good or being down for a few days or weeks.

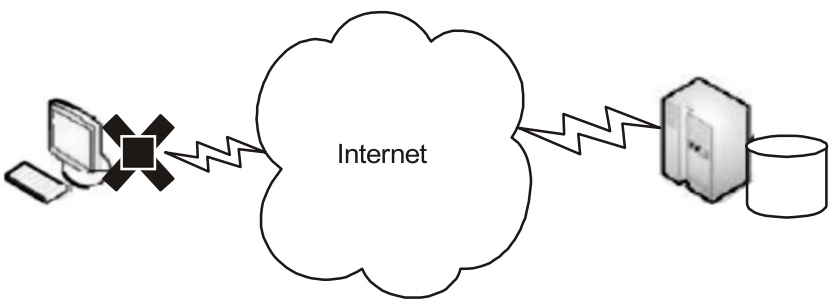


Figure If there is a catastrophe at your organization, having your files backed up at a cloud storage provider means you won't have lost all your data.

- Which storage vendor to go with can be a complex issue, and how your technology interacts with the cloud can be complex. For instance, some products are agent-based, and the application automatically transfers information to the cloud via FTP. But others employ a web front end, and the user has to select local files on their computer to transmit.

- Amazon S3 is the best-known storage solution, but other vendors might be better for large enterprises. For instance, those who offer service level agreements and direct access to customer support are critical for a business moving storage to a service provider.

3.7.6 Cautions

- A mixed approach might be the best way to embrace the cloud, since cloud storage is still immature. That is, don't commit everything to the cloud, but use it for a few, noncritical purposes.
- Large enterprises might have difficulty with vendors like Google or Amazon, because they are forced to rewrite solutions for their applications and there is a lack of portability.
- A vendor like 3tera, however, supports applications developed in LAMP, Solaris, Java, or Windows.NET.
- The biggest deal-breakers when it comes to cloud storage seem to be price and reliability.
- This is where you have to vet your vendor to ensure you're getting a good deal with quality service. One mistake on your vendor's part could mean irretrievable data.
- A lot of companies take the "appetizer" approach, testing one or two services to see how well they mesh with their existing IT systems. It's important to make sure the services will provide what you need before you commit too much to the cloud.

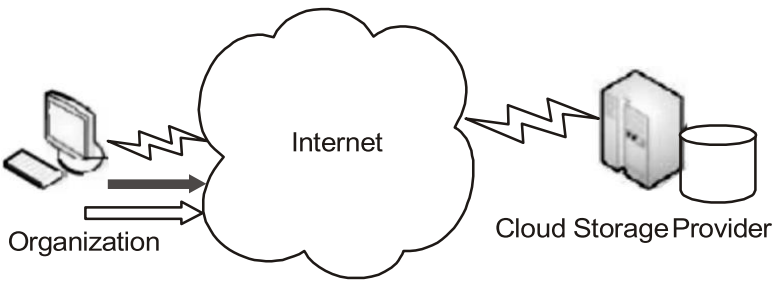


Figure Many companies test out a cloud storage vendor with one or

two services before committing too much to them. This "appetizer" approach ensures the provider can give you what you want.

- Legal issues are also important. For instance, if you have copyrighted material—like music or video—that you want to maintain on the cloud, such an option might not be possible for licensing reasons.
- Also, keep in mind the accountability of your storage provider. Vendors offer different assurances with the maintenance of data. They may offer the service, but make sure you know exactly what your vendor will or will not do in case of data loss or compromise.
- The best solution is to have multiple redundant systems: local and offsite backup; sync and archive.

3.7.7 Outages

- Further, organizations have to be cognizant of the inherent danger of storing their data on the Internet. Amazon S3, for example, dealt with a massive outage in February 2008. The result was numerous client applications going offline.
- Amazon reports that they have responded to the problem, adding capacity to the authentication system blamed for the problem. They also note that no data was lost, because they store multiple copies of every object in several locations.
- The point remains, however, that clients were not able to access their data as they had intended, and so you need to use caution when deciding to pursue a cloud option.

3.7.8 Theft

- You should also keep in mind that your data could be stolen or viewed by those who are not authorized to see it. Whenever your data is let out of your own datacenter, you risk trouble from a security point of view.

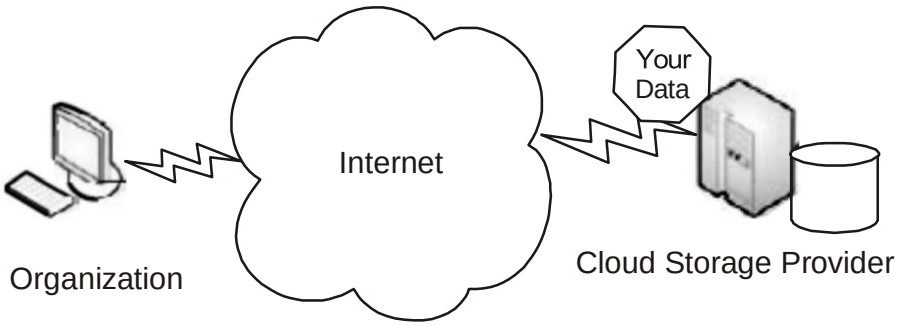


Figure Whenever you let your data out of your organization, you give up a measure of security.

- Also, because storage providers put everything into one pot, so to speak, your company's data could be stored next to a competitor's, and the risk of your competition seeing your proprietary information is real.
- If you do store your data on the cloud, make sure you're encrypting data and securing data transit with technologies like SSL.

3.8 CLOUD STORAGE PROVIDERS

- Amazon and Nirvanix are the current industry top dogs, but many others are in the field, including some well-known names. Google is ready to launch its own cloud storage solution called GDrive. EMC is readying a storage solution, and IBM already has a number of cloud storage options called Blue Cloud.

3.8.1 Amazon Simple Storage Service (S3)

- The best-known cloud storage service is Amazon's Simple Storage Service (S3), which launched in 2006. Amazon S3 is designed to make web-scale computing easier for developers.
- Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the Web. It gives any developer access to the same highly scalable data storage infrastructure that Amazon uses to run its own global

network of web sites.

The service aims to maximize benefits of scale and to pass those benefits on to developers.

- Amazon S3 is intentionally built with a minimal feature set that includes the following functionality:

□ Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each. The number of objects that can be stored is unlimited.

□ Each object is stored and retrieved via a unique developer-assigned key.

□ Objects can be made private or public, and rights can be assigned to specific users.

□ Uses standards-based REST and SOAP interfaces designed to workwith any Internet-development toolkit.

3.8.1.1 Design Requirements

Amazon built S3 to fulfill the following design requirements:

□ **Scalable** Amazon S3 can scale in terms of storage, request rate, and users to support an unlimited number of web-scale applications.

□ **Reliable** Store data durably, with 99.99 percent availability. Amazon says it does not allow any downtime.

□ **Fast** Amazon S3 was designed to be fast enough to support high-performance applications. Server-side latency must be insignificant relative to Internet latency. Any performance bottlenecks can be fixed by simply adding nodes to the system.

□ **Inexpensive** Amazon S3 is built from inexpensive commodity hardware components. As a result, frequent node failure is the norm and must not affect the overall system. It must be hardware-agnostic, so that savings can be captured as Amazon continues to drive down infrastructure costs.

□ **Simple** Building highly scalable, reliable, fast, and inexpensive storage is

difficult. Doing so in a way that makes it easy to use for any application anywhere is more difficult. Amazon S3 must do both.

A forcing function for the design was that a single Amazon S3 distributed system must support the needs of both internal Amazon applications and external developers of any application. This means that it must be fast and reliable enough to run Amazon.com's web sites, while flexible enough that any developer can use it for any data storage need.

3.8.1.2 Design Principles

Amazon used the following principles of distributed system design to meet Amazon S3 requirements:

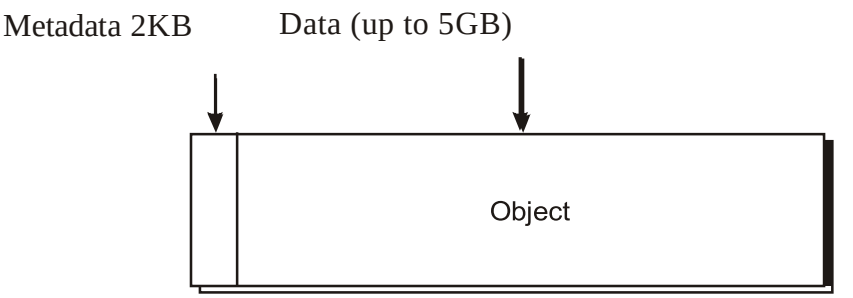
- **Decentralization** It uses fully decentralized techniques to remove scaling bottlenecks and single points of failure.
- **Autonomy** The system is designed such that individual components can make decisions based on local information.
- **Local responsibility** Each individual component is responsible for achieving its consistency; this is never the burden of its peers.
- **Controlled concurrency** Operations are designed such that no or limited concurrency control is required.
- **Failure toleration** The system considers the failure of components to be a normal mode of operation and continues operation with no or minimal interruption.
- **Controlled parallelism** Abstractions used in the system are of such granularity that parallelism can be used to improve performance and robustness of recovery or the introduction of new nodes.
- **Small, well-understood building blocks** Do not try to provide a single service that does everything for everyone, but instead build small components that can be used as building blocks for other services.
- **Symmetry** Nodes in the system are identical in terms of functionality, and

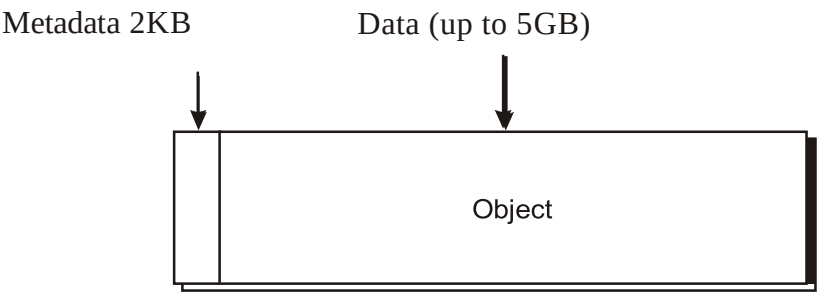
require no or minimal node-specific configuration to function.

- **Simplicity** The system should be made as simple as possible, but no simpler.

3.8.1.3 How S3 Works

- Amazon keeps its lips pretty tight about how S3 works, but according to Amazon, S3's design aims to provide scalability, high availability, and low latency at commodity costs.
- S3 stores arbitrary objects at up to 5GB in size, and each is accompanied by up to 2KB of metadata. Objects are organized by *buckets*. Each bucket is owned by an AWS account and the buckets are identified by a unique, user- assigned key.





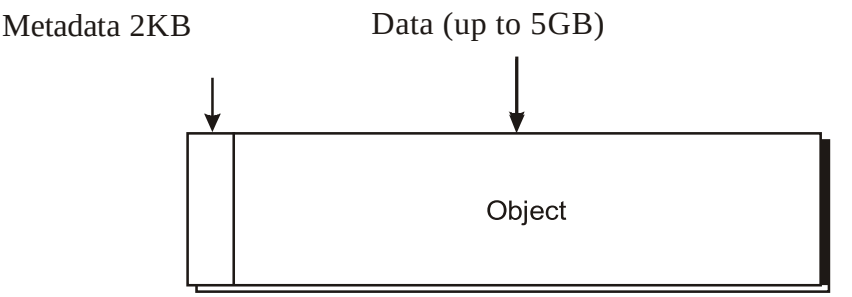


Figure Multiple objects are stored in buckets in Amazon S3.

- Buckets and objects are created, listed, and retrieved using either a REST-style or SOAP interface. Objects can also be retrieved using the HTTP GET interface or via BitTorrent. An access control list restricts who can access the data in each bucket.
- Bucket names and keys are formulated so that they can be accessed using HTTP.
- Requests are authorized using an access control list associated with each bucket and object, for instance:

http://s3.amazonaws.com/examplebucket/examplekey

http://examplebucket.s3.amazonaws.com/examplekey

- The Amazon AWS Authentication tools allow the bucket owner to create an authenticated URL with a set amount of time that the URL willbe valid. For instance, you could create a link to your data on the cloud, give that link to someone else, and they could access your data for an amount of time you predetermine, be it 10 minutes or 10 hours.
- Bucket items can also be accessed via a BitTorrent feed, enabling S3 to act as a seed for the client. Buckets can also be set up to save HTTP log information to another bucket. This information can be used for later data mining.
- "Amazon S3 is based on the idea that quality Internet-based storage should be taken for granted," said Andy Jassy, vice president of Amazon Web Services. "It helps free developers from worrying about where they are going to store data, whether it will be safe and secure, if it will be available when they need it, the costs associated with server maintenance, or whether they have enough storage available. Amazon S3 enables developers to focus on innovating with data, rather than figuring out how to store it."
- S3 lets developers pay only for what they consume, and there is no minimum fee.
- Developers pay just $0.15 per gigabyte of storage per month and $0.20 per gigabyte of data transferred. This might not seem like a lot of money but storing 1TB would be $1800 per year alone, whereas an internal 1TB drive these days costs about $100 to own outright.
- So it's really not so much about the cost of storage as it is about the total cost to serve.

3.8.1.4 Early S3 Applications

- The science team at the University of California Berkeley responsible for

NASA's "Stardust@Home" project (http://stardustathome.ssl.berkeley.edu) is using Amazon S3 to store and deliver the 60 million images that represent the data collected from their dust particle aerogel experiment. These images will be delivered to 100,000 volunteers around the world who scan the images looking for dust particles from comet Wild2.

- "We quickly ran into challenges when we started the project using our own infrastructure," said Andrew Westphal, project director of Stardust@Home.
- "Using Amazon S3 has allowed us to proceed without having to worry about building out the massive storage infrastructure we realized that we needed to successfully complete the project.
- The fact that Amazon S3 is an Internet-connected storage service is particularly useful to us as we expect the data examination phase of the project to take only a few months. We can quickly ramp up and back down again without a huge investment."

UNIT 3

PART A

1. **Define cloud computing?**

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

The NIST definition also identifies

- 5 essential characteristics
- 3 service models

- 4 deployment models

2. List out Actors in Cloud Computing.

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

3. Define Cloud Consumer.

- The cloud consumer is the principal stakeholder for the cloud computing service. A cloud consumer represents a person or organization that maintains a business relationship with, and uses the service from a cloud provider.
- A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider, and uses the service. The cloud consumer may be billed for the service provisioned, and needs to

arrange payments accordingly.

4. What are the features of Cloud Auditor?

- A cloud auditor is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon.
- Audits are performed to verify conformance to standards through review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc.

5. How Cloud Broker is useful for cloud services?

- A cloud consumer may request cloud services from a cloud broker, instead of contacting a cloud provider directly.
- A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers.

6. What are the three categories of cloud broker?

- ***Service Intermediation*:**
- ***Service Aggregation*:**
- ***Service Arbitrage*:**

7. Define Cloud Carrier.

1. A cloud carrier acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers. Cloud carriers provide access to consumers through network, telecommunication and other access devices.
2. For example, cloud consumers can obtain cloud services through

network access devices, such as computers, laptops, mobile phones, mobile Internet devices (MIDs), etc.

