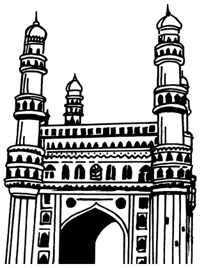
***Rahul's* ✔**

Topper'sVoice

NEW SYLLABUS



B.C.A.

II Year IV Sem

Latest 2023 Edition

DISTRIBUTED AND CLOUD COMPUTING

- ☞ **Study Manual**
- ☞ **Important Questions**
- ☞ **Short Question & Answers**
- ☞ **Choose the Correct Answers**
- ☞ **Fill in the blanks**
- ☞ **Solved Model Papers**

` .169/-

- by -

WELL EXPERIENCED LECTURER

Rahul Publications™

Hyderabad. Cell : 9391018098, 9505799122

All disputes are subjects to Hyderabad Jurisdiction only

B.C.A.

II Year IV Sem

DISTRIBUTED AND CLOUD COMPUTING

Inspite of many efforts taken to present this book without errors, some errors might have crept in. Therefore we do not take any legal responsibility for such errors and omissions. However, if they are brought to our notice, they will be corrected in the next edition.

Price ` 169

Sole Distributors : **Cell : 9391018098, 9505799122**

VASU BOOK CENTRE

Shop No. 2, Beside Gokul Chat, Koti, Hyderabad.

Maternity Hospital Opp. Lane, Narayan Naik Complex, Koti, Hyderabad.
Near Andhra Bank, Subway, Sultan Bazar, Koti, Hyderabad -195.

CONTENTS

DISTRIBUTED AND CLOUD COMPUTING

STUDY MANUAL

Important Questions V - VIII

Unit - I 1 - 28

Unit - II 29 - 54

Unit - III 55 - 76

Unit - IV 77 - 106

Unit - V 107 - 122

SOLVED MODEL PAPERS

MODEL PAPER - I 123 - 124

MODEL PAPER - II 125 - 126

MODEL PAPER - III 127 - 128

SYLLABUS

UNIT - I

Examples of Distributed Systems–Trends in Distributed Systems – Focus on resource sharing – Challenges. Case study: World Wide Web. System Model – Inter process Communication – the API for internet protocols – External data representation and Multicast communication.

UNIT - II

Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation and Objects: Remote Invocation – Introduction – Request-reply protocols – Remote procedure call – Remote method invocation. Case study: Java RMI – Group communication – Publish-subscribe systems – Message queues – Shared memory approaches – Distributed objects – Case study: Enterprise Java Beans -from objects to components.

UNIT - III

Introduction to Cloud Computing: Cloud Computing in a Nutshell, System Models for Distributed and Cloud Computing, Roots of Cloud Computing, Grid and Cloud, Layers and Types of Clouds, Desired Features of a Cloud, Basic Principles of Cloud Computing, Challenges and Risks, Service Models.

UNIT - IV

Virtual Machines and Virtualization of Clusters and Data Centers: Levels of Virtualization, Virtualization Structures Tools and Mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Virtualization Data-Center Automation. Case studies: Xen Virtual machine monitors- Xen API. VMware - VMware products-Vmware Features.

UNIT - V

Cloud computing architectures over Virtualized Data Centers: Data-Center design and Interconnection networks, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms, GAE, AWS, Azure, Inter-cloud Resource Management.

Contents

Topic **Page No.**

UNIT - I

1.1 Examples of Distributed Systems 1
1.2 Trends in Distributed Systems 8
1.3 Focus on Resource Sharing 9
1.4 Challenges In Distributed Computing 10
1.5 Case Study: World Wide Web 11
1.6 System Model 13
1.7 Inter Process Communication 17
1.8 The API For Internet Protocols 20
1.9 External Data Representation and Multicast Communication 22
➢ **Short Question and Answers 23**
➢ **Choose the Correct Answers 27**
➢ **Fill in the Blanks 28**

UNIT - II

2.1 Network Virtualization 29
2.1.1 Overlay Networks 29
2.2 Case Study: MPI 30
2.3 Remote Method Invocation and Objects 31
2.3.1 Remote Invocation 31
2.3.1.1 Introduction 31
2.3.1.2 Request Reply Protocols 32
2.3.1.3 Remote Procedure Call 34
2.3.1.4 Remote Method Invocation 37

