Study Material

(Course Name: Cloud Computing and Course Code: BCA50115)

Module I: Introduction to Cloud Computing

Table of Contents

Introduction to Cloud Computing

1.1 Basic Concepts of Computer Networks

Computer network refers to the connection of multiple computers with independent functions and their peripheral devices in different geographical locations through communication lines. Under the management and coordination of network operating systems, network management software, and network communication protocols, a computer system realizes resource sharing and information transmission through a connecting medium. The connection medium can be cable, twisted pair, optical fiber, microwave, carrier wave, and communication satellite.

Computer network has the function of sharing hardware, software, and data resources and can centrally process, manage, and maintain shared data

1.1.1 The Composition of a Computer Network

A typical computer network can be logically divided into two parts: resource subnet and communication subnet. The **resource subnet** is composed of hosts, terminals and peripherals, various software resources and information resources, and is mainly responsible for the entire network's information processing, providing network services and resource sharing functions for network users. The **communication subnet** is the network, which is composed of node switches, communication lines, and other communication equipment. It is mainly responsible for the data communication of the whole network, providing the network users with data transmission, transfer, processing and transformation and other communication processing work.

1.1.2 Classification of Computer Networks

Based on the characteristics of computer networks, there are also many forms of division. For example, it can be classified according to geographic scope, scope of use, and transmission medium, and it can also be classified according to information exchange methods and topological structures.

Classification by geographic region

Classified by geographic area, the network can be divided into three types: Local Area Network (LAN), Metropolitan Area Network (MAN), and Wide Area Network(WAN).

(1) **Local Area Network (LAN):** The local area network is usually installed in a building or campus (park), covering a geographic range of 10 meter to 1000 meters with high transmission rate of upto 10 Gbps. Through the local area network, various computers can share resources, such as sharing printers and databases.

(2) **Metropolitan Area Network (MAN)** The metropolitan area network is confined to a city, covering a geographic range of 10 km to 100 km with moderate transmission rate upto 100 Mbps. The metropolitan area network is an extension of the local area network, used to connect the local area network, covering a wide range in terms of transmission media and wiring structure.

(3) **Wide Area Network (WAN)** The WAN's geographic range is 100 km to 10000 km or even more with a low transmission rate upto 10-20 Mbps. This range can be a region or a country, or even the whole world, so it is called a wide area network.

Classification by scope of use

Classified by the scope of use, the network can be divided into public and private networks.

(1) **Public network** The public network is usually set up, managed, and controlled by the telecommunications department. The transmission and switching devices in the network can be provided (such as leased) for use by any department and unit, such as the Public Switched Telephone Network (PSTN), Digital Data Network (DDN), and Integrated Services Digital Network (ISDN).

(2) **Private network**: A specific unit or department establishes the private network, and other units or departments are not allowed to use it. For example, industries such as finance, petroleum, and railways have their own private networks. When setting up a private network, you can lease the transmission lines of the telecommunications department, or you can lay the lines yourself, but the cost of the latter is very high.

Classified by Transmission Medium

According to the classification of transmission media, the network can be divided into the following two categories:

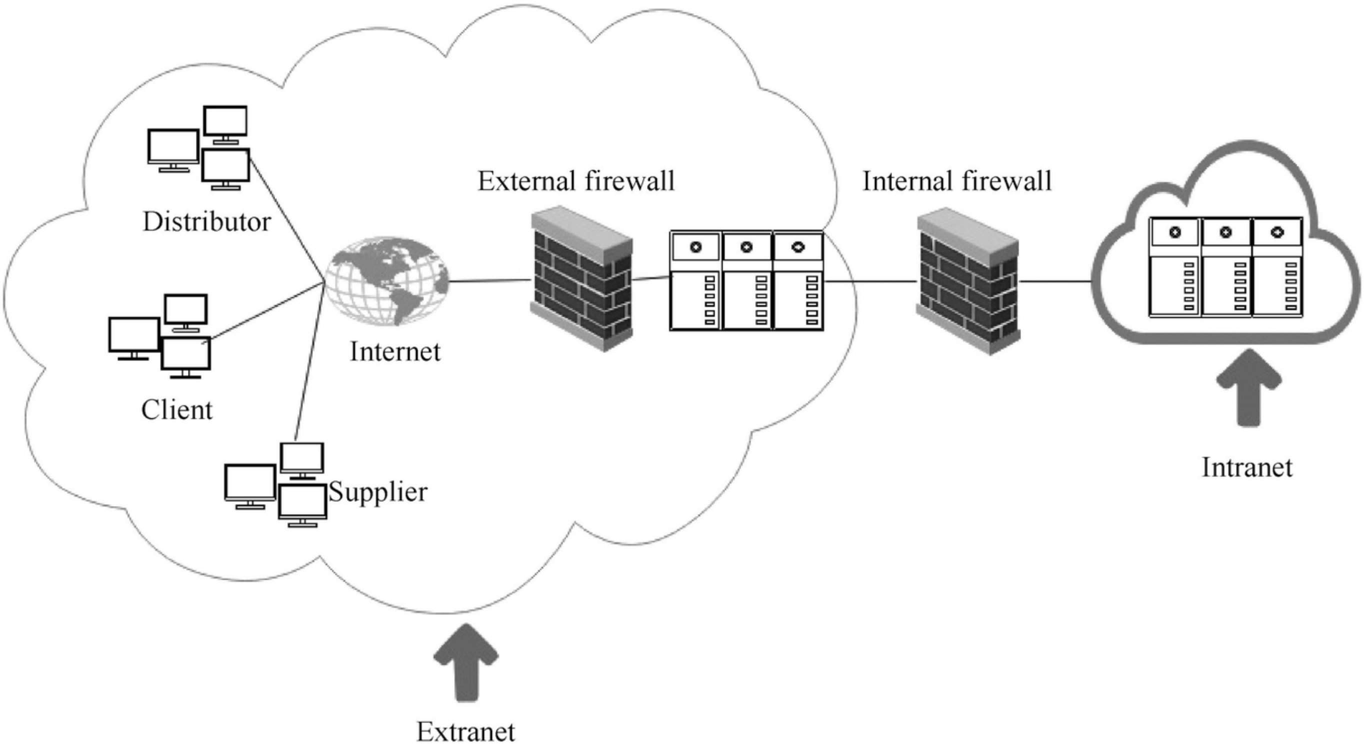
(1) **Wired network**: Wired network refers to a network that uses wired physical media such as coaxial cables, twisted pairs, and optical fibers to transmit data. ‘

(2) **Wireless network:** Wireless network refers to a network that uses wireless forms such as radio waves, satellites, microwaves, and lasers to transmit data.

Classified by Application Type

Classified by application type, the network can be divided into Intranet, Extranet, and Internet.

(1) **Intranet**: Intranet refers to the intranet of an enterprise, which is composed of internal computers and equipment, network environment, software platform, etc., for security reasons, usually only allows access to the internal data center or internal shared resources of the enterprise and does not allow access to extranets and the Internet. It is generally isolated from the extranet by a firewall. The intranet uses the same technology as the Internet, uses TCP/IP as the communication protocol
(2) **Extranet** : Extranet is a cooperative network that uses information technology to interconnect the networks of companies with their customers or assist companies to accomplish their common goals. Users can access it through different technologies, such as using IP tunnels, VPNs, or dedicated dial-up networks.
(3) **Internet**: The internet is a global network of interconnected devices that communicate using protocols like TCP/IP. TCP (Transmission Control Protocol) ensures reliable delivery of data packets, while IP (Internet Protocol) handles addressing and routing. Other important protocols include HTTP (Hypertext Transfer Protocol) for web browsing, SMTP (Simple Mail Transfer Protocol) for email, and FTP (File Transfer Protocol) for file sharing. These protocols enable various services and applications to function seamlessly on the internet.



1.1.3 Topology Structure of Computer Network

Topological structure refers to the geometric arrangement between the network's communication lines and each site (computer or network communication equipment, hereinafter referred to as nodes). Computer network topology refers to the physical or logical arrangement of devices and connections in a computer network. Here are some common network topologies:

Bus Topology: In a bus topology, all devices are connected to a single communication line called a bus. Data is transmitted along the bus, and each device receives the data, but only the intended recipient processes it.