8. What is public cloud?

- A public cloud is built over the Internet and can be accessed by any user who has paid for the service.
- Public clouds are owned by service providers and are accessible through a subscription.
- Many public clouds are available, including Google App Engine (GAE),Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com's Force.com.

9. What is private cloud?

- A private cloud is built within the domain of an intranet owned by a single organization.
- Therefore, it is client owned and managed, and its access is limited to the owning clients and their partners.

10. Highlights six design objectives for cloud computing

- Shifting computing from desktops to data centers
- Service provisioning and cloud economics
- Scalability in performance
- Data privacy protection

- High quality of cloud services
- New standards and interfaces

11. What are the essential characteristics of cloud computing?

- On-demand self services,
- Broad network access,
- Resource pooling,
- Rapid elasticity,
- Measured service.

12. What are the advantages of cloud computing?

- Cost efficient
- Almost unlimited storage
- Backup and recovery
- Automatic software integration
- Easy access to information
- Quick development.

13. **List out the disadvantages of cloud computing?**

- Technical issues

- Security in the cloud
- Prone to attack

14. Discuss the features of Community Cloud.

- A community cloud serves a group of Cloud Consumers which have shared concerns such as mission objectives, security, privacy and compliance policy, rather than serving a single organization as does a private cloud.
- Similar to private clouds, a community cloud may be managed by the organizations or by a third party, and may be implemented on customer premise (i.e. *on-site community cloud*) or outsourced to a hosting company (i.e. *outsourced community cloud*).

15. What is Hybrid Cloud?

A hybrid cloud is a composition of two or more clouds (on-site private, on- site community, off-site private, off-site community or public) that remain as distinct entities but are bound together by standardized or proprietary technology that enables data and application portability.

16. Define Service Orchestration

Service Orchestration refers to the composition of system components to support the Cloud Providers activities in arrangement, coordination and management of computing resources in order to provide cloud services to Cloud Consumers.

17. What is the user role in Infrastructure as a Service model?

- This model allows users to use virtualized IT resources for computing, storage, and networking.
- In short, the service is performed by rented cloud infrastructure. The user can deploy and run his applications over his chosen OS environment.
- The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components.

18. Define the term PAAS.

- Platform as a Service
- To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment. Such a platform includes operating system and runtime library support.
- This has triggered the creation of the PaaS model to enable users to develop and deploy their user applications

19. Discuss about Amazon S3.

- Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the Web.
- It gives any developer access to the same highly scalable data storage infrastructure that Amazon uses to run its own global network of web sites.

20. What are the functionalities of Amazon S3.

- Amazon S3 is intentionally built with a minimal feature set that includes the following functionality:

- Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each. The number of objects that can be stored is unlimited.
- Each object is stored and retrieved via a unique developer-assigned key.
- Objects can be made private or public, and rights can be assigned to specific users.
- Uses standards-based REST and SOAP interfaces designed to work with any Internet-development toolkit.

21. List out the design requirements of Amazon built S3.

- Scalable
- Reliable
- Fast
- Inexpensive
- Simple

22. What are the principles of distributed system design used in Amazon S3?

- Decentralization
- Autonomy
- Local responsibility
- Controlled concurrency
- Failure toleration
- Controlled parallelism
- Small, well-understood building blocks

- Symmetry
- Simplicity

23. List out the functions of cloud provider.

- A cloud provider is a person, an organization; it is the entity responsible for making a service available to interested parties.
- A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access.

PART B

1. Explain the Layered architectural development of the cloud platform in detail.
2. Discuss the NIST cloud computing reference architecture with neat diagram.
3. Explain cloud deployment models with necessary diagrams.
4. Discuss IaaS with examples.
5. Explain Paas with examples.
6. Discuss in detail about the Saas.
7. Explain the six open challenges in cloud architecture development.
8. What is storage as a services? and explain the advantages of cloud storage in detail.
9. Explain amazon simple storage service with its design requirements.

UNIT IV
RESOURCE MANAGEMENTIN AND SECURITY CLOUD

Inter Cloud Resource Management - Resource Provisioning and Resource Provisioning Methods - Global Exchange of Cloud Resources - Security Overview - Cloud Security Challenges - Software-as-a-Service Security - Security Governance - Virtual Machine Security - IAM - Security Standards.

4.1 INTER-CLOUD RESOURCE MANAGEMENT

4.1.1 Extended Cloud Computing Services

- Figure 4.1 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, respectively. The cloud platform provides PaaS, which sits on top of the IaaS infrastructure. The top layer offers SaaS.
- These must be implemented on the cloud platforms provided. Although the three basic models are dissimilar in usage, as shown in Table 4.1, they are built one on top of another.
- The bottom three layers are more related to physical requirements. The bottommost layer provides Hardware as a Service (HaaS). The next layer is for interconnecting all the hardware components, and is simply called Network as a Service (NaaS). Virtual LANs fall within the scope of NaaS. The next layer up offers Location as a Service (LaaS), which provides a collocation service to house, power, and secure all the physical hardware

and network resources. Some authors say this layer provides Security as a Service ("SaaS"). The cloud infrastructure layer can be further subdivided as Data as a Service (DaaS) and Communication as a Service (CaaS) in addition to compute and storage in IaaS.

- As shown in Table 4.1, cloud players are divided into three classes: (1) cloud service providers and IT administrators, (2) software developers or vendors, and (3) end users or business users. These cloud players vary in their roles under the IaaS, PaaS, and SaaS models. The table entries distinguish the three cloud models as viewed by different players.
- From the software vendors' perspective, application performance on a given cloud platform is most important. From the providers' perspective, cloud infrastructure performance is the primary concern. From the end users' perspective, the quality of services, including security, is the most important.

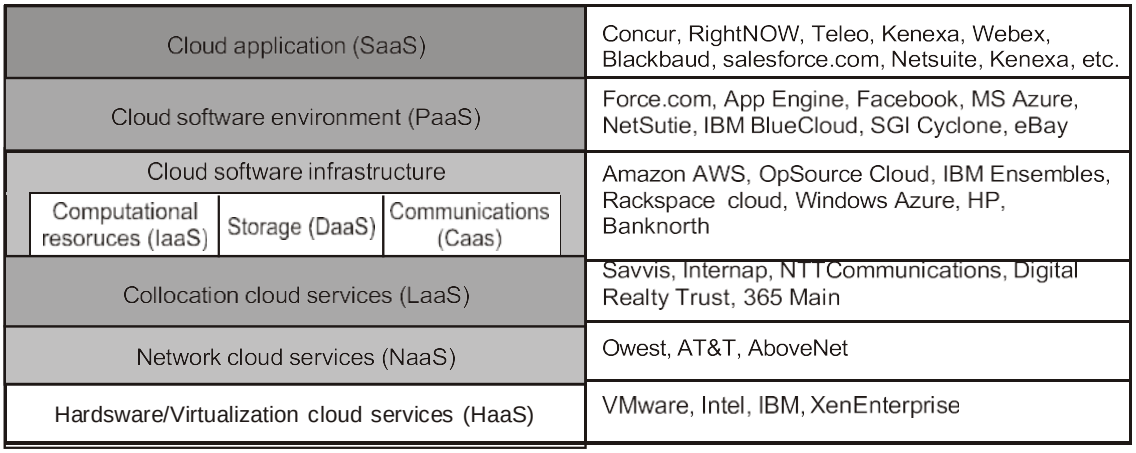


FIGURE 4.1 A stack of six layers of cloud services and their providers.

Table 4.1 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 storedata	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

4.1.1.1 Cloud Service Tasks and Trends

- Cloud services are introduced in five layers. The top layer is for SaaS applications, as further subdivided into the five application areas in Figure 4.1, mostly 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, 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, 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.
- The vertical cloud services in Figure 4.3 refer to a sequence of cloud services that are mutually supportive. Often, cloud mashup is practiced in vertical cloud applications.

4.1.1.2 Software Stack for Cloud Computing

- Despite the various types of nodes in the cloud computing cluster, the overall software stacks are built from scratch to meet rigorous goals (see Table 4.1).
- Developers have to consider how to design the system to meet critical requirements such as high throughput, HA, and fault tolerance.
- Even the operating system might be modified to meet the special requirement of cloud data processing.
- Based on the observations of some typical cloud computing instances, such as Google, Microsoft, and Yahoo!, 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 transparentto the upper layers.

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

4.1.1.3 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.
- One of the most important facilities is the cluster job management system. The scheduler queues the tasks submitted to the whole cluster and assignsthe tasks to the processing nodes according to node availability. The distributed scheduler for the cloud application has special characteristics that can support cloud applications, such as scheduling the programs written in MapReduce style.
- The runtime support system keeps the cloud cluster working properly with high efficiency. Runtime support is software needed in browser-initiated applications applied by thousands of cloud customers.
- The SaaS model provides the software applications as a service, rather than letting users purchase the software.
- As a result, on the customer side, there is no upfront investment in serversor software licensing. On the provider side, costs are rather low, compared with conventional hosting of user applications. The customer data is stored in the cloud that is either vendor proprietary or a publicly hosted cloud supporting PaaS and IaaS.

4.2 RESOURCE PROVISIONING AND PLATFORM DEPLOYMENT

- The emergence of computing clouds suggests fundamental changes in software and hardware architecture.
- Cloud architecture puts more emphasis on the number of processor cores or VM instances.
- Parallelism is exploited at the cluster node level.

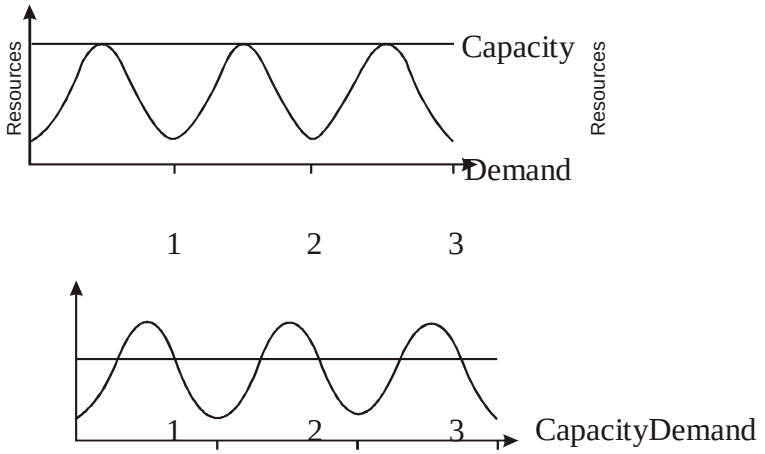
4.2.1 Provisioning of Compute Resources (VMs)

- 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, power management, and conflicts in signed SLAs between consumers and service providers.
- Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures.
- Resource provisioning schemes also demand fast discovery of services and data in cloud computing infrastructures.
- In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration, and fast recovery from failures.
- To deploy VMs, users treat them as physical hosts with customized operating systems for specific applications.
- For example, Amazon's EC2 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. Power-efficient schemes for caching, query processing, and thermal management are mandatory due to increasing energy waste by heat dissipation from data centers.
- Public or private clouds promise to streamline the on-demand provisioning of software, hardware, and data as a service, achieving economies of scale in IT deployment and operation.

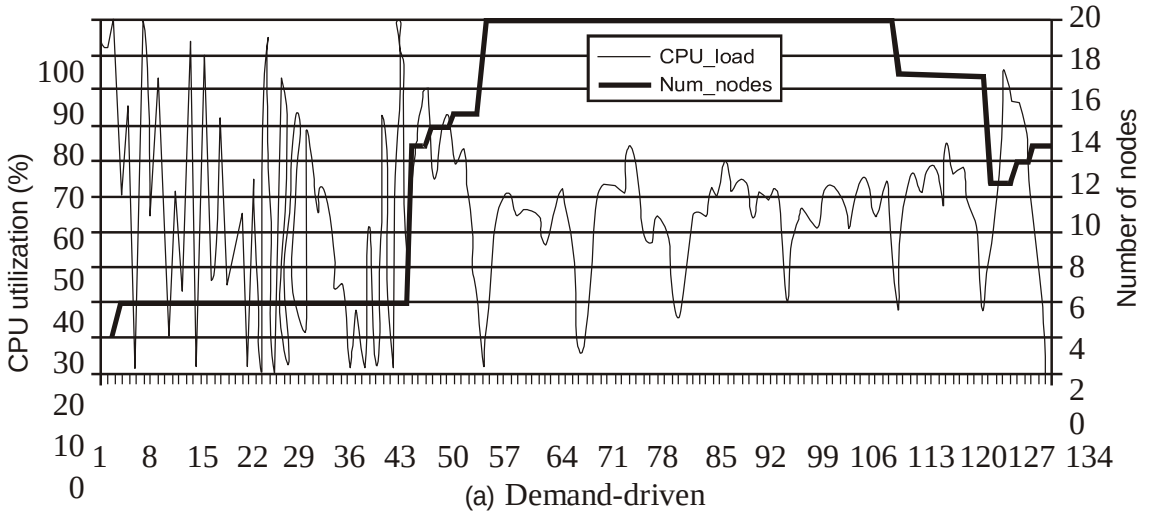
4.2.2 Resource Provisioning Methods

- Figure 4.2 shows three cases of static cloud resource provisioning policies.
- In case (a), overprovisioning with the peak load causes heavy resource waste (shaded area).
- In case (b), underprovisioning (along the capacity line) of resources results in losses by both user and provider in that paid demand by the users (the shaded area above the capacity) is not served and wasted resources still exist for those demanded areas below the provisioned capacity.
- In case (c), the constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste.
- The user may give up the service by canceling the demand, resulting in reduced revenue for the provider. Both the user and provider may be losersin resource provisioning without elasticity.
- The demand-driven method provides static resources and has been used in grid computing for many years.
- The event-driven method is based on predicted workload by time.
- The popularity-driven method is based on Internet traffic monitored.
- The characterization of these resource provisioning methods as follows (see Figure 4.3).