Topic **Page No.**

2.4 Case Stydt: Java Rmi 38

2.4.1 Group Communication 39

2.4.2 Publish Subscribe Systems 44

2.4.3 Message Queues 45

2.4.4 Shared Memory Approaches 46

2.5 Distributed Objects 47

2.6 Case Study: Enterprise Java Beans 49

2.6.1 From Objects To Components 50

➢ **Short Question and Answers 51**

➢ **Choose the Correct Answers 53**

➢ **Fill in the Blanks 54**

UNIT - III

3.1 Introduction to Cloud Computing 55

3.1.1 Cloud Computing in a Nutshell 55

3.2 System Models for Distributed and Cloud Computing 58

3.3 Roots of Cloud Computing 60

3.4 Grid and Cloud Layers 61

3.5 Types of Clouds 61

3.6 Desired Features of A Cloud 66

3.7 Basic Principles of Cloud Computing 67

3.8 Challenges and Risks 68

3.9 Service Models 69

➢ **Short Question and Answers 71**

➢ **Choose the Correct Answers 75**

➢ **Fill in the Blanks 76**

Topic **Page No.**

UNIT - IV

4.1 Virtual Machines and Virtualization of Clusters and Data Centers 77

4.1.1 Levels Of Virtualization 77

4.1.2 Virtualization Structures Tools And Mechanisms 79

4.1.3 Virtualization of CPU 82

4.1.4 Memory and I/O Devices 83

4.1.5 Virtual Clusters and Resource Management 85

4.1.6 Virtualization 90

4.1.7 Data Center Automation 91

4.2 Case Studies: Xen Virtual Machine Monitors 91

4.2.1 XEN API 91

4.3 VMware 93

4.3.1 Vmware products 98

4.3.2 Vmware Features 99

➢ **Short Question and Answers 101**

➢ **Choose the Correct Answers 105**

➢ **Fill in the Blanks 106**

UNIT - V

5.1 Cloud Computing Architectures Over Virtualized Data Centers 107

5.1.1 Data Center Design And Inter-connection Networks 107

5.2 Architectural Design of Compute and Storage Clouds 108

5. 3 Public Cloud Platforms 111

5.4 GAE 113

Topic **Page No.**

5.5 AWS 114

5.6 AZURE 115

5.7 Inter Cloud Resource Management 116

➢ **Short Question and Answers 119**

➢ **Choose the Correct Answers 121**

➢ **Fill in the Blanks 122**

Important Questions

UNIT - I

1. **What is distributed system? Write various examples of distributed systems.**

Ans :

Refer Unit-I, Q.No. 1

2. **Explain about various types of distributed systems.**

Ans :

Refer Unit-I, Q.No. 2

3. **What are the emerging trends in distributed systems? Explain.**

Ans :

Refer Unit-I, Q.No. 3

4. **Explain about Resource Sharing in Distributed System**

Ans :

Refer Unit-I, Q.No. 4

5. **Explain the challenges in distributed computing.**

Ans :

Refer Unit-I, Q.No. 5

6. **Explain about distributed system model.**

Ans :

Refer Unit-I, Q.No. 8

7. **Explain about inter process communication is distributed systems.**

Ans :

Refer Unit-I, Q.No. 9

8. **What is the process of synchronization in IPC? Explain.**

Ans :

Refer Unit-I, Q.No. 13

9. **Explain about the API for the Internet protocols.**

Ans :

Refer Unit-I, Q.No. 14

UNIT - II

1. What is an overlay network? Explain its applications.

Ans :

Refer Unit-II, Q.No. 1

2. Give the introduction about RPC and distributed objects?

Ans :

Refer Unit-II, Q.No. 4

3. Explain different types of RPC.

Ans :

Refer Unit-II, Q.No. 7

4. Explain briefly about RMI state in architecture and working mechanism.

Ans :

Refer Unit-II, Q.No. 9

5. Write a Case Study on JAVA RMI.

Ans :

Refer Unit-II, Q.No. 11

6. Write a note on Group Communication in distributed systems.

Ans :

Refer Unit-II, Q.No. 12

7. Explain the concept of Publish-Subscribe?

Ans :

Refer Unit-II, Q.No. 15

8. Explain about distributed objects.

Ans :

Refer Unit-II, Q.No. 18

9. Write a case study on Java Beans.

Ans :

Refer Unit-II, Q.No. 20

UNIT - III

1. Define Cloud Computing? Explain the benefits of cloud computing.

Ans :

Refer Unit-III, Q.No. 1

2. Write about the challenges of cloud computing.

Ans :