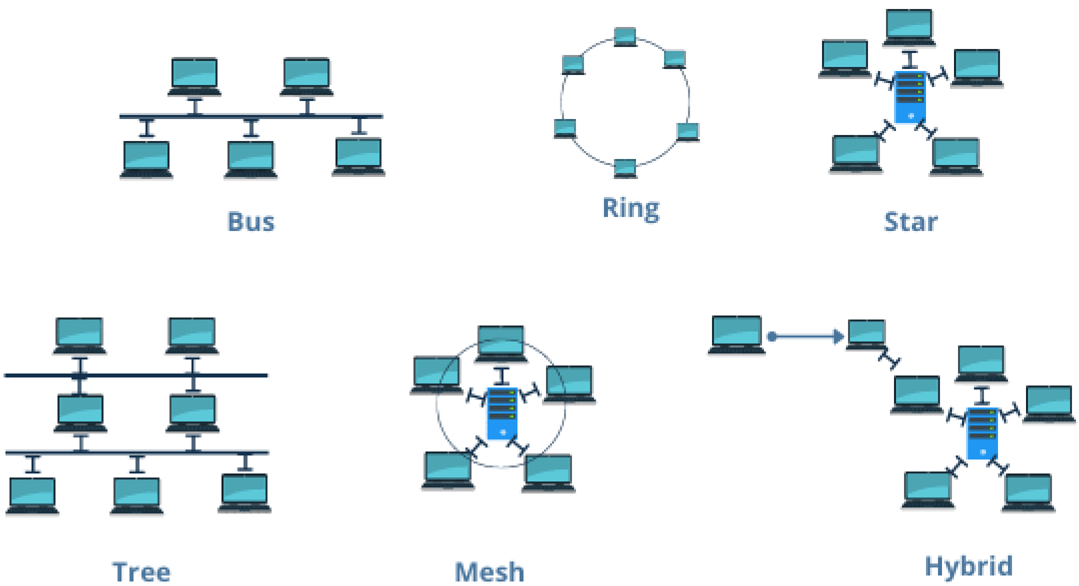
Star Topology: A star topology features a central hub or switch that acts as a central connection point for all devices in the network. Each device is connected directly to the hub, and communication occurs through the hub.

Ring Topology: In a ring topology, devices are connected in a circular manner, where each device is connected to two neighboring devices forming a ring. Data travels in a unidirectional manner around the ring, passing through each device.

Mesh Topology: A mesh topology provides a direct point-to-point connection between all devices in the network. Each device is connected to every other device, allowing for multiple paths for data transmission, enhancing reliability and redundancy.

Tree Topology: A tree topology resembles a hierarchical structure with a root node connected to multiple intermediate nodes, which are further connected to individual end devices.

Hybrid Topology: Hybrid topologies combine two or more different network topologies to meet specific requirements. For example, a network might use a combination of star and bus topologies, where each star network is connected to a central bus network.



1.1.4 Network Layering and Encapsulation

Network layering refers to the division of network functionality into separate layers, each responsible for specific tasks in the communication process. Layering provides a modular and hierarchical structure, allowing for easier implementation, maintenance, and interoperability of network protocols and technologies. Two commonly used models for network layering are the OSI (Open Systems Interconnection) model and the TCP/IP (Transmission Control Protocol/Internet Protocol) model.

OSI Model Layers:

Physical Layer: The physical layer deals with the physical transmission of raw data bits over the network medium, such as copper wires, optical fibers, or wireless signals.

Data Link Layer: The data link layer provides error-free and reliable data transfer between directly connected devices. It handles framing, error detection, and correction, and controls access to the physical medium.

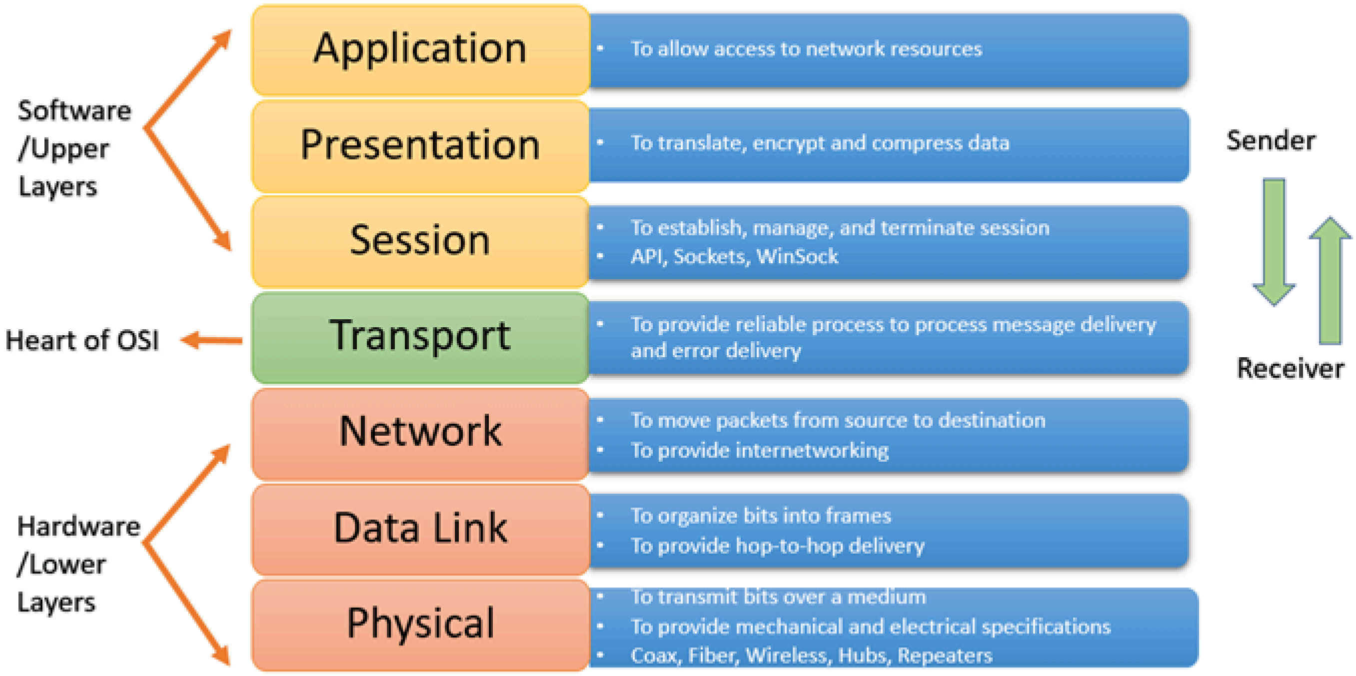
Network Layer: The network layer manages logical addressing and routing of data packets between different networks. It determines the best path for packet delivery and provides services like IP addressing and routing protocols.

Transport Layer: The transport layer ensures reliable and orderly data delivery between end systems. It establishes connections, breaks data into segments, performs error recovery, and manages flow control. Protocols like TCP and UDP operate at this layer.

Session Layer: The session layer establishes, manages, and terminates communication sessions between applications on different devices. It provides mechanisms for synchronization, checkpointing, and recovery in case of interruptions.

Presentation Layer: The presentation layer handles data formatting, encryption, compression, and conversion between different data formats, ensuring that data from different systems can be understood by the receiving device.

Application Layer: The application layer provides network services directly to the end-user applications. It includes protocols for services like file transfer (FTP), email (SMTP), web browsing (HTTP), and remote login (SSH).



1.1.5 Network Interconnection Equipment

Different computer networks can be connected by internet-connected devices to form a larger network and realize data communication and resource sharing between networks. There are many kinds of Internet-connected devices that work at different levels of network protocols. Common network-connected devices include repeaters, hubs, bridges, switches, and routers.

Repeaters: They are used to extend the length of the network. They were created to regenerate and amplify weak signals, thus extending the length of the network. The basic function of a repeater is to restructure and amplify the data signal up to its original level. Repeaters operate at the physical layer of the OSI model

Hub: A Hub is basically a multi-port repeater, it acts as a multiplexor and connects multiple cables coming from different connections. Hubs cannot filter data, so packets of data are sent to all connected devices;

Bridge: A bridge operates on the data link layer. It is a repeater with additional filtering functionality based on reading the source and destination MAC addresses. It is also used to interconnect two LANs that operate under the same protocol.