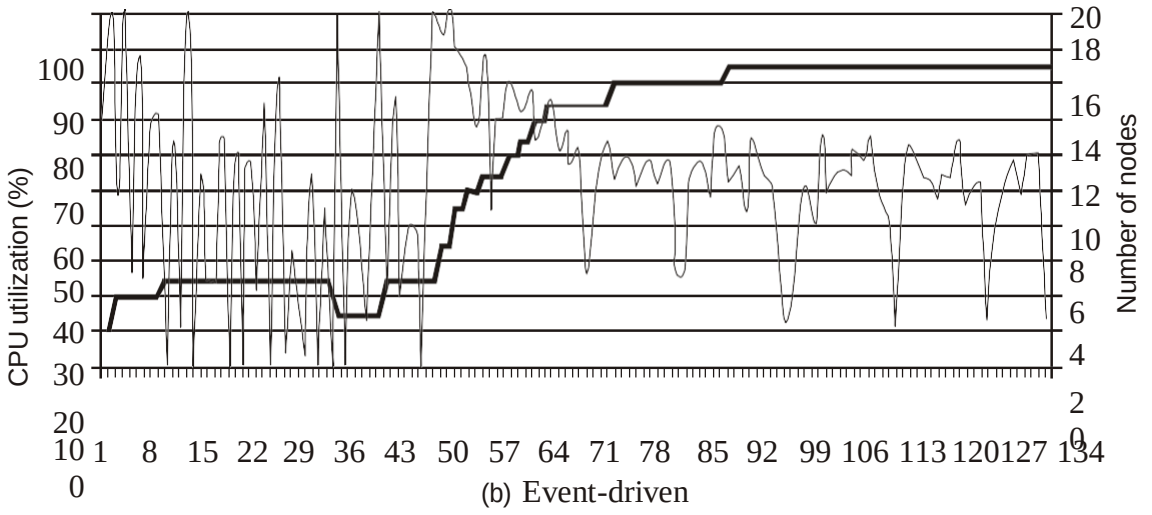


4.2.3 Demand-Driven Resource Provisioning

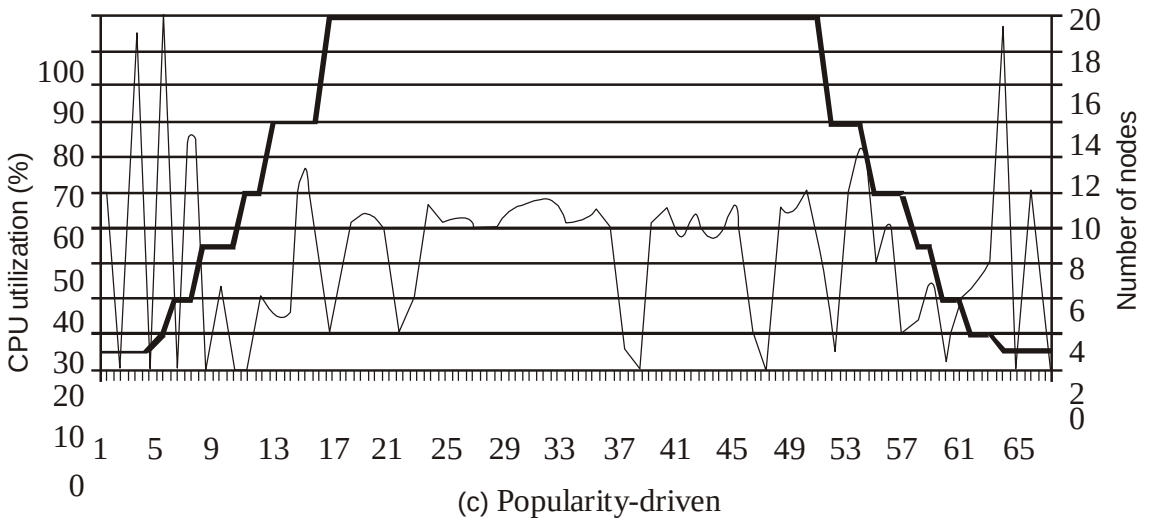
- This method adds or removes computing instances based on the current utilization level of the allocated resources.
- The demand-driven method automatically allocates two Xeon processorsfor the user application, when the user was using one Xeon processor more than 60 percent of the time for an extended period.
- In general, when a resource has surpassed a threshold for a certain amountof time, the scheme increases that resource based on demand. When a resource is below a threshold for a certain amount of time, that resource could be decreased accordingly. Amazon implements such an auto-scale feature in its EC2 platform. This method is easy to implement. The scheme does not work out right if the workload changes abruptly.
- The x-axis in Figure 4.3 is the time scale in milliseconds. In the beginning, heavy fluctuations of CPU load are encountered. All three methods have demanded a few VM instances initially. Gradually, the utilization rate becomes more stabilized with a maximum of 20 VMs (100 percent utilization) provided for demand-driven provisioning in Figure 4.3(a).
- However, the event-driven method reaches a stable peak of 17 VMs toward the end of the event and drops quickly in Figure 4.3(b).
- The popularity provisioning shown in Figure 4.3(c) leads to a similar fluctuation with peak VM utilization in the middle of the plot.



(a) Demand-driven



(b) Event-driven



(c) Popularity-driven

FIGURE 4.3 EC2 performance results on the AWS EC2 platform, collected from experiments at the University of Southern California using three resource provisioning methods.

4.2.4 Event-Driven Resource Provisioning

- 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.

4.2.5 Popularity-Driven Resource Provisioning

- In this method, the Internet searches for popularity of certain applications and creates the instances by popularity demand. The scheme anticipates increased traffic with popularity. Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct.
- Resources may be wasted if traffic does not occur as expected.
- In Figure 4.3(c), EC2 performance by CPU utilization rate (the dark curve with the percentage scale shown on the left) is plotted against the number of VMs provisioned (the light curves with scale shown on the right, with a maximum of 20 VMs provisioned).

4.2.6 Dynamic Resource Deployment

- The cloud uses VMs as building blocks to create an execution environment across multiple resource sites.
- Dynamic resource deployment can be implemented to achieve scalability in performance.
- The Inter-Grid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources. Peering arrangements established between gateways enable the allocationof resources from multiple grids to establish the execution environment.

- In Figure 4.4, a scenario is illustrated by which an intergrid gateway (IGG) allocates resources from a local cluster to deploy applications in three steps: (1) requesting the VMs, (2) enacting the leases, and (3) deploying the VMs as requested.
- Under peak demand, this IGG interacts with another IGG that can allocate resources from a cloud computing provider.

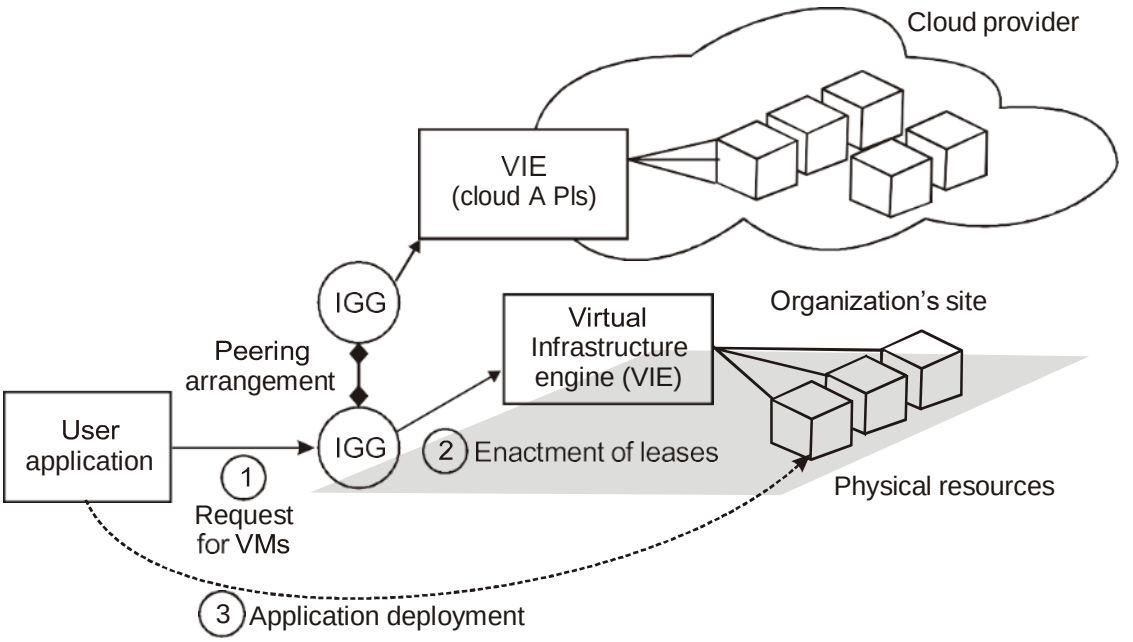


FIGURE 4.4 Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from aLocal cluster to interact with the IGG of a public cloud provider.

- 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.
- Request redirection policies determine which peering grid InterGrid selectsto process a request and a price for which that grid will perform the task. An IGG can also allocate resources from a cloud provider.
- The cloud system creates a virtual environment to help users deploy their applications. These applications use the distributed grid resources.

- The InterGrid allocates and provides a distributed virtual environment (DVE). This is a virtual cluster of VMs that runs isolated from other virtual clusters.
- A component called the DVE manager performs resource allocation and management on behalf of specific user applications.
- The core component of the IGG is a scheduler for implementing provisioning policies and peering with other gateways.
- The communication component provides an asynchronous message-passing mechanism. Received messages are handled in parallel by a thread pool.

4.2.7 Provisioning of Storage Resources

- The data storage layer is built on top of the physical or virtual servers. As the cloud computing applications often provide service to users, it is unavoidable that the data is stored in the clusters of the cloud provider. The service can be accessed anywhere in the world. One example is e-mail systems.
- A typical large e-mail system might have millions of users and each user can have thousands of e-mails and consume multiple gigabytes of disk space. Another example is a web searching application.
- In storage technologies, hard disk drives may be augmented with solid-state drives in the future. This will provide reliable and high-performance data storage.
- The biggest barriers to adopting flash memory in data centers have been price, capacity, and, to some extent, a lack of sophisticated query processing techniques. However, this is about to change as the I/O bandwidth of solid-state drives becomes too impressive to ignore.
- A distributed file system is very important for storing large-scale data. However, other forms of data storage also exist. Some data does not need the namespace of a tree structure file system, and instead, databases are built with stored data files.
- In cloud computing, another form of data storage is (Key, Value) pairs. Amazon S3 service uses SOAP to access the objects stored in the cloud.
- Table 4.2 outlines three cloud storage services provided by Google, Hadoop, and Amazon.

- For example, Google's GFS stores web data and some other data, such as geographic data for Google Earth. A similar system from the open source community is the Hadoop Distributed File System (HDFS) for Apache. Hadoop is the open source implementation of Google's cloud computing infrastructure. Similar systems include Microsoft's Cosmos file system for the cloud.
- Despite the fact that the storage service or distributed file system can be accessed directly, similar to traditional databases, cloud computing does provide some forms of structure or semistructure database processing capability.
- For example, applications might want to process the information contained in a web page. Web pages are an example of semistructural data in HTML format.
- If some forms of database capability can be used, application developers will construct their application logic more easily. Another reason to build a database-like service in cloud computing is that it will be quite convenient for traditional application developers to code for the cloud platform. Databases are quite common as the underlying storage device for many applications. Thus, such developers can think in the same way they do for traditional software development.
- Hence, in cloud computing, it is necessary to build databases like large- scale systems based on data storage or distributed file systems. The scale of such a database might be quite large for processing huge amounts of data.
- The main purpose is to store the data in structural or semi-structural ways so that application developers can use it easily and build their applications rapidly. Traditional databases will meet the performance bottleneck while the system is expanded to a larger scale. However, some real applications do not need such strong consistency. The scale of such databases can be quite large. Typical cloud databases include BigTable from Google, SimpleDB from Amazon, and the SQL service from Microsoft Azure.

Table 4.2 Storage Services in Three Cloud Computing Systems

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

4.3 GLOBAL EXCHANGE OF CLOUD RESOURCES

- In order to support a large number of application service consumers from around the world, cloud infrastructure providers (i.e., IaaS 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, currently Amazon expects its cloud customers (i.e., SaaS providers) to express a preference regarding where they want their application services to be hosted. Amazon does not provide seamless/automatic mechanisms for scaling its hosted services across multiple geographically distributed data centers.

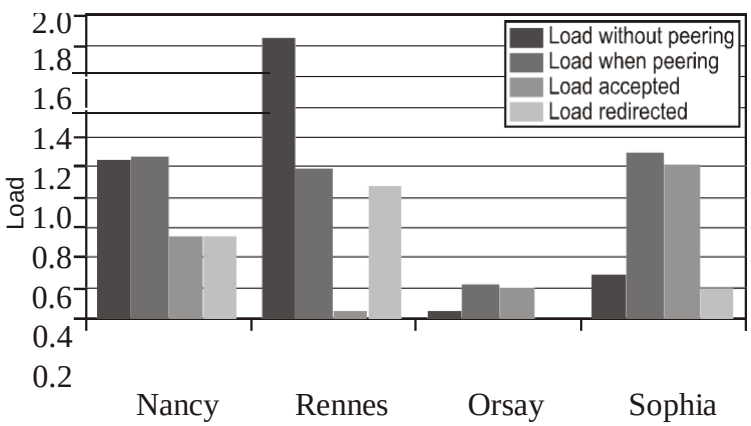


FIGURE 4.5 Cloud loading results at four gateways at resource sites in the Grid'5000 system.

This approach has many shortcomings.

- First, 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.
- Second, 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 centersof a cloud provider or providers supporting dynamic scaling of applications across multiple domains in order to meet QoS targets of cloud customers Figure 4.6 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 application service (SaaS) 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 kind of requirement often arises in enterprises with global operations and applications such as Internet services, media hosting, and Web 2.0 applications.
- This necessitates federation of cloud infrastructure service providers for seamless provisioning of services across different cloud providers.
- To realize this, the Cloudbus Project at the University of Melbourne has proposed InterCloud architecture supporting brokering and exchange of cloud resources for scaling applications across multiple clouds.
- By realizing InterCloud architectural principles in mechanisms in their offering, 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, where application service providers such as Salesforce.com host

their services based on negotiated SLA contracts driven by competitive market prices; and deliver on-demand, reliable, cost-effective, and QoS- aware services based on virtualization technologies while ensuring high QoS standards and minimizing service costs.

- They need to be able to utilize market-based utility models as the basis for provisioning of virtualized software services and federated hardware infrastructure among users with heterogeneous applications.

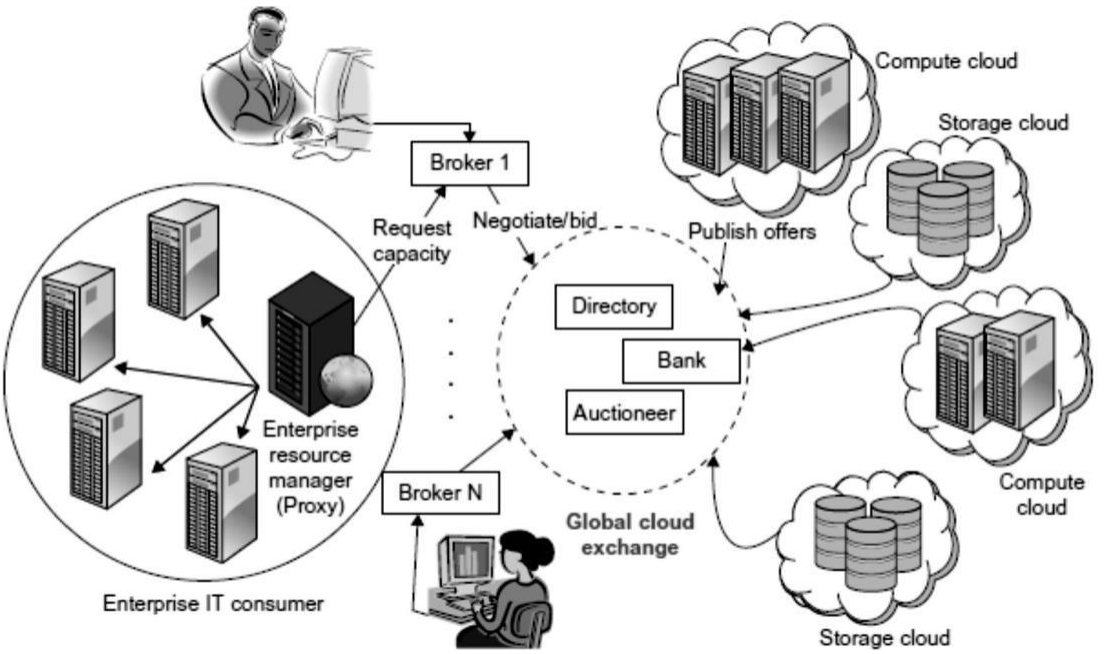


FIGURE 4.6 Inter-cloud exchange of cloud resources through brokering.