Refer Unit-III, Q.No. 3

3. Explain about system models for distributed and cloud computing.

Ans :

Refer Unit-III, Q.No. 4

4. Explain different types of clouds.

Ans :

Refer Unit-III, Q.No. 6

5. Describe briefly about desired features of a Cloud.

Ans :

Refer Unit-III, Q.No. 7

6. Explain the basic principles of cloud compuitng.

Ans :

Refer Unit-III, Q.No. 8

7. Explain the challenges and risks of cloud computing.

Ans :

Refer Unit-III, Q.No. 9

8. Explain about different cloud service models.

Ans :

Refer Unit-III, Q.No. 10

UNIT - IV

1. Explain about the levels of virtualization.

Ans :

Refer Unit-IV, Q.No. 1

2. Explain about virtualization structures tools and mechanisms.

Ans :

Refer Unit-IV, Q.No. 2

3. Explain about Memory Virtualization.

Ans :

Refer Unit-IV, Q.No. 5

4. Explain about various virtual clusters.

Ans :

Refer Unit-IV, Q.No. 7

5. Explain XEN API with architecture diagram.

Ans :

Refer Unit-IV, Q.No. 11

6. Expalin about evolution of VMware.

Ans :

Refer Unit-IV, Q.No. 12

UNIT - V

1. Explain about various public cloud platforms.

Ans :

Refer Unit-V, Q.No. 3

2. Write about Amazon Web services (AWS).

Ans :

Refer Unit-V, Q.No. 5

3. What is data centre? Explain different types of data centres.

Ans :

Refer Unit-V, Q.No. 1

4. Explain about inter cloud resource management.

Ans :

Refer Unit-V, Q.No. 7

UNIT I

Examples of Distributed Systems - Trends in Distributed Systems - Focus on resource sharing - Challenges. Case study: World Wide Web. System Model - Inter process Communication - the API for internet protocols - External data representation and Multicast communication.

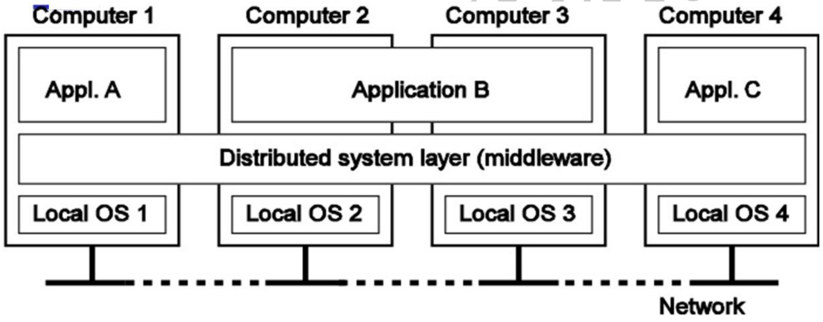
1.1 Examples of Distributed Systems

Q1. What is distributed system? Write various examples of distributed systems.

Ans : **(Imp.)**

Meaning

A distributed system is a network that consists of autonomous computers that are connected using a distribution middleware. They help in sharing different resources and capabilities to provide users with a single and integrated coherent network.



The key features of a distributed system are:

Components in the system are concurrent. A distributed system allows resource sharing, including software by systems connected to the network at the same time.

- The components could be multiple but will generally be autonomous in nature.
- A global clock is not required in a distributed system. The systems can be spread across different geographies.
- Compared to other network models, there is greater fault tolerance in a distributed model.
- Price/performance ratio is much better.

Examples

(i) Web Search

Web search has emerged as a major growth industry in the last decade, with recent figures indicating that the global number of searches has risen to over 10 billion per calendar month. The task of a web search engine is to index the entire contents of the World Wide Web, encompassing a wide range of information styles including web pages, multimedia sources and (scanned) books. This is a very complex task, as current estimates state that the Web consists of over 63 billion pages and one trillion unique web.

(ii) Finance and Commerce

The growth of eCommerce as exemplified by companies such as Amazon and eBay, and underlying payments technologies such as PayPal; the associated emergence of online banking and trading and also complex information dissemination systems for financial markets.

(iii) Information Society

The growth of the World Wide Web as a repository of information and knowledge; the development of web search engines such as Google and Yahoo to search this vast repository; the emergence of digital libraries and the large-scale digitization of legacy information sources such as books (for example, Google Books); the increasing significance of user-generated content through sites such as YouTube, Wikipedia and Flickr; the emergence of social networking through services such as Facebook and MySpace.