Switch: A switch is a multiport bridge; it is a data link layer device. The switch is very efficient; it performs error checking before forwarding packets

Router: Routers link two or more different networks; these can consist of various types of LAN network segments. A router receives packets and selects the optimal route to forward the packet across the network. The routers work on the network level of the OSI model.

Gateways: A gateway is a network device or software that acts as an entry point or interface between two different networks or network segments. It provides a bridge or connection between networks with different protocols, architectures, or communication methods

1.1.6 Network Virtualization

Network virtualization is one of the basic core technologies supporting cloud computing and plays a vital role in the cloud computing system.

Cloud computing is a new computing model based on the principle of distributed computing. Although cloud computing has more flexible service capabilities for multiple users than previous grid computing and service computing, distributed computing is still the foundation of cloud computing. Therefore, computer networks play an important role in all aspects of cloud computing: cloud service providers need to coordinate the management and scheduling of resources through the network and integrate different types of resources for users to access in the form of services.

Network virtualization simulates multiple logical networks on a physical network. Network virtualization can help protect IT environments from threats from external networks such as the Internet, while enabling users to access applications and data quickly and securely.

Traditional network virtualization content generally refers to VPN, VLAN, and virtual network devices.

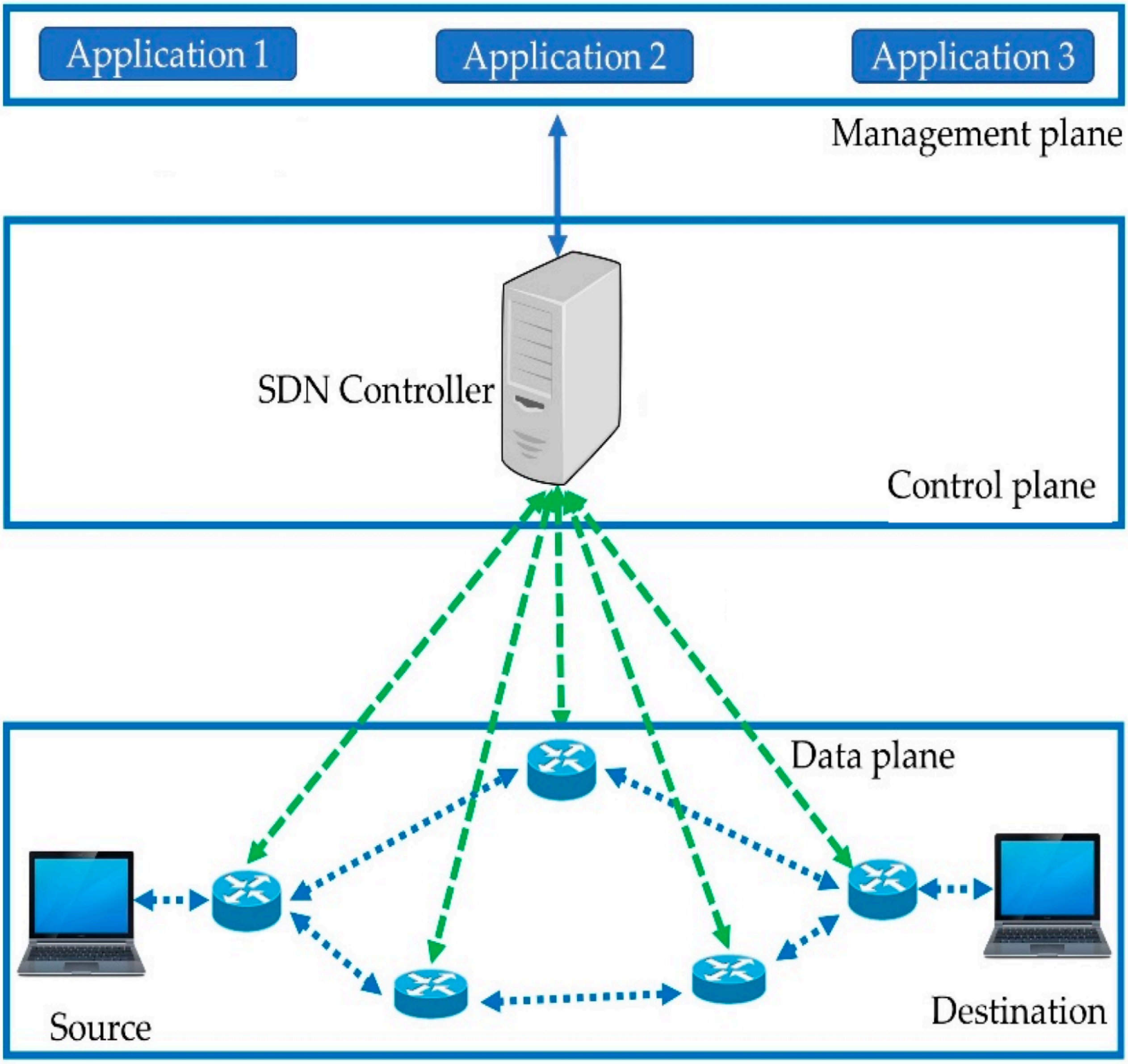
VPN : A VPN is a temporary, secure connection over a public network, usually the Internet, a secure, stable tunnel through a chaotic public network. The tunnel allows data to be encrypted to use the Internet safely.

VLAN: VLAN is a virtual LAN that network administrators can logically divide into different broadcast areas according to the actual application needs of different users in the same physical LAN

1.1.7 Software-Defined Network

Software Defined Network is a networking architecture which enables the control and management of network using software application, by separating control plane from the data plane.

Control plane is the routing process which consists of one or more controllers which are considered brain of the SDN network and Data plane is the forwarding process of the network packets.



1.2 Concepts of Distributed System

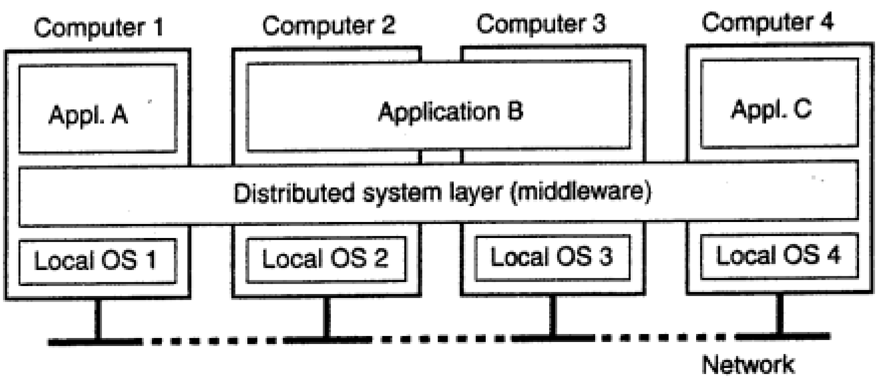
1.2.1 Introduction

Distributed systems are a type of computer system composed of multiple interconnected nodes that work together to achieve a common goal. Each node in a distributed system has its own memory and processing capabilities, and they communicate and coordinate with each other to perform tasks.

Distributed System can be defined as **"A distributed system is a collection of independent computers that appears to its users as a single coherent system"**

In principle, distributed systems should also be relatively easy to expand or scale. This characteristic is a direct consequence of having independent computers, but at the same time, hiding how these computers actually take part in the system as a whole. A distributed system will normally be continuously available, although perhaps some parts may be temporarily out of order. Users and applications should not notice that parts are being replaced or fixed, or that new parts are added to serve more users or applications.

In order to support heterogeneous computers and networks while offering a single-system view, distributed systems are often organized by means of a layer of software-that is, logically placed between a higher-level layer consisting of users and applications, and a layer underneath consisting of operating systems and basic communication facilities



1.2.2 Goals of Distributed System

A distributed system should make resources easily accessible; it should reasonably hide the fact that resources are distributed across a network; it should be open; and it should be scalable

Making Resources Accessible: The main goal of a distributed system is to make it easy for the users (and applications) to access remote resources, and to share them in a controlled and efficient way. Resources can be just about anything, but typical examples include things like printers, computers, storage facilities, data, files, Web pages, and networks, to name just a few. There are many reasons for wanting to share resources.