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

- The architecture cohesively couples the administratively and topologically distributed storage and compute capabilities of clouds as part of a single resource leasing abstraction.

- The system will ease the crossdomain capability integration for on-demand, flexible, energy-efficient, and reliable access to the infrastructure based on virtualization technology.

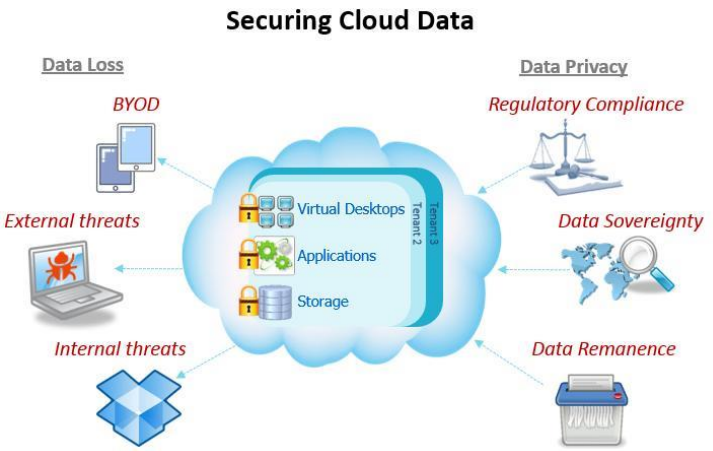
- The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers. It aggregates the infrastructure demands from application brokers and evaluates them against the available supply currently published by the cloud coordinators.
- It supports trading of cloud services based on competitive economic models such as commodity markets and auctions.
- CEx allows participants to locate providers and consumers with fitting offers. Such markets enable services to be commoditized, and thus will pave the way for creation of dynamic market infrastructure for trading based on SLAs.
- 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.

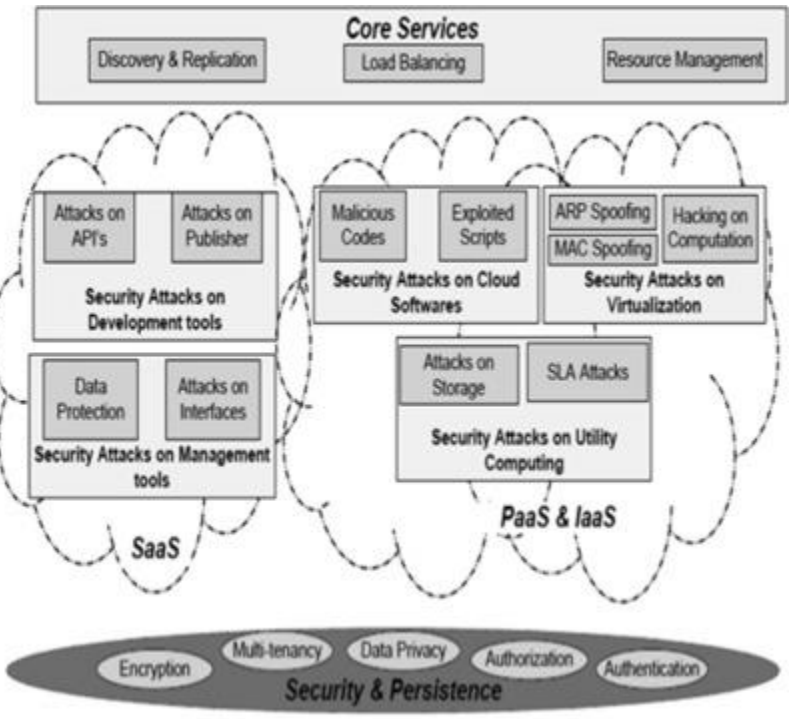
4.4 SECURITY IN THE CLOUD – OVERVIEW

- Cloud security is a set of policies, procedures, and technologies used to protect cloud-based data, applications, and infrastructure from cyber threats. Cloud security is a shared responsibility between the cloud provider and the customer. The cloud provider is responsible for securing the underlying infrastructure, while the customer is responsible for securing their data and applications.
- Here is an overview of the key components of cloud security:
- Identity and access management (IAM): IAM controls who has access to cloud resources and what they can do with those resources.
- Data encryption: Data encryption protects data from unauthorized access, even if it is intercepted.
- Network security: Network security controls protect cloud resources from unauthorized access and malicious traffic.
- Application security: Application security controls protect cloud-based applications from vulnerabilities and attacks.
- Security monitoring and incident response: Security monitoring and incident response tools help to detect and respond to security threats in real time.
- In addition to these key components, cloud security also includes other

important areas such as:

- Compliance: Compliance with industry regulations and standards is essential for many businesses. Cloud security solutions can help businesses to comply with these requirements.
- Risk management: Risk management is the process of identifying, assessing, and mitigating security risks. Cloud security solutions can help businesses to manage their security risks more effectively.
- Cloud security is essential for any business that is using cloud-based services. By implementing a comprehensive cloud security strategy, businesses can protect their data, applications, and infrastructure from cyber threats.
- Here are some tips for improving cloud security:
- Use strong passwords and multi-factor authentication for all cloud accounts.
- Implement least privilege access controls, so that users only have access to the resources they need.
- Encrypt all data at rest and in transit.
- Regularly back up your data and test your backup and restore procedures.
- Use a cloud security monitoring solution to detect and respond to security threats in real time.
- Keep your cloud software and applications up to date with the latest security patches.





Security Risk in Cloud Computing

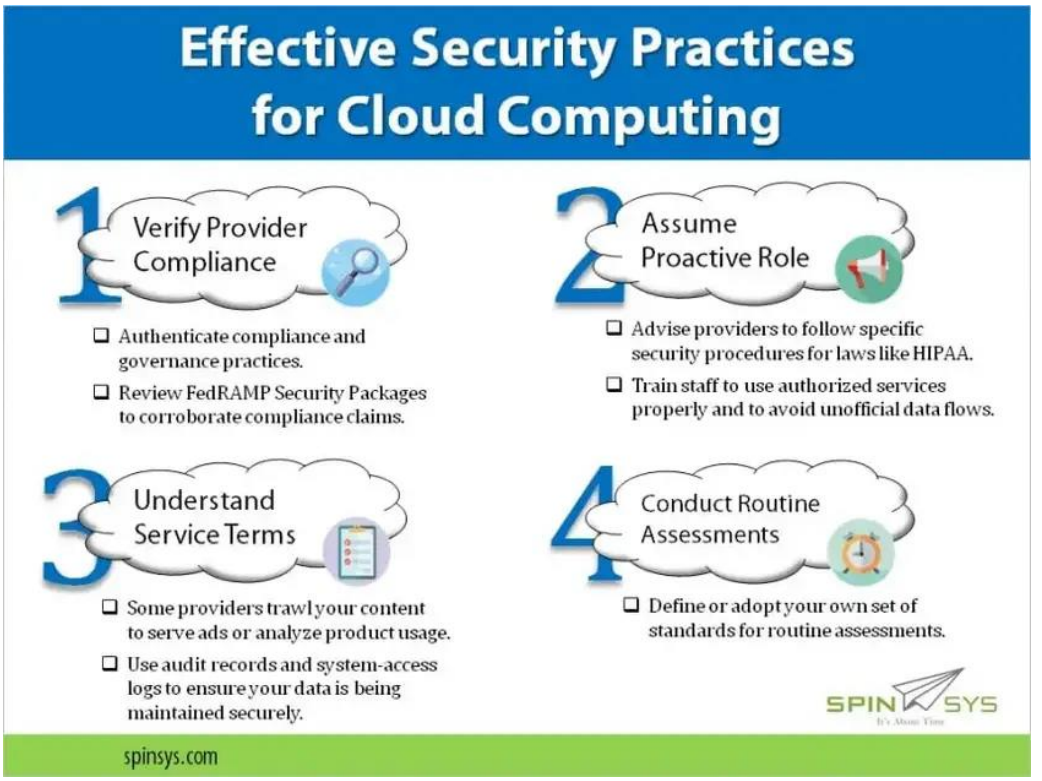
Cloud service providers are leveraging virtualization technologies combined with self-service capabilities for computing resources via the Internet. In these service provider environments, virtual machines from multiple organizations have to be co-located on the same physical server in order to maximize the efficiencies of virtualization.

- Cloud service providers must learn from the managed service provider (MSP) model and ensure that their customers' applications and data are secure if they hope to retain their customer base and competitiveness.
- Today, enterprises are looking toward cloud computing horizons to expand their on-premises infrastructure, but most cannot afford the risk of compromising the security of their applications and data. For example, IDC recently conducted a survey (see Figure 4.7) of 244 IT executives/CIOs and their line-of-business (LOB) colleagues to gauge their opinions andunderstand their companies' use of IT cloud services.
- Security ranked first as the greatest challenge or issue of cloud computing.

- **Software-as-a-Service** is a model of software deployment in which an application is licensed for use as a service provided to customers on demand. On-demand licensing and use relieves the customer of the burden of equipping a device with every application to be used. Gartner predicts that 30% of new software will be delivered via the SaaS model by 2010.

- **Platform-as-a-Service** is an outgrowth of the SaaS application delivery model. With the PaaS model, all of the facilities required to support the complete life cycle of building and delivering web applications and services are available to developers, IT managers, and end users entirely from the Internet, without software downloads or installation. PaaS is also sometimes known as "cloudware." PaaS offerings include workflow facilities for application design, application development, testing, deployment, and hosting, as well as application services such as team collaboration, web service integration and marshalling, database integration, security, scalability,storage, persistence, state management, application versioning, application instrumentation, and developer community facilitation. These services are provisioned as an integrated solution over the web.

- **Infrastructure-as-a-Service** is the delivery of computer infrastructure (typically a platform virtualization environment) as a service. These "virtual infrastructure stacks" are an example of the everything-as-a-service trend and share many of the common characteristics. Rather than purchasing servers, software, data center space, or network equipment, clients buy these resources as a fully outsourced service. The service is typically billed on a utility computing basis, and the quantity of resources consumed (and therefore the cost) typically reflects the level of activity. It is an evolution of web hosting and virtual private server offerings.



- Inspired by the IT industry's move toward SaaS, in which software is not purchased but rented as a service from providers, **IT-as-a-Service (ITaaS)** is being proposed to take this concept further, to bring the service model right to your IT infrastructure. The modern IT organization must run itself as a separate operation and become more strategic in operational decisions. Many organizations are in the process of transforming their IT departments into self-sustaining cost-center operations, treating internal users as if they were customers.

- Many large IT organizations have adopted the Information Technology Infrastructure Library (ITIL) framework to help with this transformation. Organizations can harness their help desks, avoid downtime resulting from unauthorized changes, and deliver better service to their internal customers simply by adopting best practices for managing service requests, changes, and IT assets. The adoption of IT-as-a-Service can help enterprise IT functions focus on strategic alignment with business goals.
- However, if efforts in this direction are poorly implemented, organizations risk further alienating their technical support staff from the rest of the organization—turning them into order takers for the enterprise rather than business advisers.
- When it is done properly, a customer-centric IT department increases productivity, drives up project success rates, and creates a higher profile for technology within the organization.
- While enterprises cope with defining the details of cloud computing, the single, unifying theme is *service*. Cloud computing, on-demand applications, and managed security are now perceived as part of an emerging ITaaS paradigm.
- Current industry buzz seems to reinforce the message that significant investments of capital, time, and intellectual resources are indeed being directed toward offering next-generation information and communication technology (ICT) infrastructure, which may allow enterprises to outsource IT completely and confidently.
- Infrastructure vendors are also jumping on this bandwagon. Amazon has been a pioneer, with the release of Amazon S3 (Storage-as-a-Service). With the maturation of virtualization technologies, the adoption of virtual infrastructure and storage-on-demand services will accelerate along with the SaaS model.
- There are some key financial benefits in moving to an ITaaS model, such has not having to incur capital costs; having a transparent, monthly pricing plan; scalability; and reasonable costs of expansion. Operational benefits of ITaaS include increased reliability because of a centralized infrastructure, which can ensure that critical services and applications are monitored continually; software flexibility, with centrally maintained products that allow for quick rollout of new functionalities and updates; and data security, since company data can be stored on owner-managed premises and backedup

using encryption to a secure off-site data center.

- Another service is **Anything-as-a-Service (XaaS),** which is also a subset of cloud computing. XaaS broadly encompasses a process of activating reusable software components over the network. The most common and successful example is Software-as-a-Service.
- The growth of "as-a-service" offerings has been facilitated by extremely low barriers to entry (they are often accessible for free or available as recurring charges on a personal credit card). As a result, such offerings have been adopted by consumers and small businesses well before pushing into the enterprise space.
- All "as-a-service" offerings share a number of common attributes, including little or no capital expenditure since the required infrastructure is owned by the service provider, massive scalability, multitenancy, and device and location independence allowing consumers remote access to systems using nearly any current available technology.
- On the surface, it appears that XaaS is a potentially game-changing technology that could reshape IT.
- The concept of pay-as-you-go applications, development platforms, processing power, storage, or any other cloud-enabled services has emerged and can be expected to reshape IT over the next decade.
- Other concerns plague IT executives. They fear their data won't be safe in the hands of cloud providers and that they won't be able to manage cloud resources effectively. They may also worry that the new technology will threaten their own data centers and staff. Collectively, these fears tend to hold back the cloud computing market that some perceive growing to nearly $100 billion in the next decade.
- Although there is a significant benefit to leveraging cloud computing, security concerns have led organizations to hesitate to move critical resourcesto the cloud. Corporations and individuals are often concerned about how security and compliance integrity can be maintained in this new environment. Even more worrying, however, may be those corporations that are jumping into cloud computing that may be oblivious to the implications of putting critical applications and data in the cloud.
- Moving critical applications and sensitive data to public and shared cloud

environments is of great concern for those corporations that are moving beyond their data center's network perimeter defense. To alleviate these concerns, a cloud solution provider must ensure that customers will continue to have the same security and privacy controls over their applications and services, provide evidence to customers that their organization and customers are secure and they can meet their service-level agreements, and that they can prove compliance to auditors.

4.5 CLOUD SECURITY CHALLENGES

- Cloud security is a set of policies, procedures, and technologies used to protect cloud-based data, applications, and infrastructure from cyber threats. Cloud security is a shared responsibility between the cloud provider and the customer. The cloud provider is responsible for securing the underlying infrastructure, while the customer is responsible for securing their data and applications.
- Here is an overview of the key components of cloud security:
- Identity and access management (IAM): IAM controls who has access to cloud resources and what they can do with those resources.
- Data encryption: Data encryption protects data from unauthorized access, even if it is intercepted.
- Network security: Network security controls protect cloud resources from unauthorized access and malicious traffic.
- Application security: Application security controls protect cloud-based applications from vulnerabilities and attacks.
- Security monitoring and incident response: Security monitoring and incident response tools help to detect and respond to security threats in real time.
- In addition to these key components, cloud security also includes other important areas such as:
- Compliance: Compliance with industry regulations and standards is essential for many businesses. Cloud security solutions can help businesses to comply with these requirements.
- Risk management: Risk management is the process of identifying, assessing, and mitigating security risks. Cloud security solutions can help businesses to manage their security risks more effectively.

- Cloud security is essential for any business that is using cloud-based services. By implementing a comprehensive cloud security strategy, businesses can protect their data, applications, and infrastructure from cyber threats.
- Here are some tips for improving cloud security:
- Use strong passwords and multi-factor authentication for all cloud accounts.
- Implement least privilege access controls, so that users only have access to the resources they need.
- Encrypt all data at rest and in transit.
- Regularly back up your data and test your backup and restore procedures.
- Use a cloud security monitoring solution to detect and respond to security threats in real time.
- Keep your cloud software and applications up to date with the latest security patches.
- Security Issues in Cloud Computing :
  There is no doubt that Cloud Computing provides various Advantages but there are also some security issues in cloud computing. Below are some following Security Issues in Cloud Computing as follows.
- Data Loss –
  Data Loss is one of the issues faced in Cloud Computing. This is also known as Data Leakage. As we know that our sensitive data is in the hands of Somebody else, and we don't have full control over our database. So, if the security of cloud service is to break by hackers then it may be possible that hackers will get access to our sensitive data or personal files.

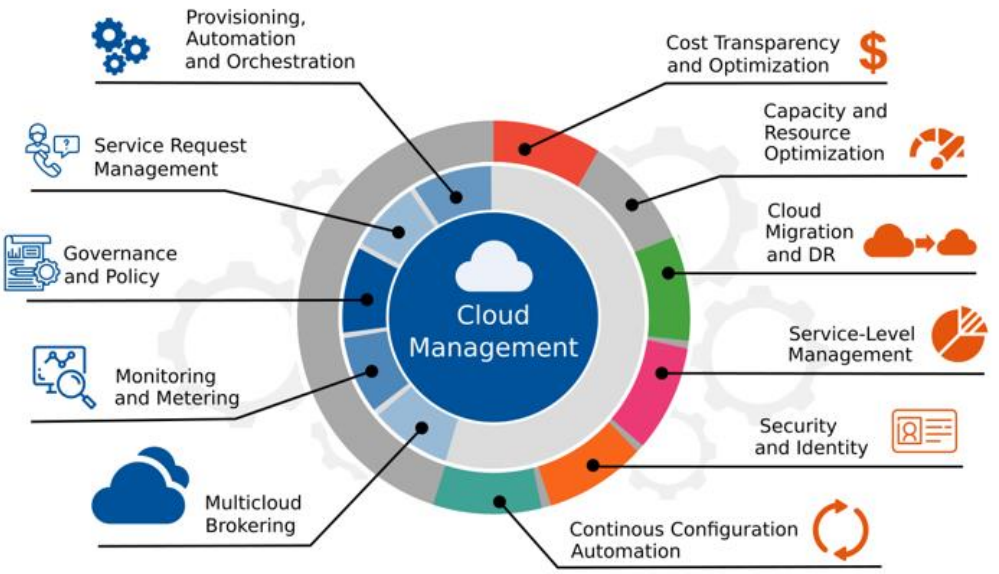
- Interference of Hackers and Insecure API's –
  As we know, if we are talking about the cloud and its services it means we are talking about the Internet. Also, we know that the easiest way to communicate with Cloud is using API. So it is important to protect the Interface's and API's which are used by an external user. But also in cloud computing, few services are available in the public domain which are the vulnerable part of Cloud Computing because it may be possible that these services are accessed by some third parties. So, it may be possible that with the help of these services hackers can easily hack or harm our data.

- User Account Hijacking –
  Account Hijacking is the most serious security issue in Cloud Computing. If somehow the Account of User or an Organization is hijacked by a hacker then the hacker has full authority to perform Unauthorized Activities.

- Changing Service Provider –
  Vendor lock-In is also an important Security issue in Cloud Computing. Many organizations will face different problems while shifting from one vendor to another. For example, An Organization wants to shift from AWS Cloud to Google Cloud Services then they face various problems like shifting of all data, also both cloud services have different techniques and functions, so they also face problems regarding that. Also, it may be possible that the charges of AWS are different from Google Cloud, etc.

- Lack of Skill –
  While working, shifting to another service provider, need an extra feature, how to use a feature, etc. are the main problems caused in IT Company who doesn't have skilled Employees. So it requires a skilled person to work with Cloud Computing.

- Denial of Service (DoS) attack –
  This type of attack occurs when the system receives too much traffic. Mostly DoS attacks occur in large organizations such as the banking sector, government sector, etc. When a DoS attack occurs, data is lost. So, in order to recover data, it requires a great amount of money as well as time to handle it.



- Although virtualization and cloud computing can help companies accomplish more by breaking the physical bonds between an IT infrastructure and its users, heightened security threats must be overcome in order to benefit fully from this new computing paradigm. This is particularly true for the SaaS provider. For example, in the cloud, you lose control over assets in some respects, so your security model must be reassessed. Enterprise security is only as good asthe least reliable partner, department, or vendor.
- With the cloud model, you lose control over physical security.
- In a public cloud, you are sharing computing resources with other companies.
- In a shared pool outside the enterprise, you don't have any knowledge or control of where the resources run. Exposing your data in an environment shared with other companies could give the government "reasonable cause" to seize your assets because another company has violated the law. Simply because you share the environment in the cloud, may put your data at risk of seizure.
- Storage services provided by one cloud vendor may be incompatible with another vendor's services should you decide to move from one to the other.
- Vendors are known for creating what the hosting world calls "sticky services"—services that an end user may have difficulty transporting from one cloud vendor to another (e.g., Amazon's "Simple Storage Service" [S3] is incompatible with IBM's Blue Cloud, or Google, or Dell).

- If information is encrypted while passing through the cloud, who controls the encryption/decryption keys? Is it the customer or the cloud vendor? Most customers probably want their data encrypted both ways across the Internet using SSL (Secure Sockets Layer protocol).
- They also most likely want their data encrypted while it is at rest in the cloud

vendor's storage pool. Be sure that you, the customer, control the encryption/decryption keys, just as if the data were still resident on your own servers.

- Data integrity means ensuring that data is identically maintained during any operation (such as transfer, storage, or retrieval). Put simply, data integrityis assurance that the data is consistent and correct. Ensuring the integrity of the data really means that it changes only in response to authorized transactions. This sounds good, but you must remember that a common standard to ensure data integrity does not yet exist.
- Using SaaS offerings in the cloud means that there is much less need for software development. For example, using a web- based customer relationship management (CRM) offering eliminates the necessity to write code and "customize" a vendor's application.
- If you plan to use internally developed code in the cloud, it is even more important to have a formal secure software development life cycle (SDLC).
- The immature use of mashup technology (combinations of web services), which is fundamental to cloud applications, is inevitably going to cause unwitting security vulnerabilities in those applications. Your development tool of choice should have a security model embedded in it to guide developers during the development phase and restrict users only to their authorized data when the system is deployed into production.
- As more and more mission-critical processes are moved to the cloud, SaaS suppliers will have to provide log data in a real-time, straightforward manner, probably for their administrators as well as their customers' personnel.
- Someone has to be responsible for monitoring for security and compliance, and unless the application and data are under the control of end users, they will not be able to.
- Will customers trust the cloud provider enough to push their mission-critical applications out to the cloud? Since the SaaS provider's logs are internal and not necessarily accessible externally or by clients or investigators, monitoring is difficult. Since access to logs is required for Payment Card Industry Data Security Standard (PCI DSS) compliance and may be requestedby auditors and regulators, security managers need to make sure to negotiate access to the provider's logs as part of any service agreement.

- Cloud applications undergo constant feature additions, and users must keep up to date with application improvements to be sure they are protected. The speed at which applications will change in the cloud will affect both the SDLC and security.
- For example, Microsoft's SDLC assumes that mission-critical software will have a three- to five-year period in which it will not change substantially, but the cloud may require a change in the application every few weeks.
- Even worse, a secure SLDC will not be able to provide a security cycle that keeps up with changes that occur so quickly. This means that users must constantly upgrade, because an older version may not function, or protect the data.
- Having proper fail-over technology is a component of securing the cloud that is often overlooked.
- The company can survive if a non-missioncritical application goes offline, but this may not be true for mission-critical applications. Core business practices provide competitive differentiation.
- Security needs to move to the data level, so that enterprises can be sure their data is protected wherever it goes. Sensitive data is the domain of the enterprise, not the cloud computing provider. One of the key challenges in cloud computing is data-level security.
- Most compliance standards do not envision compliance in a world of cloud computing. There is a huge body of standards that apply for IT security and compliance, governing most business interactions that will, over time, have to be translated to the cloud. SaaS makes the process of compliance more complicated, since it may be difficult for a customer to discern where its data resides on a network controlled by its SaaS provider, or a partner of that provider, which raises all sorts of compliance issues of data privacy, segregation, and security. Many compliance regulations require that data not be intermixed with other data, such as on shared servers or databases.
- Some countries have strict limits on what data about its citizens can be stored and for how long, and some banking regulators require that customers' financial data remain in their home country.
- Compliance with government regulations such as the Sarbanes-Oxley Act (SOX), the Gramm-Leach-Bliley Act (GLBA), and the Health Insurance

Portability and Accountability Act (HIPAA), and industry standards such as the PCI DSS, will be much more challenging in the SaaS environment.

- There is a perception that cloud computing removes data compliance responsibility; however, it should be emphasized that the data owner is still fully responsible for compliance. Those who adopt cloud computing must remember that it is the responsibility of the data owner, not the service provider, to secure valuable data.

- Government policy will need to change in response to both the opportunity and the threats that cloud computing brings. This will likely focus on the off-shoring of personal data and protection of privacy, whether it is data

  being controlled by a third party or off-shored to another country. There will be a corresponding drop in security as the traditional controls such as VLANs (virtual local-area networks) and firewalls prove less effective during the transition to a virtualized environment.

- Security managers will need to pay particular attention to systems that contain critical data such as corporate financial information or source code duringthe transition to server virtualization in production environments.

- Outsourcing means losing significant control over data, and while this isn't a good idea from a security perspective, the business ease and financial savings will continue to increase the usage of these services.

- Security managers will need to work with their company's legal staff to ensure that appropriate contract terms are in place to protect corporate data and provide for acceptable service-level agreements.

- Cloud-based services will result in many mobile IT users accessing business data and services without traversing the corporate network. This will increase the need for enterprises to place security controls between mobile users and cloud-based services.

- Placing large amounts of sensitive data in a globally accessible cloud leaves organizations open to large distributed threats—attackers no longer have to come onto the premises to steal data, and they can find it all in the one "virtual" location.

- Virtualization efficiencies in the cloud require virtual machines from multiple organizations to be co-located on the same physical resources. Although traditional data center security still applies in the cloud environment, physical

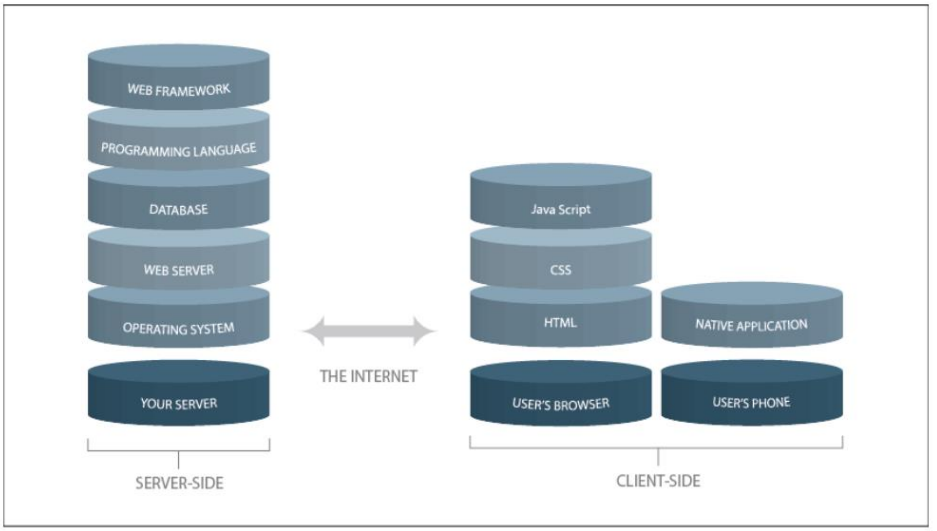
segregation and hardware-based security cannot protect against attacks between virtual machines on the same server.

- Administrative access is through the Internet rather than the controlled and restricted direct or on-premises connection that is adhered to in the traditional data center model.
- This increases risk and exposure and will require stringent monitoring for changes in system control and access control restriction.
- The dynamic and fluid nature of virtual machines will make it difficult to maintain the consistency of security and ensure the auditability of records. The ease of cloning and distribution between physical servers could result in the propagation of configuration errors and other vulnerabilities.
- Proving the security state of a system and identifying the location of an insecure virtual machine will be challenging. Regardless of the location of the virtual machine within the virtual environment, the intrusion detection and prevention systems will need to be able to detect malicious activity at virtual machine level.
- The co-location of multiple virtual machines increases the attack surface and risk of virtual machine-to-virtual machine compromise.
- Localized virtual machines and physical servers use the same operating systems as well as enterprise and web applications in a cloud server environment, increasing the threat of an attacker or malware exploiting vulnerabilities in these systems and applications remotely.
- Virtual machines are vulnerable as they move between the private cloud and the public cloud. A fully or partially shared cloud environment is expected to have a greater attack surface and therefore can be considered to be at greater risk than a dedicated resources environment.
- Operating system and application files are on a shared physical infrastructure in a virtualized cloud environment and require system, file, and activity monitoring to provide confidence and auditable proof to enterprise customers that their resources have not been compromised or tampered with.
- In the cloud computing environment, the enterprise subscribes to cloud computing resources, and the responsibility for patching is the subscriber's rather than the cloud computing vendor's. The need for patch maintenance vigilance is imperative.

- Lack of due diligence in this regard could rapidly make the task unmanageable or impossible, leaving you with "virtual patching" as the only alternative.
- Enterprises are often required to prove that their security compliance is in accord with regulations, standards, and auditing practices, regardless of the location of the systems at which the data resides.
- Data is fluid in cloud computing and may reside in on-premises physical servers, on-premises virtual machines, or off-premises virtual machines running on cloud computing resources, and this will require some rethinking on the part of auditors and practitioners alike.
- In the rush to take advantage of the benefits of cloud computing, not least of which is significant cost savings, many corporations are likely rushing into cloud computing without a serious consideration of the security implications.
- To establish zones of trust in the cloud, the virtual machines must be self-defending, effectively moving the perimeter to the virtual machine itself.
- Enterprise perimeter security (i.e., firewalls, demilitarized zones [DMZs], network segmentation, intrusion detection and prevention systems [IDS/ IPS], monitoring tools, and the associated security policies) only controls the data that resides and transits behind the perimeter.
- In the cloud computing world, the cloud computing provider is in charge of customer data security and privacy.