One obvious reason is that of economics. For example, it is cheaper to let a printer be shared by several users in a small office than having to buy and maintain a separate printer for each user.

Distribution Transparency: An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers. A distributed system that is able to present itself to users and applications as if it were only a single computer system is said to be transparent.

Openness: Another important goal of distributed systems is openness. An open distributed system is a system that offers services according to standard rules that describe the syntax and semantics of those services. For example, in computer networks, standard rules govern the format, contents, and meaning of messages sent and received. Such rules are formalized in protocols

Scalability: Scalability is the ability of a distributed system to handle an increasing workload by adding more resources, accommodating more users, or processing larger amounts of data.

Distributed systems should provide high performance and responsiveness, even as the workload grows. This ensures that the system can handle increased demand without degradation in performance or user experience. Scalable distributed systems often incorporate fault-tolerant mechanisms to handle failures gracefully. By distributing resources and workload, the system can continue operating even if individual components or nodes fail, ensuring high availability and reliability. Elasticity is an important aspect of scalability in cloud computing. It refers to the ability to automatically and dynamically scale resources up or down based on workload demand. Elasticity allows for cost-effective resource allocation, as resources can be provisioned or released in real-time as needed.

1.2.3 TYPES OF DISTRIBUTED SYSTEMS

1. **Distributed Computing Systems:** Distributed computing systems focus on the efficient execution of computational tasks across multiple nodes or machines. These systems leverage the power of distributed processing to solve complex problems, handle large datasets, and achieve high-performance computing. Examples include cluster computing systems, grid computing systems, and distributed processing frameworks like Apache Spark and Hadoop.

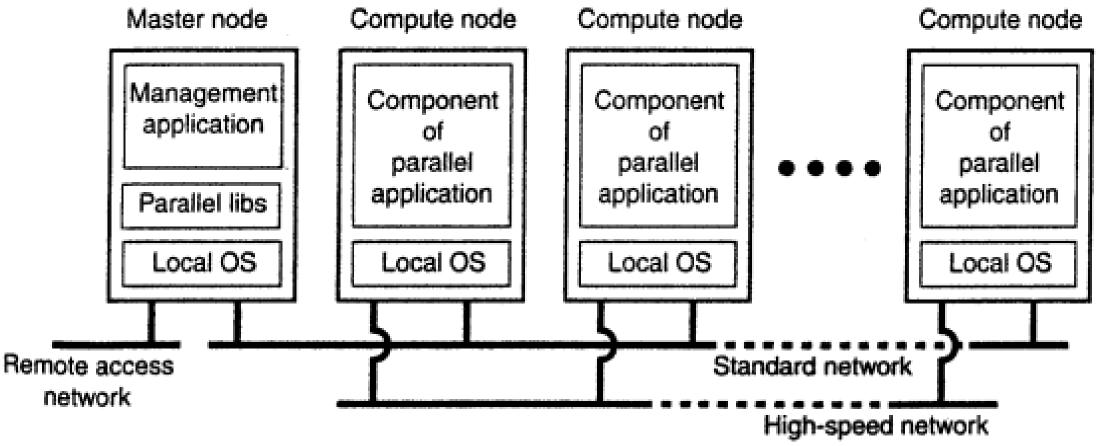


Fig. Distributed Computing (Cluster)

a) Cluster Computing: Cluster computing involves a group of interconnected computers working together as a single system. Each node in the cluster performs parallel computing tasks, sharing resources and coordinating their activities. Cluster computing is used for high-performance computing, scientific simulations, and data-intensive applications.

b) Grid Computing: Grid computing connects geographically dispersed resources, such as computers, storage, and networks, to provide distributed computing power. Grid computing allows resource sharing and collaboration across organizations or institutions for complex scientific or computational tasks.

c) Cloud Computing: Cloud computing is a model that provides on-demand access to computing resources over a network. It encompasses various types of distributed systems, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud computing offers scalability, flexibility, and cost efficiency.

2. **Distributed Information Systems:** Distributed information systems deal with the storage, retrieval, and management of data across multiple nodes. These systems aim to provide access to distributed data repositories, ensuring data consistency, availability, and reliability. Examples of distributed information systems include distributed databases and distributed file systems

    a) Distributed File Systems: Distributed file systems provide access to files stored across multiple nodes. They distribute data across multiple servers, allowing users to access and manipulate files transparently, regardless of their physical location. Examples include the Google File System (GFS) and Hadoop Distributed File System (HDFS).

    b) Distributed Databases: Distributed databases store data across multiple nodes, providing scalability, fault tolerance, and high availability. They distribute data based on different partitioning schemes and replication strategies. Examples include Cassandra, MongoDB, and Amazon DynamoDB.

3. **Distributed Pervasive Systems:** Distributed pervasive systems focus on integrating and coordinating distributed computing devices in everyday environments. These systems aim to create seamless, context-aware, and interconnected environments by incorporating various devices, sensors, and actuators. Examples include Internet of Things (IoT) networks, smart homes, intelligent transportation systems, and sensor networks.

    Each type of distributed system has its own characteristics and requirements:

    1. Distributed Computing Systems prioritize computation and task distribution across multiple nodes, enabling parallel processing and scalability.
    2. Distributed Information Systems focus on data management, ensuring efficient storage, retrieval, and consistency of distributed data.
    3. Distributed Pervasive Systems emphasize the integration and coordination of distributed devices to create intelligent and interconnected environments.

1.2.4 Advantages and Challenges of distributed systems :

Advantages

Scalability: Distributed systems can scale horizontally by adding more nodes, allowing for increased performance and resource availability.

Reliability and Fault Tolerance: Distributed systems can tolerate failures in individual nodes or components, ensuring high availability and fault tolerance.

Resource Sharing: Distributed systems enable efficient resource utilization by allowing multiple users or applications to share and access resources.

Flexibility: Distributed systems provide flexibility in terms of location and access, as users can access resources from anywhere in the network.

Challenges

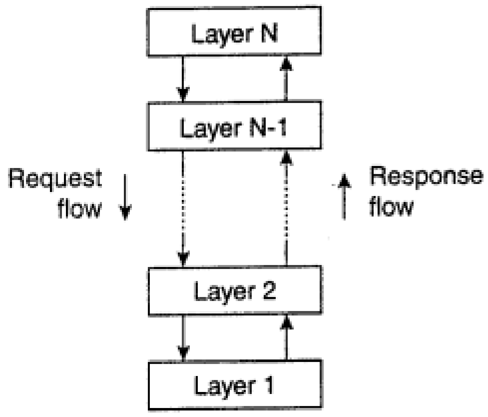
Complexity: Building and managing distributed systems can be complex due to the need for effective communication, coordination, and synchronization among nodes.

Consistency and Data Integrity: Ensuring data consistency and integrity across distributed nodes can be challenging, especially in the presence of concurrent updates.

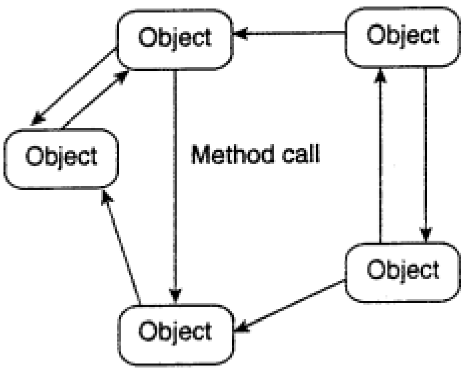
Network Communication: Distributed systems rely heavily on network communication, which can introduce latency, bandwidth limitations, and potential network failures.

1.2.5 Architectural Styles which are important for distributed system

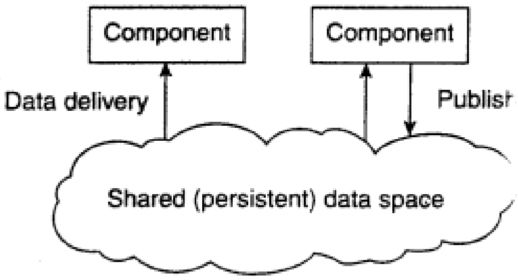
Layered Architectures: Layered architectures divide the system into multiple layers, where each layer has a specific set of responsibilities. Each layer provides services to the layer above it and utilizes services from the layer below it. This modular design allows for separation of concerns and promotes flexibility, scalability, and maintainability. Examples of layered architectures include the OSI model and the TCP/IP protocol stack.