4.6 SOFTWARE-AS-A-SERVICE SECURITY

- SaaS security is the managing, monitoring, and safeguarding of sensitive data from cyber-attacks. With the increase in efficiency and scalability of cloud-based IT infrastructures, organizations are also more vulnerable.
- SaaS maintenance measures such as SaaS security posture management ensure privacy and safety of user data. From customer payment information to inter-departmental exchange of information, strengthening the security of SaaS applications is vital to your success.



- Every organization offering a cloud-based service can leverage preventive measures such as SaaS security posture management to continuously monitor and protect sensitive information.
- Let us understand the anatomy of SaaS security in cloud computing environments. If we look at an ideal SaaS product technology stack from a bird's eye view, it forms a three-layer cake where each part represents different environments.
- Three layers of SaaS security:
- Infrastructure (server-side)
- Network (the internet)
- Application and Software (client-side)

Challenges in SaaS security

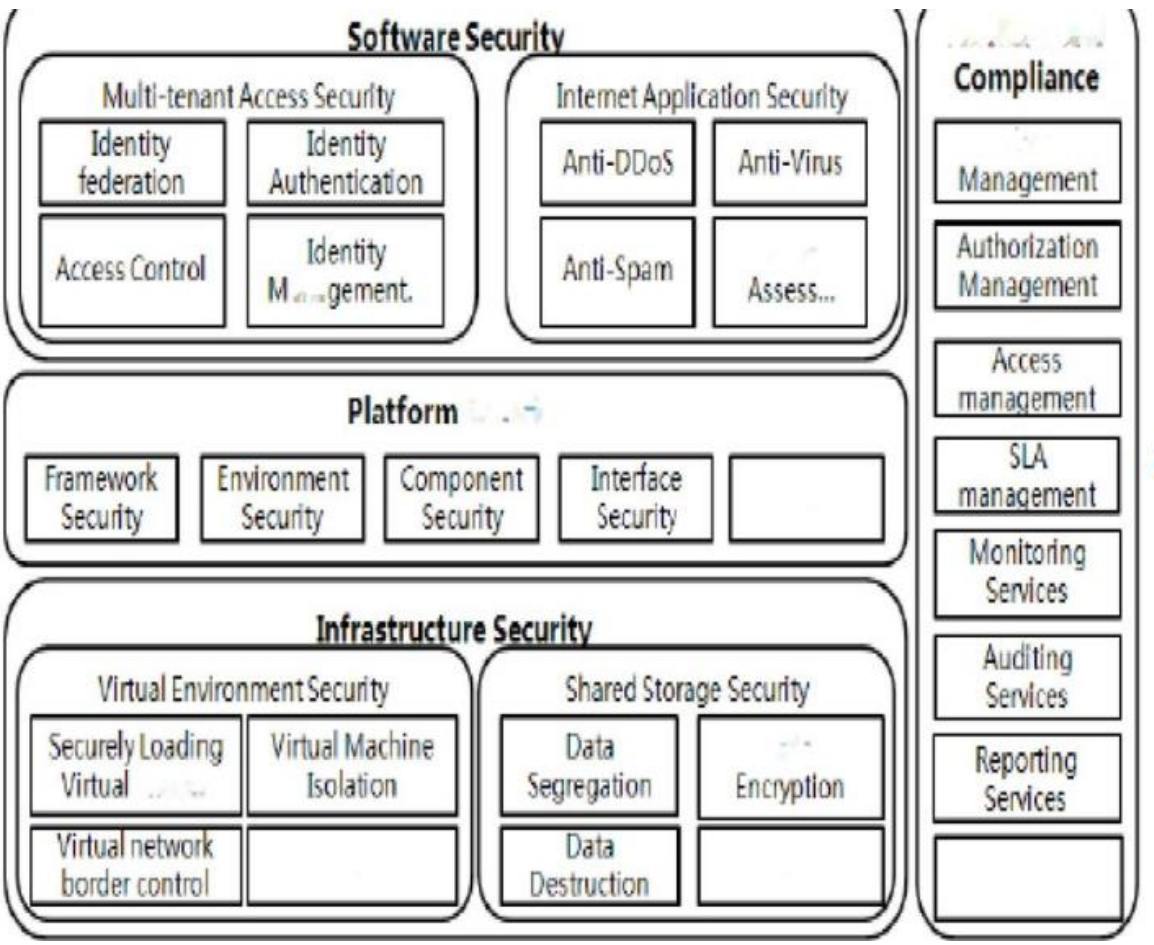
- Some of the most significant challenges in SaaS security include:
- 1. Lack of Control
- SaaS providers typically host applications and data in the cloud, meaning that customers have less direct control over their security. This can make it challenging for customers to monitor and manage security effectively.
- 2. Access Management

- SaaS applications typically require users to log in and authenticate their identity. However, managing user access can be challenging, particularly if the provider is hosting applications for multiple customers with different access requirements.
- 3. Data Privacy
- SaaS providers may be subject to data privacy regulations, which can vary by jurisdiction. This can make it challenging to ensure compliance with all relevant laws and regulations, particularly if the provider hosts data for customers in multiple countries.
- 4. Third-party integration
- SaaS providers may integrate with third-party applications, such as payment processors or marketing platforms. However, this can increase the risk of security incidents, as vulnerabilities in third-party software can potentially affect the entire system.
- 5. Continuous monitoring
- SaaS providers must continuously monitor their systems for security threats and vulnerabilities. This requires a high level of expertise and resources to detect and respond to security incidents effectively.

- Cl**oud SaaS security solutions**
- Several types of security solutions can help organizations improve SaaS security. The solutions can be implemented separately or together as part of a CASB:
- Data loss prevention (DLP) safeguards intellectual property and protects sensitive data in cloud applications, as well as at endpoints such as laptops. Organizations can define data access policies that DLP enforces.
- Compliance solutions provide controls and reporting capabilities to ensure compliance with government and industry regulations.
- Advanced malware prevention includes technologies such as behavioral analytics and real-time threat intelligence that can help detect and block zero-day attacks and malicious files that may be spread through cloud email and file sharing applications.

- CASBs protect enterprise data and users across all cloud services, including SaaS, PaaS, and IaaS. According to Gartner's Magic Quadrant for Cloud Access Security Brokers, CASBs detect threats and provide IT departments with greater visibility into data usage and user behavior for cloud services, end users, and devices. CASBs also act immediately to remediate security threats by eliminating security misconfigurations and correcting high-risk user activities applications. CASBs provide a variety of security services, including:
- Monitoring for unauthorized cloud services
- Enforcing data security policies including encryption
- Collecting details about users who access data in cloud services from any device or location
- Restricting access to cloud services based on the user, device, and application
- Providing compliance reporting
- CASB solutions, which are typically SaaS applications, may provide additional capabilities. These may include:
- File encryption
- Pre-built policy templates to guide IT staff through the process of policy creation
- User entity behavior analytics (UEBA) backed by machine learning
- In-application coaching to help end users learn improved security practices
- Security configuration audits to suggest changes to security settings based on best practices
- IT departments can learn to protect their cloud applications and data by following cloud security best practices and implementing effective SaaS security solutions. Cloud security solutions from Skyhigh Security enable organizations to accelerate their business growth by giving them visibility and control over their applications, devices, and data.

- Computing models of the future will likely combine the use of SaaS (andother XaaS's as appropriate), utility computing, and Web 2.0 collaboration technologies to leverage the Internet to satisfy their customers' needs.
- New business models being developed as a result of the move to cloud computing are creating not only new technologies and business operational processes but also new security requirements and challenges as described previously.
- As the most recent evolutionary step in the cloud service model (see Figure 4.8), SaaS will likely remain the dominant cloud service model for the foreseeable future and the area where the most critical need for security practices and oversight will reside.



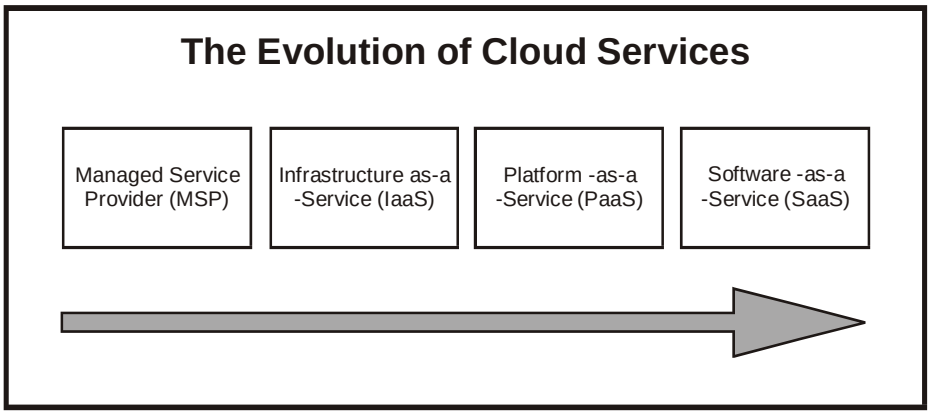


Figure 4.8 The evolution of cloud services.

- Just as with an managed service provider, corporations or end users will need to research vendors' policies on data security before using vendor services to avoid losing or not being able to access their data.
- The technology analyst and consulting firm Gartner lists seven security issues which one should discuss with a cloud-computing vendor:

1. **Privileged user access** —Inquire about who has specialized access to data, and about the hiring and management of such administrators.
2. **Regulatory compliance** —Make sure that the vendor is willing to undergo external audits and/or security certifications.
3. **Data location** —Does the provider allow for any control over the location of data?
4. **Data segregation** —Make sure that encryption is available at all stages, and that these encryption schemes were designed and tested by experienced professionals.
5. **Recovery** —Find out what will happen to data in the case of a disaster. Do they offer complete restoration? If so, how long would that take?
6. **Investigative support** —Does the vendor have the ability to investigate any inappropriate or illegal activity?
7. **Long-term viability** —What will happen to data if the company goes out of business? How will data be returned, and in what format?

- Determining data security is harder today, so data security functions have become more critical than they have been in the past.
- A tactic not covered by Gartner is to encrypt the data yourself. If you encrypt the data using a trusted algorithm, then regardless of the service provider's security and encryption policies, the data will only be accessible with the decryption keys.
- Of course, this leads to a follow-on problem: How do you manage private keys in a pay-on-demand computing infrastructure?

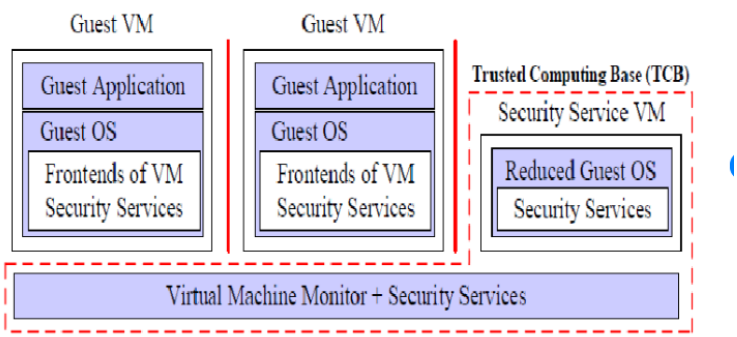
4.6.1 Security Governance

- Security governance is the framework of policies, processes, and tools that an organization uses to manage its security posture. It is a critical component of any organization's overall risk management strategy.
- Security governance helps organizations to:
- Align their security posture with their business goals and objectives.
- Identify and assess their security risks.
- Implement and maintain appropriate security controls.
- Monitor and improve their security posture over time.
- Security governance is especially important for organizations that are using cloud computing services. Cloud computing can offer a number of security benefits, but it also introduces a number of new security risks. Security governance can help organizations to manage these risks and protect their data and applications in the cloud.
- Here are some key elements of a cloud security governance framework:
- Risk management: The organization should have a process for identifying, assessing, and mitigating security risks.
- Compliance: The organization should ensure that its cloud environment complies with all applicable regulations and standards.
- Identity and access management (IAM): The organization should have a robust IAM system in place to control who has access to cloud resources and what they can do with them.

- Data security: The organization should have a plan for protecting its data in the cloud, both at rest and in transit.
- Network security: The organization should have a plan for protecting its network from unauthorized access and malicious traffic.
- Application security: The organization should have a plan for securing its cloud-based applications from vulnerabilities and attacks.
- Security monitoring and incident response: The organization should have a system in place for monitoring its cloud environment for security threats and responding to incidents in a timely manner.
- Security governance is an ongoing process. Organizations should regularly review and update their security governance framework to ensure that it is effective in managing their security risks.
- Here are some tips for implementing effective cloud security governance:
- Get executive buy-in: Security governance must be supported by senior management. This will help to ensure that the necessary resources are allocated to security and that security is given the appropriate priority.
- Involve all stakeholders: Security governance is not just the responsibility of the IT department. All stakeholders, including business leaders, users, and vendors, should be involved in the development and implementation of the security governance framework.
- Use a risk-based approach: Security governance should be based on a risk assessment. This will help to ensure that resources are allocated to the areas where they are most needed.
- Adopt a continuous improvement approach: Security governance should be an ongoing process. Organizations should regularly review and update their security governance framework to ensure that it is effective in managing their security risks.
- By following these tips, organizations can implement effective cloud security governance and protect their data and applications in the cloud.

- A security steering committee should be developed whose objective is to focus on providing guidance about security initiatives and alignment with business and IT strategies.
- Lack of a formalized strategy can lead to an unsustainable operating model and security level as it evolves. In addition, lack of attention to security governance can result in key needs of the business not being met, including but not limited to, risk management, security monitoring, application security, and sales support.
- Lack of proper governance and management of duties can also result in potential security risks being left unaddressed and opportunities to improve the business being missed because the security team is not focused on the key security functions and activities that are critical to the business.

4.6.2 Virtual Machine Security



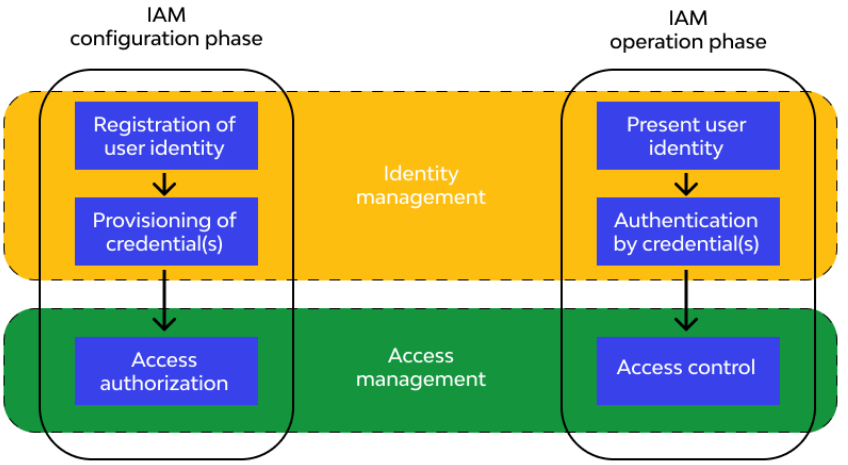
- In the cloud environment, physical servers are consolidated to multiple virtual machine instances on virtualized servers.
- Not only can data center security teams replicate typical security controls for

the data center at large to secure the virtual machines, they can also advise their customers on how to prepare these machines for migration to a cloud environment when appropriate.

- Firewalls, intrusion detection and prevention, integrity monitoring, and log inspection can all be deployed as software on virtual machines to increase protection and maintain compliance integrity of servers and applications as virtual resources move from on-premises to public cloud environments.
- By deploying this traditional line of defense to the virtual machine itself, you can enable critical applications and data to be moved to the cloud securely.
- To facilitate the centralized management of a server firewall policy, the security software loaded onto a virtual machine should include a bidirectional stateful firewall that enables virtual machine isolation and location awareness, thereby enabling a tightened policy and the flexibility to move the virtual machine from on-premises to cloud resources.
- Integrity monitoring and log inspection software must be applied at the virtual machine level.
- This approach to virtual machine security, which connects the machine back to the mother ship, has some advantages in that the security software can be put into a single software agent that provides for consistent control and management throughout the cloud while integrating seamlessly back into existing security infrastructure investments, providing economies of scale, deployment, and cost savings for both the service provider and the enterprise.

4.6.3 Identity Access Management (IAM)

- **Identity and Access Management (IAM) lets administrators authorize who can take action on specific resources, giving you full control and visibility to manage Google Cloud resources centrally.**



User Identity Management

- With IAM, one can keep track of user identity for various purposes. It allows businesses to generate, modify, and eliminate user information as per the need of the hours. Additionally, end-users are allowed to integrate multiple user directories and synchronize them with each other. Conditional user identities can also be generated with the help of IAM.
- **Provisioning or de-provisioning of the users**
- Based upon the resource accessibility, certain tools should only be used by certain professionals. Granting such conditioned access to tools is called provisioning and IAM makes this possible. Provisioning can be done as per roles, departments, and groups within a company or institution. When such conditioned access is not required, IAM makes de-provisioning possible too.
- **Authenticating users at various levels**
- IAM is one of the safest ways to define a user and confirm their identity. It's done using adaptive or multi-factor authentication (MFA) standards.
- **User Authorization**
- It's possible to grant access to a certain tool to a certain professional by deploying access control for system users. Depending upon the needs, users can be distributed into groups and grant privileges accordingly.
- **Operational reporting**

- IAM systems are capable of generating reports on key actions taken over a single click and let you complete the security compliances and keep security risks under control.
- **Implementing single sign-on**
- In SSO, a user is authorized in such a way that its access from a single portal could be confirmed. Some of the IAM solutions come with integrated SSO features/tools to make the process easier for you.
-
- How Does IAM Work?
- The functionality of IAM depends on confirming the humans, hardware, and virtual applications for their identity. It uses IAM systems checks to see if the user/resource is authorized to complete a certain task or enter a system. Once the users and other elements are defined, IAM grants access as per the defined rules.
- Types Of User Authentication
- IAM uses extensive user identity verification methods and gives full freedom to the end-user regarding their implementation. Here is a detailed explanation of key user authentication procedures.
- Multi-factor Authentication
- The Procedure involves combining two or more verification procedures and login credentials together. So, when a user requests access to a particular tool and software, he will have to enter a code after login into a resource.
- Unique Passwords
- This method is mostly adopted by users for secured access. It involves using safe and tough passwords for protecting the resource. Usually, a unique combination of digits, characters, and special characters.
- Pre-shared Key (PSK)
- PSK or Pre-shared Key is a kind of digital authentication wherein a unique password is shared only with authorized personals. The most common example of PSK is sharing Wi-Fi passwords with the team. It is not as safe as individual passwords.

- Behavioral Authentication
- It involves AI to figure out that the user trying to gain access to a particular resource is a human or machine. The process checks if the behavior is appropriate or not. For this, it analyzes granular characteristics, such as mouse-use tracking. Once the IAM figures out that something is not right, the system gets closed automatically.
- Biometrics
- Mostly used and managed by high-end IAM systems, certain biometrics traits are matched in this type of identity verification or authentication method.
- Based upon the needs, the biometrics components used in this kind of authentication are the face and finger authentication, iris and voice recognition, and even DNA matching.
- It is considered a highly reliable and dependable method for identity verification, but there is a catch here.
- As the gathered information is sensitive, companies with behavioral authentication deployed for their IT infrastructure must use high-end data safety methods, keep the process utterly transparent, and optional for the users.
- Identity And Access Management Tools
- IAM takes the help of assorted tools to accomplish its purpose. The key tools used by an IAM system are a security policy tool, password management applications, reporting & monitoring tools, provisioning software, and identity repositories.
- Other than this, MFA is another crucial tool assisting IAM. It deals with 2 or more access control related factors.
- Single sign-on (SSO) is also a widely used IAM tool that allows users to sign in only once and provide a centralized platform to manage all sorts of access.
- IAM Technologies
- The primary capabilities of an IAM system depends on the use of technology that allows a system to integrate with different systems. The IAM technologies are bound to adhere to certain standards to make sure everything has complied. Here are key technologies and standards adopted for IAM

framework:

SAML (Security Access Markup Language)

- It is an open standard that is most widely utilized for exchanging identity approval-related details in a proper manner. This information exchange happens between an identity provider and a service/application. IAM systems often make use of it to ensure a secure login to an application.

OpenID Connect (OIDC)

- OIDC is a relatively new standard allowing end-users to gain instant access to the resource from an identity provider. On the surface, it looks similar to the SAML. But, it's different as it is built using the OAuth 2.0 standards and takes the help of JSON for data transmission.

System for Cross-domain Identity Management (SCIM)

- This technology is called up for help to do quick and effective identity information exchange between two different systems. It's true that SAML and OIDC are capable of sharing the information independently but SCIM is used to keep every piece of information updated during the deployment of new users. Because of this technology, user data is always updated in an IAM system.

-

- AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access.

- AM stands for Identity Access Management. IAM allows you to manage users and their level of access to the aws console. It is used to set users, permissions and roles. It allows you to grant access to the different parts of the aws platform.

- IAM roles are of 4 types, primarily differentiated by who or what can assume the role: Service Role. Service-Linked Role. Role for Cross-Account Access.

- Identity and Access Management (IAM) is a combination of policies and technologies that allows organizations to identify users and provide the right form of access as and when required. There has been a burst in the market with new applications, and the requirement for an organization to use these applications has increased drastically. The services and resources you want to access can be specified in IAM. IAM doesn't provide any replica or backup. IAM can be used for many purposes such as, if one want's to control access of individual and group access for your AWS resources. With IAM policies, managing permissions to your workforce and systems to ensure least-privilege permissions becomes easier. The AWS IAM is a global service.

- Components of IAM

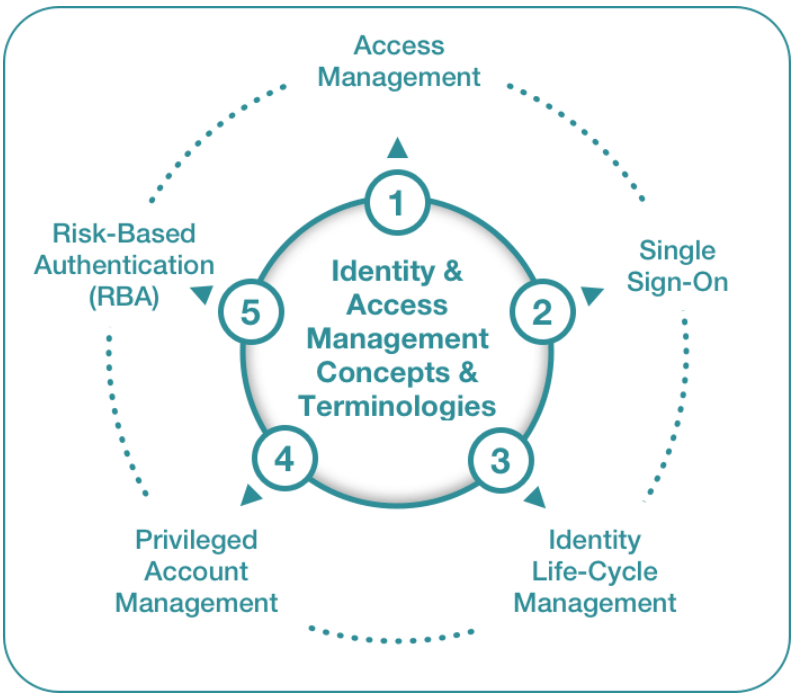
- Users

- Roles

- Groups

- Policies

- With these new applications being created over the cloud, mobile and on-premise can hold sensitive and regulated information. It's no longer acceptable and feasible to just create an Identity server and provide access based on the requests. In current times an organization should be able to track the flow of information and provide least privileged access as and when required, obviously with a large workforce and new applications being added every day it becomes quite difficult to do the same. So organizations specifically concentrate on managing identity and its access with the help of a few IAM tools. It's quite obvious that it is very difficult for a single tool to manage everything but there are multiple IAM tools in the market that help the organizations with any of the few services given below.



- Identity and access management is a critical function for every organization, and a fundamental expectation of SaaS customers is that the principle of least privilege is granted to their data.
- The principle of least privilege states that only the minimum access necessary to perform an operation should be granted, and that access should be granted only for the minimum amount of time necessary.
- However, business and IT groups will need and expect access to systems and applications.
- The advent of cloud services and services on demand is changing the identity management landscape.
- Most of the current identity management solutions are focused on the enterprise and typically are architected to work in a very controlled, static environment.
- User-centric identity management solutions such as federated identity management, also make some assumptions about the parties involved and their

related services.

In the cloud environment, where services are offered on demand and they can continuously evolve, aspects of current models such as trust assumptions, privacy implications, and operational aspects of authentication and authorization, will be challenged.

- Meeting these challenges will require a balancing act for SaaS providers as they evaluate new models and management processes for IAM to provide end-to-end trust and identity throughout the cloud and the enterprise.
- Another issue will be finding the right balance between usability and security. If a good balance is not achieved, both business and IT groups may be affected by barriers to completing their support and maintenance activities efficiently.

4.7 STANDARDS FOR SECURITY

- Security standards define the processes, procedures, and practices necessary for implementing a security program.
- These standards also apply to cloud related IT activities and include specific steps that should be taken to ensure a secure environment is maintained that provides privacy and security of confidential information in a cloud environment.
- Security standards are based on a set of key principles intended to protect this type of trusted environment.
- Messaging standards, especially for security in the cloud, must also include nearly all the same considerations as any other IT security endeavor.

4.7.1 Security (SAML OAuth, OpenID, SSL/TLS)

- A basic philosophy of security is to have layers of defense, a concept known as *defense in depth*. This means having overlapping systems designed to provide security even if one system fails.
- An example is a firewall working in conjunction with an intrusion-detection system (IDS). Defense in depth provides security because there is no single point of failure and no single entry vector at which an attack can occur. For this reason, a choice between implementing network security in the middle part of a network (i.e., in the cloud) or at the endpoints is a false dichotomy.
- No single security system is a solution by itself, so it is far better to secure all systems. This type of layered security is precisely what we are seeing develop in cloud computing.
- Traditionally, security was implemented at the endpoints, where the user controlled access. An organization had no choice except to put firewalls, IDSs, and antivirus software inside its own network.
- Today, with the advent of managed security services offered by cloud providers, additional security can be provided inside the cloud.

4.7.1.1 Security Assertion Markup Language (SAML)

- SAML is an XML-based standard for communicating authentication, authorization, and attribute information among online partners.
- It allows businesses to securely send assertions between partner organizations regarding the identity and entitlements of a principal.
- The Organization for the Advancement of Structured Information Standards (OASIS) Security Services Technical Committee is in charge of defining, enhancing, and maintaining the SAML specifications.9 SAML is built on a number of existing standards, namely, SOAP, HTTP, and XML. SAML relies on HTTP as its communications protocol and specifies the use of SOAP (currently, version 1.1).
- Most SAML transactions are expressed in a standardized form of XML. SAML assertions and protocols are specified using XML schema.
- Both SAML 1.1 and SAML 2.0 use digital signatures (based on the XML Signature standard) for authentication and message integrity.

- XML encryption is supported in SAML 2.0, though SAML 1.1 does not have encryption capabilities. SAML defines XML-based assertions and protocols, bindings, and profiles.
- The term SAML Core refers to the general syntax and semantics of SAML assertions as well as the protocol used to request and transmit those assertions from one system entity to another. SAML protocol refers to what is transmitted, not how it is transmitted.
- A SAML binding determines how SAML requests and responses map to standard messaging protocols. An important (synchronous) binding is the SAML SOAP binding.
- SAML standardizes queries for, and responses that contain, user authentication, entitlements, and attribute information in an XML format.
- This format can then be used to request security information about a principal from a SAML authority. A SAML authority, sometimes called the asserting party, is a platform or application that can relay security information.
- The relying party (or assertion consumer or requesting party) is a partner site that receives the security information.
- The exchanged information deals with a subject's authentication status, access authorization, and attribute information.
- A subject is an entity in a particular domain. A person identified by an email address is a subject, as might be a printer.
- SAML assertions are usually transferred from identity providers to service providers. Assertions contain statements that service providers use to make access control decisions.
- Three types of statements are provided by SAML: authentication statements, attribute statements, and authorization decision statements.
- SAML assertions contain a packet of security information in this form:

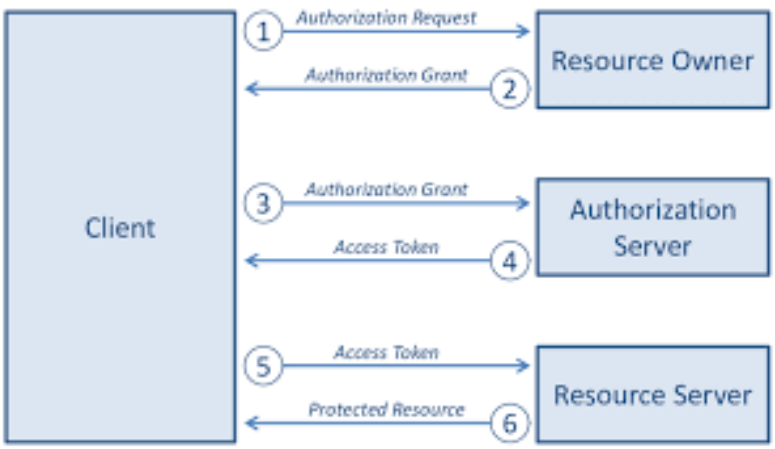
```
<saml:Assertion A...>
<Authentication>
...
</Authentication>
<Attribute>
```

...
</Attribute>
<Authorization>
...
</Authorization>
</saml:Assertion A>

- ws: Assertion A, issued at time T by issuer I, regarding subject S, provided conditions C are valid.
- Authentication statements assert to a service provider that the principal did indeed authenticate with an identity provider at a particular time using a particular method of authentication.
- Other information about the authenticated principal (called the authentication context) may be disclosed in an authentication statement.
- An attribute statement asserts that a subject is associated with certain attributes. An attribute is simply a name–value pair.
- Relying parties use attributes to make access control decisions. An authorization decision statement asserts that a subject is permitted to perform action A on resource R given evidence E. The expressiveness of authorization decision statements in SAML is intentionally limited.
- A SAML protocol describes how certain SAML elements (including assertions) are packaged within SAML request and response elements.
- It provides processing rules that SAML entities must adhere to when using these elements. Generally, a SAML protocol is a simple request–response protocol.
- The most important type of SAML protocol request is a query. A service provider makes a query directly to an identity provider over a secure back channel. For this reason, query messages are typically bound to SOAP.
- Corresponding to the three types of statements, there are three types of SAML queries: the authentication query, the attribute query, and the authorization decision query.
- Of these, the attribute query is perhaps most important. The result of an attribute query is a SAML response containing an assertion, which itself contains an attribute statement.