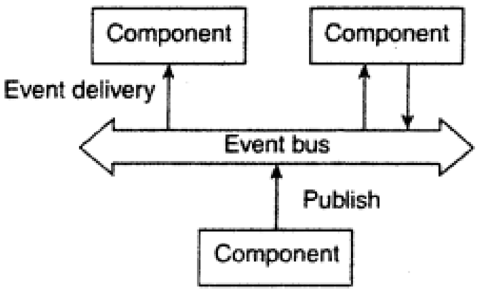
Object-based Architectures: Object-based architectures focus on the organization of the system around objects. Objects encapsulate both data and behavior, and they communicate with each other through method invocations or message passing. This architecture promotes modularity, reusability, and extensibility. Examples of object-based architectures include distributed object systems like CORBA (Common Object Request Broker Architecture) and Java RMI (Remote Method Invocation).



Data-centered Architectures: Data-centered architectures place data at the center of the system's design. Data is stored in a distributed manner, and components interact with the data to perform operations. This architecture ensures data consistency, availability, and scalability. Examples of data-centered architectures include distributed databases, content delivery networks (CDNs), and data grids.



Event-based Architectures: Event-based architectures focus on the flow of events and the reactions triggered by those events. Components in an event-based system communicate by producing and consuming events. When an event occurs, it is published to the system, and interested components (subscribers) can react to that event. This architecture is particularly suitable for real-time systems, event-driven programming, and event processing applications. Examples of event-based architectures include publish-subscribe systems, message queues, and event-driven programming frameworks like Apache Kafka and Node.js.



1.2.6 VIRTUALIZATION

Threads and processes can be seen as a way to do more things at the same time. In effect, they allow us build (pieces of) programs that appear to be executed simultaneously. On a single-processor computer, this simultaneous execution is, of course, an illusion. As there is only a single CPU, only an instruction from a single thread or process will be executed at a time. By rapidly switching between threads and processes, the illusion of parallelism is created.

This separation between having a single CPU and being able to pretend there are more can be extended to other resources as well, leading to what is known as resource virtualization. This virtualization has been applied for many decades, but has received renewed interest as (distributed) computer systems have become more commonplace and complex, leading to the situation that application software is mostly always outliving its underlying systems software and hardware.

Roles of virtualization in distributed systems:

Resource Abstraction: Virtualization abstracts physical resources such as compute, storage, and networking, and presents them as virtual resources. This allows distributed systems to allocate and manage resources based on logical needs rather than physical constraints. Virtualization enables the creation of virtual machines (VMs) or containers that encapsulate applications and their dependencies, providing flexibility and portability.

Resource Isolation: Virtualization provides isolation between different components or tenants in a distributed system. Each virtualized instance, whether it's a VM or container, operates independently, with its own dedicated resources and runtime environment. This isolation ensures that failures or security breaches in one component do not impact others, enhancing reliability and security.

Elasticity and Scalability: Virtualization enables dynamic scaling and elasticity in distributed systems. By abstracting physical resources, virtualization allows for the easy provisioning and deprovisioning of virtual instances based on workload demands. This elasticity allows distributed systems to efficiently scale resources up or down, adapting to changing demands and optimizing resource utilization.

Fault Tolerance and High Availability: Virtualization plays a role in achieving fault tolerance and high availability in distributed systems. By decoupling applications and services from the underlying hardware, virtualization allows for the migration of virtual instances to different physical hosts in the event of failures. Live migration techniques enable seamless movement of VMs or containers, ensuring continuity of service and minimizing downtime.

Resource Consolidation: Virtualization enables efficient resource utilization by consolidating multiple virtual instances onto a single physical host. Through techniques such as hypervisor-based virtualization or containerization, multiple applications or services can run concurrently on shared resources, effectively utilizing the available compute, storage, and networking capacities.

Sandboxing and Testing: Virtualization provides a sandboxed environment for testing and development purposes. Developers can create isolated virtual instances to test new software, configurations, or updates without affecting the production environment. This allows for experimentation, debugging, and validation of distributed system components before deployment.

Migration and Mobility: Virtualization enables the mobility of virtual instances across different physical hosts or cloud environments. This flexibility allows for workload migration, load balancing, disaster recovery, and data center maintenance without interrupting services. Virtual instances can be moved between data centers or cloud providers, providing agility and avoiding vendor lock-in.

Virtualization technology, such as hypervisors and containerization platforms, plays a pivotal role in the design, management, and operation of distributed systems. It provides the foundation for efficient resource allocation, scalability, fault tolerance, and flexibility, making distributed systems more agile, reliable, and cost-effective.

1.2.7 Communication Mechanism in Distributed Computing

RPC (Remote Procedure Call) and RMI (Remote Method Invocation) are communication mechanisms used in distributed computing to enable remote interactions between different components or processes. Here's an explanation of RPC and RMI:

Remote Procedure Call (RPC):

RPC is a communication protocol that allows a program or process on one computer to invoke a procedure or method on a remote computer. It enables distributed systems to communicate and interact by abstracting the network communication details. The basic idea behind RPC is that a client program makes a procedure call as if it were a local call, and the RPC mechanism takes care of marshaling the parameters, transmitting the request over the network, invoking the remote procedure, and returning the results to the client.

In RPC, the client and server use stubs (client-side) and skeletons (server-side) to hide the complexities of network communication. The client-side stub acts as a proxy for the remote procedure, and the server-side skeleton receives the RPC request and invokes the corresponding procedure. The RPC framework takes care of marshaling/unmarshaling data, handling error conditions, and managing network connections.

RPC allows distributed systems to achieve a high level of abstraction, making it easier to develop client-server applications and enabling seamless communication between different nodes in a network. Examples of RPC frameworks include gRPC, XML-RPC, and JSON-RPC.

Remote Method Invocation (RMI):

RMI is a Java-specific technology that enables the invocation of methods on remote objects. It provides a mechanism for Java objects to interact and communicate across different Java Virtual Machines (JVMs) in a distributed system. RMI allows Java programs to perform method invocations on remote objects as if they were local method calls, abstracting the complexities of network communication.

In RMI, the client and server interact through Java interfaces and objects. The client-side stub and the server-side skeleton handle the communication details, including parameter marshaling/unmarshaling and network transmission. RMI provides transparent object serialization and deserialization, allowing Java objects to be passed between the client and server.

RMI includes features such as object activation, garbage collection, and remote exception handling. It simplifies the development of distributed Java applications by providing a seamless and familiar programming model.

Both RPC and RMI are essential in distributed computing as they enable remote communication and method invocation, allowing components in a distributed system to interact and exchange data. They abstract the complexities of network communication and provide a higher level of abstraction, making it easier to develop distributed applications.

1.3 Concepts of Cloud Computing and its Necessity

1.3.1 Introduction:

Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards.

Cloud computing represents a real paradigm shift in the way in which systems are deployed. The massive scale of cloud computing systems was enabled by the popularization of the Internet and the growth of some large service companies. Cloud computing makes the long-held dream of utility computing possible with a pay-as-you-go, infinitely scalable, universally available system. With cloud computing, you can start very small and become big very fast.

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

Abstraction: Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous.

Virtualization: Cloud computing virtualizes systems by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

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

Cloud Computing Examples

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

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

3. Amazon Web Services: One of the most successful cloud-based businesses is Amazon Web Services, which is an Infrastructure as a Service offering that lets you rent virtual computers on Amazon's own infrastructure.

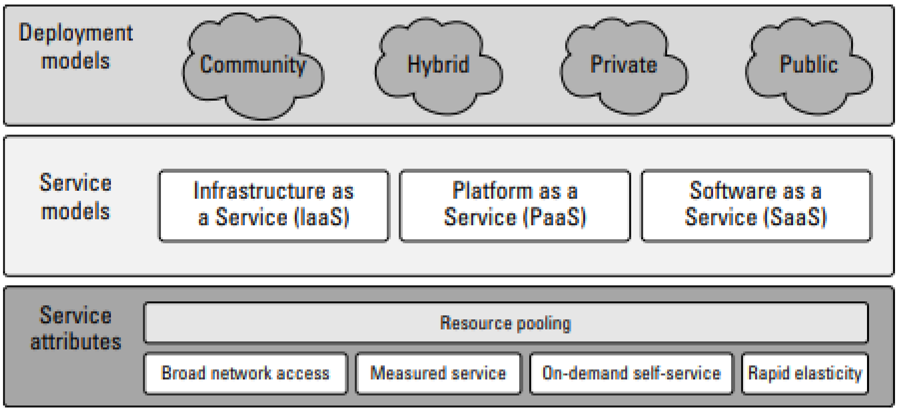
1.3.2 Cloud Types:

Deployment Models: This refers to the location and management of the clouds infrastructure

Service Models: This consists of the particular types of services that you can access on the cloud computing platform.

1.3.3 NIST Working Definition of Cloud Computing

National Institute of Standards and Technology (NIST) has defined cloud computing as "Cloud computing 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. This cloud model is composed of five essential characteristics, three service models, and four deployment models"



1.3.4 Essential Characteristics:

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

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, tablets, laptops, and workstations).

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. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

1.3.5 Deployment models

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

1. **Public cloud:** The public cloud infrastructure is available for public use alternatively for a large industry group and is owned by an organization selling cloud services.

2. **Private cloud:** The private cloud infrastructure is operated for the exclusive use of an organization. The cloud may be managed by that organization or a third party. Private clouds may be either on- or off-premises.

3. **Hybrid cloud:** A hybrid cloud combines multiple clouds (private, community of public) where those clouds retain their unique identities, but are bound together as a unit. A hybrid cloud may offer standardized or proprietary access to data and applications, as well as application portability.

4. **Community cloud:** A community cloud is one where the cloud has been organized to serve a common function or purpose. It may be for one organization or for several organizations, but they share common concerns such as their mission, policies, security, regulatory compliance needs, and so on. A community cloud may be managed by the constituent organization(s) or by a third party.

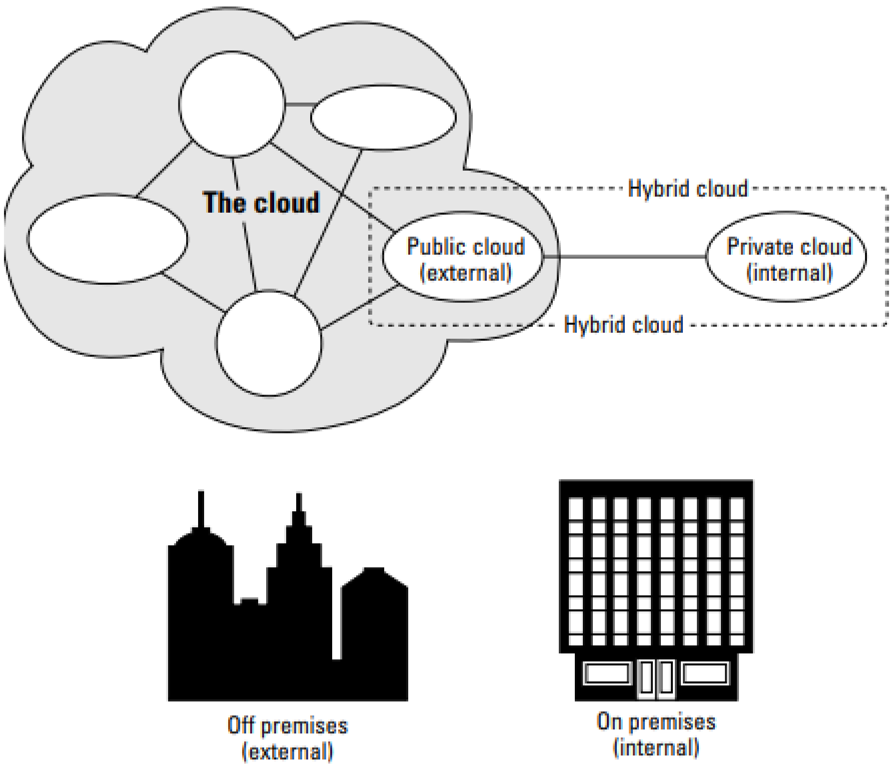


Fig. Deployment locations for different cloud types

1.3.6 Service Models

Three service types have been universally accepted:

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

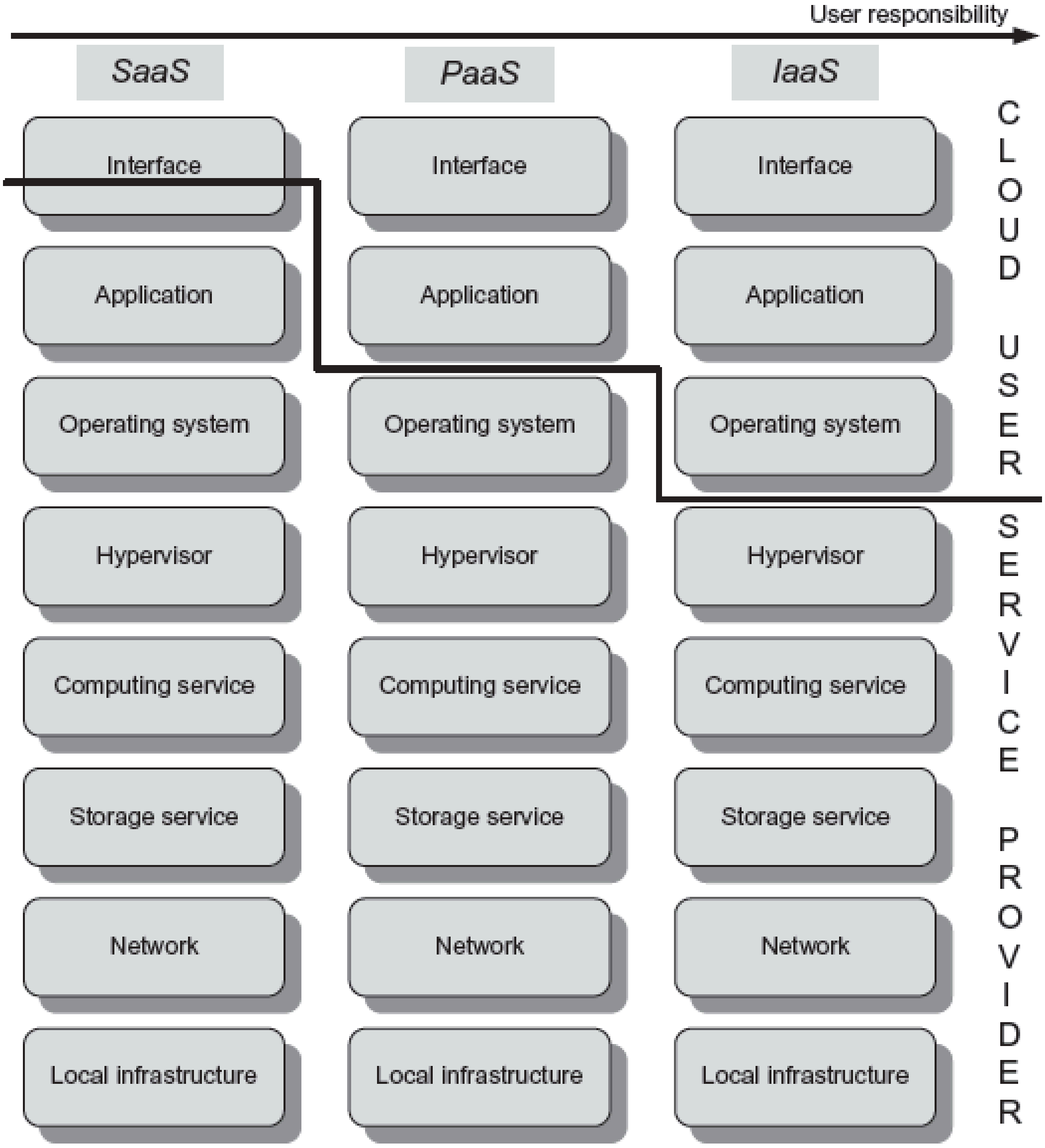
2. **Platform as a Service:** PaaS provides virtual machines, operating systems, applications, services, development frameworks, transactions, and control structures. The client can deploy its applications on the cloud infrastructure or use applications that were programmed using languages and tools that are supported by the PaaS service provider. The service provider manages the cloud infrastructure, the operating systems, and the enabling software. The client is responsible for installing and managing the application that it is deploying.