4.7.1.2 Open Authentication (OAuth)

- OAuth is an open protocol, initiated by Blaine Cook and Chris Messina, to allow secure API authorization in a simple, standardized method for various types of web applications.
- OAuth doesn't share password data but instead uses authorization tokens to prove an identity between consumers and service providers. OAuth is an authentication protocol that allows you to approve one application interacting with another on your behalf without giving away your password.
- OAuth (Open Authorization) is an open standard protocol for authorization of an application for using user information, in general, it allows a third party application access to user related info like name, DOB, email or other required data from an application like Facebook, Google
- There are 3 Components in OAuth Mechanism–
- OAuth Provider – This is the OAuth provider Eg. Google, FaceBook etc.
- OAuth Client – This is the website where we are sharing or authenticating the usage of our information. Eg. GeeksforGeeks etc.
- Owner – The user whose login authenticates sharing of information.
- OAuth can be implemented via google console for “Login/Sign Up with Google” on a web app.
  Pattern to be Followed –
- Get OAuth 2.0 Client ID from Google API Console
- Next, Obtain an access token from the Google Authorization Server to access the API.
- Send the request with the access token to an API .
- Get Refresh token if longer access is required. etc



- Cook and Messina had concluded that there were no open standards for API access delegation.
- The OAuth discussion group was created in April 2007, for the small group of implementers to write the draft proposal for an open protocol.
- DeWitt Clinton of Google learned of the OAuth project and expressed interest in supporting the effort.
- In July 2007 the team drafted an initial specification, and it was released in October of the same year.
- OAuth is a method for publishing and interacting with protected data.
- For developers, OAuth provides users access to their data while protecting account credentials. OAuth allows users to grant access to their information, which is shared by the service provider and consumers without sharing all of their identity.
- The Core designation is used to stress that this is the baseline, and other extensions and protocols can build on it.
- By design, OAuth Core 1.0 does not provide many desired features (e.g., automated discovery of endpoints, language support, support for XML-RPC and SOAP, standard definition of resource access, OpenID integration, signing algorithms, etc.).
- This intentional lack of feature support is viewed by the authors as a significant benefit. The Core deals with fundamental aspects of the protocol, namely, to establish a mechanism for exchanging a user name and password for a token with defined rights and to provide tools to protect the token.
- It is important to understand that security and privacy are not guaranteed by the protocol.
- In fact, OAuth by itself *provides no privacy at all* and depends on other protocols such as SSL to accomplish that. OAuth can be implemented in a secure manner, however.
- In fact, the specification includes substantial security considerations that must be taken into account when working with sensitive data.

- With Oauth, sites use tokens coupled with shared secrets to access resources. Secrets, just like passwords, must be protected.

4.7.1.3 OpenID

- OpenID is an open, decentralized standard for user authentication and access control that allows users to log onto many services using the same digital identity. It is a single-sign-on (SSO) method of access control.
- OpenID is an open specification for authentication and single sign-on (SSO). OpenID, which was first created in 2005, allows web sites and authentication services to exchange security information in a standardized way.
- As such, it replaces the common log-in process (i.e., a log-in name and a password) by allowing users to log in once and gain access to resources across participating systems.
- The original OpenID authentication protocol was developed in May 2005by Brad Fitzpatrick, creator of the popular community web site Live- Journal.
- In late June 2005, discussions began between OpenID developers and other developers from an enterprise software company named Net-Mesh.
- These discussions led to further collaboration on interoperability between OpenID and NetMesh's similar Light-Weight Identity (LID) protocol.
- The direct result of the collaboration was the Yadis discovery protocol, which was announced on October 24, 2005.
- The Yadis specification provides a general-purpose identifier for a person and any other entity, which can be used with a variety of services.
- It provides a syntax for a resource description document identifying services available using that identifier and an interpretation of the elements of that document.
- Yadis discovery protocol is used for obtaining a resource description document, given that identifier.
- Together these enable coexistence and interoperability of a rich variety of services using a single identifier. The identifier uses a standard syntax and a well- established namespace and requires no additional namespace administration infrastructure.

- An OpenID is in the form of a unique URL and is authenticated by the entity hosting the OpenID URL. The OpenID protocol does not rely on a central authority to authenticate a user's identity. Neither the OpenID protocol nor any web sites requiring identification can mandate that a specific type of authentication be used; nonstandard forms of authentication such as smart cards, biometrics, or ordinary passwords are allowed.
- A typical scenario for using OpenID might be something like this: A user visits a web site that displays an OpenID log-in form somewhere on the page. Unlike a typical log-in form, which has fields for user name and password, the OpenID log-in form has only one field for the OpenID identifier (which is an OpenID URL).
- This form is connected to an implementation of an OpenID client library. A user will have previously registered an OpenID identifier with an OpenID identity provider. The user types this OpenID identifier into the OpenID log-in form.
- The relying party then requests the web page located at that URL and reads an HTML link tag to discover the identity provider service URL. With OpenID 2.0, the client discovers the identity provider service URL by requesting the XRDS document (also called the Yadis document) with the content type **application/xrds+xml,** which may be available at the target URL but is always available for a target XRI.
- There are two modes by which the relying party can communicate with the identity provider: **checkid_immediate** and **checkid_setup.**
- In **checkid_immediate,** the relying party requests that the provider not interact with the user. All communication is relayed through the user's browser without explicitly notifying the user.
- In **checkid_setup,** the user communicates with the provider server directly using the same web browser as is used to access the relying party site. The second option is more popular on the web.
- To start a session, the relying party and the identity provider establish a shared secret—referenced by an associate handle—which the relying party then stores.
- Using **checkid_setup,** the relying party redirects the user's web browser to the identity provider so that the user can authenticate with the provider.

- The method of authentication varies, but typically, an OpenID identity provider prompts the user for a password, then asks whether the user trusts the relying party web site to receive his or her credentials and identity details.
- If the user declines the identity provider's request to trust the relying party web site, the browser is redirected to the relying party with a message indicating that authentication was rejected. The site in turn refuses to authenticate the user.
- If the user accepts the identity provider's request to trust the relying party web site, the browser is redirected to the designated return page on the relyingparty web site along with the user's credentials.
- That relying party must then confirm that the credentials really came from the identity provider. If they had previously established a shared secret, the relying party can validate the shared secret received with the credentials against the one previously stored. In this case, the relying party is considered to be stateful, because it stores the shared secret between sessions (a process sometimes referred to as persistence).
- In comparison, a stateless relying party must make background requests using the **check_authentication** method to be sure that the data came from the identity provider.
- After the OpenID identifier has been verified, OpenID authentication is considered successful and the user is considered logged in to the relying party web site.
- The web site typically then stores the OpenID identifier in the user's session. OpenID does not provide its own authentication methods, but if an identity provider uses strong authentication, OpenID can be used for secure transactions.
- **Single sign-on (SSO)** is an important cloud security technology that reduces all user application logins to one login for greater security and convenience. A single sign-on solution can simplify username and password management for both users and administrators. Users no longer have to keep track of different sets of credentials and can simply remember a single more complex password. SSO often enables users to just get access to their applications much faster.

4.7.1.4 SSL/TLS

- Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographically secure protocols designed to provide security and data integrity for communications over TCP/IP.
- TLS and SSL encrypt the segments of network connections at the transport layer. Several versions of the protocols are in general use in web browsers, email, instant messaging, and voice-over-IP. TLS is an IETF standard protocol which was last updated in RFC 5246.
- The TLS protocol allows client/server applications to communicate across a network in a way specifically designed to prevent eavesdropping, tampering, and message forgery.
- TLS provides endpoint authentication and data confidentiality by using cryptography. TLS authentication is one way—the server is authenticated, because the client already knows the server's identity.
- In this case, the client remains unauthenticated. At the browser level, this means that the browser has validated the server's certificate— more specifically, it has checked the digital signatures of the server certificate's issuing chain of Certification Authorities (CAs).
- Validation does not identify the server to the end user. For true identification, the end user must verify the identification information contained in the server's certificate (and, indeed, its whole issuing CA chain).
- This is the only way for the end user to know the "identity" of the server, and this is the only way identity can be securely established, verifying thatthe URL, name, or address that is being used is specified in the server's certificate.
- Malicious web sites cannot use the valid certificate of another web site because they have no means to encrypt the transmission in a way that it can be decrypted with the valid certificate. Since only a trusted CA can embed a URL in the certificate, this ensures that checking the apparent URL with the URL specified in the certificate is an acceptable way of identifying the site.
- TLS also supports a more secure bilateral connection mode whereby both ends of the connection can be assured that they are communicating with whom they believe they are connected.
- This is known as mutual (assured) authentication. Mutual authentication requires the TLS clientside to also maintain a certificate. TLS involves three

basic phases:

1. Peer negotiation for algorithm support
2. Key exchange and authentication
3. Symmetric cipher encryption and message authentication

- During the first phase, the client and server negotiate cipher suites, which determine which ciphers are used; makes a decision on the key exchange and authentication algorithms to be used; and determines the message authentication codes.
- The key exchange and authentication algorithms are typically public key algorithms. The message authentication codes are made up from cryptographic hash functions. Once these decisions are made, data transfer may begin.

UNIT 4

PART A

1. What are the six layers of cloud services and their providers?

1. Cloud application (SaaS)
2. Cloud software environment (PaaS)
3. Cloud software infrastructure
4. Collocation cloud services (LaaS)
5. Network cloud services (NaaS)
6. Hardware/Virtualization cloud services (HaaS)

2. Differentiate cloud services in the 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 storedata	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

3. List out the three cases of static cloud resource provisioning policies.

1. Overprovisioning with the peak load causes heavy resource waste.
2. Underprovisioning (along the capacity line) of resources results in losses by both user and provider in that paid demand by the users (the shaded area above the capacity) is not served and wasted resources still exist for those demanded areas below the provisioned capacity.
3. The constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste.

4. Discuss about demand-driven method.

1. This method adds or removes computing instances based on the current utilization level of the allocated resources.
2. 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.

5. Define event-driven method.

1. This scheme adds or removes machine instances based on a specific time event.
2. 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.

6. What are the features of the popularity-driven method?

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

7. What Is IGG?

- The Inter-Grid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources.
- Peering arrangements established between gateways enable the allocation of resources from multiple grids to establish the execution environment.

8. Discuss the features of Google File System.

- Very large sustainable reading and writing bandwidth
- Mostly continuous accessing instead of random accessing.
- The programming interface is similar to that of the POSIX file system accessing interface.

9. Define HDFS and its features.

- Hadoop comes with a distributed filesystem called HDFS, which stands for Hadoop Distributed Filesystem. HDFS is a filesystem designed for storing very large files with streaming data access patterns, running on clusters of commodity hardware.
- The open source clone of GFS.

- Written in Java.
- The programming interfaces are similar to POSIX but not identical.

10. List out the features of CEx.

- 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.

11. What is XaaS?

- XaaS is **Anything-as-a-Service (XaaS),** which is also a subset of cloud computing. XaaS broadly encompasses a process of activating reusable software components over the network.
- The most common and successful example is Software-as-a-Service.

12. Discuss security issues with a cloud-computing vendor.

1. **Privileged user access** —Inquire about who has specialized access to data, and about the hiring and management of such administrators.
2. **Regulatory compliance** —Make sure that the vendor is willing to undergo external audits and/or security certifications.
3. **Data location** —Does the provider allow for any control over the location of data?
4. **Data segregation** —Make sure that encryption is available at all stages, and that these encryption schemes were designed and tested by experienced professionals.
5. **Recovery** —Find out what will happen to data in the case of a disaster. Do

they offer complete restoration? If so, how long would that take?

6. **Investigative support** —Does the vendor have the ability to investigate any inappropriate or illegal activity?
7. **Long-term viability** —What will happen to data if the company goes out of business? How will data be returned, and in what format?

13. What is SAML?

- SAML is an XML-based standard for communicating authentication, authorization, and attribute information among online partners.
- It allows businesses to securely send assertions between partner organizations regarding the identity and entitlements of a principal.

14. How OAuth is used for secure authorization?

- OAuth is an open protocol, initiated by Blaine Cook and Chris Messina, to allow secure API authorization in a simple, standardized method for various types of web applications.
- OAuth is a method for publishing and interacting with protected data.
- For developers, OAuth provides users access to their data while protecting account credentials. OAuth allows users to grant access to their information, which is shared by the service provider and consumers without sharing all of their identity.

15. What are the features of OpenID?

- OpenID is an open, decentralized standard for user authentication and access control that allows users to log onto many services using the same digital identity. It is a single-sign-on (SSO) method of access control.
- As such, it replaces the common log-in process (i.e., a log-in name and a password) by allowing users to log in once and gain access to resources across

participating systems.

16. List out three basic phases of TLS.

1. Peer negotiation for algorithm support
2. Key exchange and authentication
3. Symmetric cipher encryption and message authentication

17. What are the necessary security features of 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
- Auditing and copyright compliance through proactive enforcement
- Shifting of control of data operations from the client environment to cloud providers .
- Protection of sensitive and regulated information in a shared environment.

18. Name a few cloud component that needs security?

- 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.

19. Differentiate Active and Passive attacks

- Passive attacks steal sensitive data or passwords.
- Active attacks manipulate kernel data structures which will cause major damage to cloud servers.

20. What is cloud security?

Cloud security is attributed to user confidentiality, data integrity, access control, firewalls, IDSes, defense capability against viruses or worm attacks, reputation systems,copyright protection, data lock-in, APIs, data-center security policies, trust negotiation,and security auditing services.

21. What are the risks of storing data in the Cloud?

- Reliability
- Security
- User error
- Access problem

22. What are the Security Issues in the Cloud

In theory, minimizing any of the issues would help:

a. Loss of Control
b. Lack of trust
c. Multi-tenancy

23. What are Physical and Cyber Security Protection at Cloud/Data Centers ?

- Secure data centers and computer buildings
- Use redundant utilities at multiple sites
- Trust delegation and negotiation

- Worm containment and DDoS defense
- Reputation system for data centers
- Fine-grained file access control
- Copyright protection and piracy prevention
- Privacy protection

24. What are the security Challenges in VMs?

Buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms. In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits, the man-in-the-middle attack for VM migrations.

PART B

1. Explain the stack of six layers of cloud services and their providers.
2. Discuss Resource Provisioning Methods with examples.
3. Explain Cloud Security Challenges in detail.
4. Explain SaaS security and its features.
5. What is security governance? Discuss virtual machine security in detail.
6. Explain IAM and its security standards.