3. **Software as a Service:** SaaS is a complete operating environment with applications, management, and the user interface. In the SaaS model, the application is provided to the client through a thin client interface (a browser, usually), and the customer's responsibility begins and ends with entering and managing its data and user interaction. Everything from the application down to the infrastructure is the vendor's responsibility

The three different service models taken together have come to be known as the **SPI model of cloud computing**. Many other service models have been mentioned: StaaS, Storage as a Service; IdaaS, Identity as a Service; CmaaS, Compliance as a Service; and so forth

1.3.7 Limits of responsibility between the cloud user and the cloud service provider

The limits of responsibility between the cloud user and the cloud service provider are different for the three service-delivery models. In the case of SaaS the user is partially responsible for the interface; the user responsibility increases in the case of PaaS and includes the interface and the application. In the case of IaaS the user is responsible for all the events occurring in the virtual machine running the application.



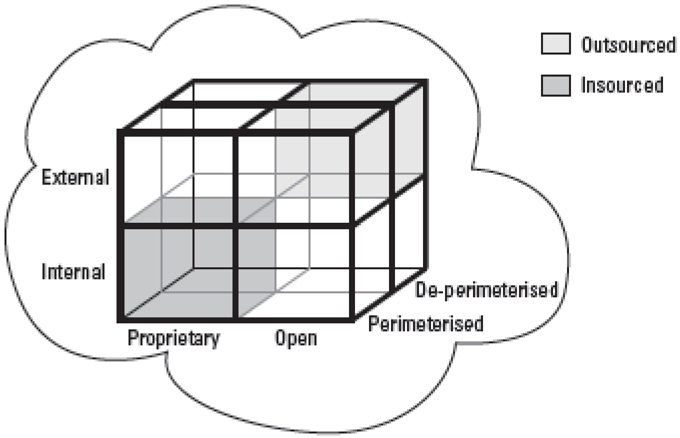
1.3.8 Cloud Cube Model

The Open Group maintains an association called the Jericho whose main focus is how to protect cloud networks. The group has an interesting model that attempts to categorize a cloud network based on four dimensional factors

The four dimensions of the Cloud Cube Model are listed here:

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

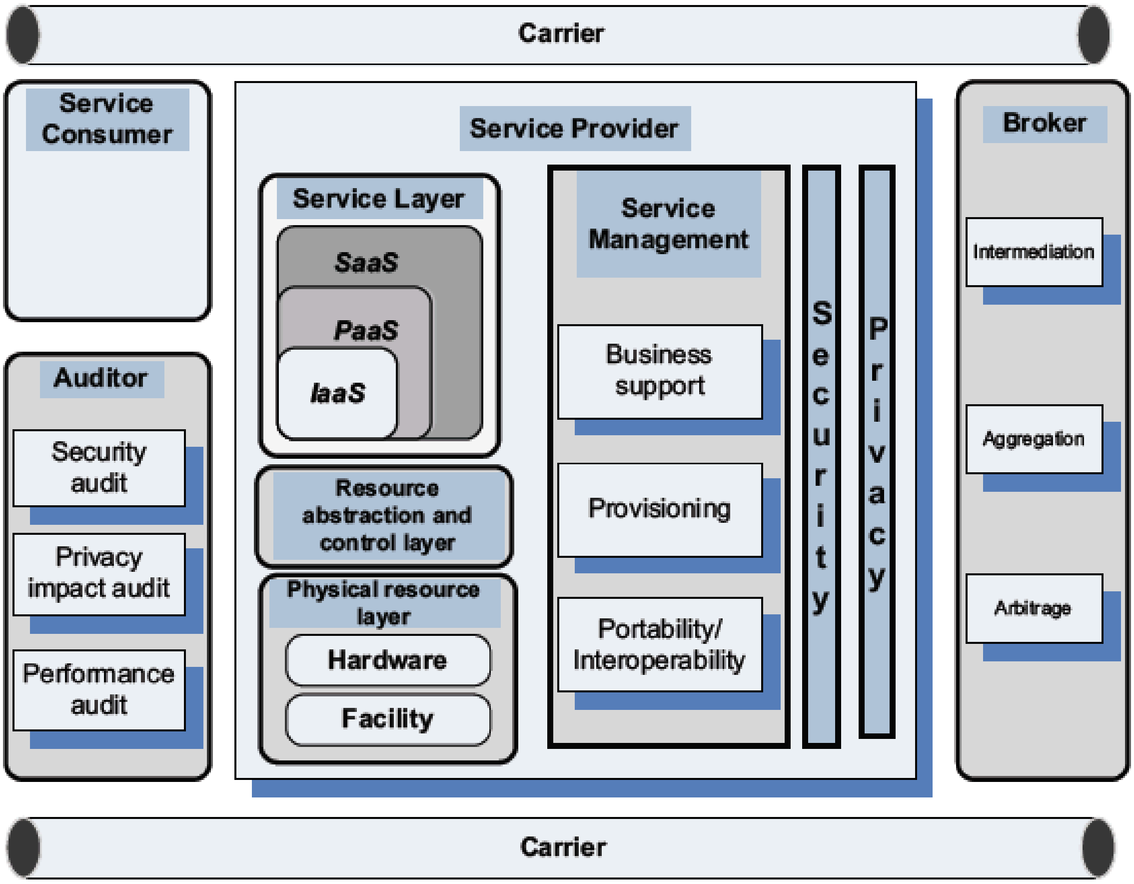
4**. Sourcing:** Insourced or Outsourced means whether the service is provided by the customer or the service provider.



1.3.9 NIST reference model

According to the NIST reference model the entities involved in cloud computing are :

a) **Service Consumer**: the entity that maintains a business relationship with and uses service from service providers;
b) **Service Provider**: the entity responsible for making a service available to service consumers;
c) **Carrier:** the intermediary that provides connectivity and transport of cloud services between providers and consumers;
d) **Broker**: an entity that manages the use, performance, and delivery of cloud services and negotiates relationships between providers and consumers; and
e) **Auditor:** a party that can conduct independent assessment of cloud services, information system operations, performance, and security of the cloud implementation.



1.3.10 Cloud Service Providers in use and their Significance.

Cloud computing has revolutionized the way businesses and individuals access and utilize computing resources. Cloud service providers play a critical role in delivering various cloud-based services, offering a range of benefits such as scalability, cost-efficiency, and accessibility. Below are some examples of cloud service providers categorized by their service offerings and their significance in the cloud computing landscape.

Infrastructure as a Service (IaaS) Providers:

1. Amazon Elastic Compute Cloud (EC2): Amazon EC2 is one of the leading IaaS providers, offering virtual machines on-demand to users. It provides scalable compute capacity in the form of instances, where users can deploy their applications and manage the underlying infrastructure. EC2 has been instrumental in popularizing the pay-as-you-go model for computing resources.

2. Eucalyptus: Eucalyptus is an open-source IaaS platform that allows users to build private or hybrid clouds compatible with Amazon Web Services (AWS) APIs. It enables organizations to create their own on-premises cloud environments while maintaining compatibility with public cloud services.

3. GoGrid: GoGrid is a cloud infrastructure provider that offers customizable cloud servers, storage solutions, and networking resources. It caters to businesses of all sizes and provides a user-friendly interface for managing cloud deployments.

4. FlexiScale: FlexiScale is a UK-based IaaS provider known for its elastic cloud servers and scalable storage solutions. It allows users to easily scale resources up or down as needed, making it suitable for dynamic workloads.

5. Linode: Linode specializes in providing high-performance virtual private servers (VPS) with a focus on simplicity and ease of use. It has gained popularity among developers and small businesses for its straightforward pricing and reliable services.

6. RackSpace Cloud: RackSpace is a well-known cloud provider offering both IaaS and managed hosting services. Its cloud platform provides a range of services, including compute instances, storage, and networking capabilities.

7. Terremark: Acquired by Verizon and now known as Verizon Enterprise Solutions, Terremark provides a suite of cloud services, including IaaS, for enterprises and government organizations. It focuses on security, compliance, and reliability.

Platform as a Service (PaaS) Providers:

1. Force.com: Force.com is a PaaS platform provided by Salesforce, offering a development environment for building custom business applications and integrating them with Salesforce's CRM capabilities.

2. GoGrid CloudCenter: GoGrid's CloudCenter is a PaaS offering that simplifies application deployment and management on the cloud. It provides a range of tools and services for developers to build, deploy, and monitor applications.

3. Google App Engine: Google App Engine is a PaaS platform that allows developers to build and deploy web applications on Google's infrastructure. It automatically scales applications based on demand, making it suitable for web developers.

4. Windows Azure Platform: Now known as Microsoft Azure, it is a comprehensive cloud platform offering a wide range of services, including PaaS capabilities. Developers can build, deploy, and manage applications using various programming languages and tools.

Software as a Service (SaaS) Providers:

1. Google Apps: Google Apps offers a suite of cloud-based productivity and collaboration tools, including Gmail, Google Docs, Google Drive, and more. It allows users to access and use these applications via a web browser.

2. Oracle On Demand: Oracle On Demand provides a range of enterprise software applications hosted in the cloud, including customer relationship management (CRM), enterprise resource planning (ERP), and human capital management (HCM) solutions.

3. Salesforce.com: Salesforce.com is a leading SaaS provider known for its cloud-based CRM solutions. It helps businesses manage customer data, sales, marketing, and customer support processes.

4. SQL Azure: SQL Azure is Microsoft's cloud-based database service, part of the Azure platform. It enables developers to create, manage, and scale relational databases in the cloud.

The significance of these cloud service providers lies in their ability to offer flexible, scalable, and cost-effective solutions to businesses and individuals. By leveraging cloud services from these providers, organizations can focus on their core competencies, reduce IT infrastructure costs, and improve agility in responding to changing market demands. Additionally, cloud services facilitate collaboration, enable access from anywhere, and ensure high availability and reliability of applications and data. As cloud computing continues to evolve, these providers will continue to play a crucial role in driving innovation and technological advancements across various industries.

Questions

MCQ

1. Which of the following is NOT an essential characteristic of cloud computing according to the NIST definition?
    a) On-demand self-service
    b) Broad network access
    c) Resource pooling
    d) Client-based management

2. Which cloud service model provides virtual machines, virtual storage, and other hardware assets as resources that clients can provision?
    a) Infrastructure as a Service (IaaS)
    b) Platform as a Service (PaaS)
    c) Software as a Service (SaaS)
    d) Cloud Cube as a Service

3. In the Cloud Cube Model, what does "Perimeterized (Per) / De-perimeterized (D-p)" refer to?
    a) Ownership of the cloud infrastructure
    b) The physical location of data
    c) The security boundary of the cloud
    d) Sourcing of the cloud service

4. According to the NIST reference model, an entity that manages the use, performance, and delivery of cloud services and negotiates relationships between providers and consumers is known as?
    a) Broker
    b) Service Provider
    c) Carrier
    d) Auditor
5. Which cloud service provider offers a suite of cloud services, including Infrastructure as a Service (IaaS), for enterprises and government organizations, with a focus on security and compliance?

a) Amazon Elastic Compute Cloud
b) Google App Engine
c) Terremark
d) Salesforce.com

6. Which service model in cloud computing offers a complete operating environment with applications, management, and user interface?
   a) Infrastructure as a Service (IaaS)
   b) Platform as a Service (PaaS)
   c) Software as a Service (SaaS)
   d) Service Provider as a Service (SPaaS)

7. The NIST reference model includes which entity responsible for conducting an independent assessment of cloud services, information system operations, performance, and security?
   a) Service Consumer
   b) Broker
   c) Carrier
   d) Auditor

8. What is the primary purpose of the Cloud Cube Model?
   a) To categorize cloud networks based on their deployment models
   b) To define the NIST reference model for cloud computing
   c) To identify the boundaries of cloud services between providers and consumers
   d) To categorize cloud networks based on their physical location, ownership, security boundary, and sourcing

9. What is the purpose of a repeater in a computer network?
   a) Filter data packets based on MAC addresses
   b) Provide error-free and reliable data transfer
   c) Extend the length of the network by regenerating and amplifying weak signals
   d) Establish a secure connection over a public network

10. Which network topology provides a direct point-to-point connection between all devices?
    a) Star Topology
    b) Bus Topology
    c) Ring Topology
    d) Mesh Topology
11. What is the role of Network Virtualization in cloud computing?
    a) It simulates multiple logical networks on a physical network
    b) It establishes, manages, and terminates communication sessions between applications
    c) It defines and controls the network through software programming
    d) It ensures reliable and orderly data delivery between end systems
12. Which layer of the OSI model handles data formatting?
    a) Application Layer
    b) Transport Layer

c) Data Link Layer
d) Presentation Layer

13. What is the primary function of a router in a computer network?
    a) Regenerate and amplify weak signals
    b) Provide error-free and reliable data transfer
    c) Connect multiple cables coming from different connections
    d) Link two or more different networks and select the optimal route for data transmission

14. What concept is used for dividing IP address spaces into smaller, manageable subnets?
    a) CIDR
    b) Virtual LAN (VLAN)
    c) Quality of Service (QoS)
    d) Border Gateway Protocol (BGP)

15. What does SDN stand for in the context of computer networks?
    a) Secure Data Networking
    b) Simplified Data Networking
    c) Software-Defined Network
    d) Static Data Networking

Short Questions (3 marks each)

1. Define cloud computing and explain its significance in the IT industry.
2. List and briefly explain three essential characteristics of cloud computing according to the NIST definition.
3. What are the four deployment models of cloud computing? Provide a brief description of each.
4. Differentiate between Infrastructure as a Service (IaaS) and Software as a Service (SaaS) cloud service models.
5. What are the three service models of cloud computing? Provide an example of a service offered under each model.
6. What is the Cloud Cube Model, and how does it categorize cloud networks based on different dimensions?
7. Write about software defined network.
8. Write about Network Interconnection Equipment
9. Write about various network topologies.
10. What are the goals of distributed system

Long Questions (5 marks each)

1. Discuss the OSI model and its seven layers, highlighting the functions of each layer in the communication process. Compare the OSI model with the TCP/IP model.
2. Explain the concepts of network virtualization with suitable examples.
3. Explain various types of distributed system
4. Explain architectural styles which are important in distributed system
5. Describe NIST reference model with diagram

Questions for Advanced Learner

1. Define distributed systems and discuss the key challenges they pose compared to traditional centralized systems. How do distributed systems handle issues like data consistency and fault tolerance?
2. Explain the fundamental characteristics of cloud computing and how they contribute to its necessity in modern business environments. Discuss the advantages and disadvantages of cloud computing over traditional on-premises IT infrastructure.
3. Compare and contrast three major cloud service providers (e.g., Amazon Web Services, Microsoft Azure, Google Cloud Platform) based on their offerings, pricing models, and geographic presence. Discuss their significance in the cloud computing market.
4. Describe the role of cloud service providers in enabling scalability and elasticity for businesses. How do these providers ensure high availability, data security, and compliance with industry standards for their customers?

MCQ Answers

Question SL No.	Answers
1	d) Client-based management
2	a) Infrastructure as a Service (IaaS)
3	c) The security boundary of the cloud
4	b) Broker
5	c) Terremark
6	c) Software as a Service (SaaS)
7	d) Auditor
8	d) To categorize cloud networks based on their physical location, ownership, security boundary, and sourcing
9	c) Extend the length of the network by regenerating and amplifying weak signals
10	d) Mesh Topology
11	a) It simulates multiple logical networks on a physical network
12	d) Presentation Layer
13	d) Link two or more different networks and select the optimal route for data transmission
14	a) CIDR
15	c) Software-Defined Network