(iv) Creative Industries and Entertainment

The emergence of online gaming as a novel and highly interactive form of entertainment; the availability of music and film in the home through networked media centres and more widely in the Internet via downloadable or streaming content; the role of user-generated content (as mentioned above) as a new form of creativity, for example via services such as YouTube; the creation of new forms of art and entertainment enabled by emergent (including networked) technologies.

(v) Healthcare

The growth of health informatics as a discipline with its emphasis on online electronic patient records and related issues of privacy; the increasing role of telemedicine in supporting remote diagnosis or more advanced services such as remote surgery (including collaborative working between healthcare teams); the increasing application of networking and embedded systems technology in assisted living, for example for monitoring the elderly in their own homes.

(vi) Education

The emergence of e-learning through for example web-based tools such as virtual learning environments; associated support for distance learning; support for collaborative or community-based learning.

(vii) Transport and Logistics

The use of location technologies such as GPS in route finding systems and more general traffic management systems; the modern car itself as an example of a complex distributed system (also applies to other forms of transport such as aircraft); the development of web-based map services such as MapQuest, Google Maps and Google Earth.

(viii) Science

The emergence - of the Grid as a fundamental technology for science including the use of complex networks of computers to support the storage, analysis, and processing of (often very large quantities of) scientific data; the associated use of the Grid as an enabling technology for worldwide collaboration between groups of scientists.

(ix) Environmental Management

The use of (networked) sensor technology to both monitor and manage *Environmental management* example to provide early warning of natural disasters such as earthquakes, floods or tsunamis and to co ordinate emergency response; the collation and analysis of global environmental parameters to better understand complex natural phenomena such as climate change addresses. Given that most search engines analyze the entire web content and then carry out sophisticated processing on this enormous database, this task itself represents a major challenge for distributed systems design.

(x) Massively multiplayer online games (MMOGs)

Massively multiplayer online games offer an immersive experience whereby very large numbers of users interact through the Internet with a persistent virtual world. A number of solutions have been proposed for the design of massively multiplayer online games:

(xi) Financial Trading

As a final example, we look at distributed systems support financial trading markets. The financial industry has long been at the cutting edge of distributed systems technology with its need, in particular, for real-time access to a wide range of information sources (for example, current share prices and trends, economic and political developments). The industry employs automated monitoring and trading applications .

Q2. Explain about various types of distributed systems.

Ans : **(Imp.)**

Types of Distributed System

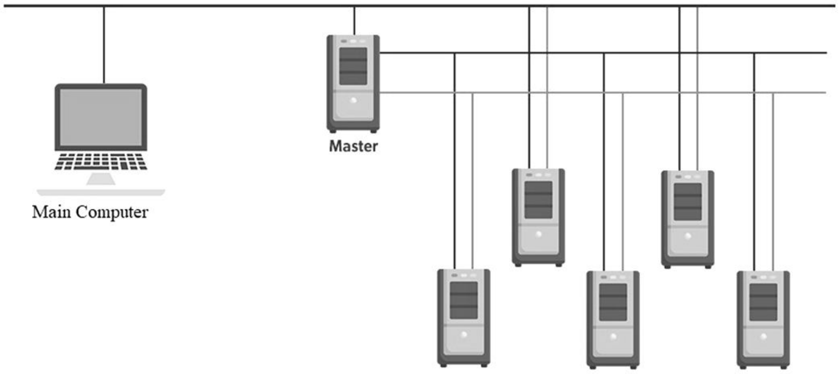
A distributed system is also known as distributed computer science and distributed databases; independent components that interact with other different machines that exchange messages to achieve common goals. As such, the distributed system appears to the end-user like an interface or a computer. Together the system can maximize resources and information while preventing system failure and did not affect service availability.

1. Distributed Computing System

This distributed system is used in performance computation which requires high computing.

(i) Cluster Computing

A collection of connected computers that work together as a unit to perform operations together, functioning in a single system. Clusters are generally connected quickly via local area networks & each node is running the same operating system.



When input comes from a client to the main computer, the master CPU divides the task into simple jobs and sends it to slave note to do it when the jobs are done by the slave nodes, they send it back to the master node, and then it shows the result to the main computer.

Advantages

- High Performance
- Easy to manage
- Scalable
- Expandability
- Availability
- Flexibility
- Cost effectiveness
- Distributed applications

Disadvantages

- High cost
- The problem in finding the fault
- More space is needed

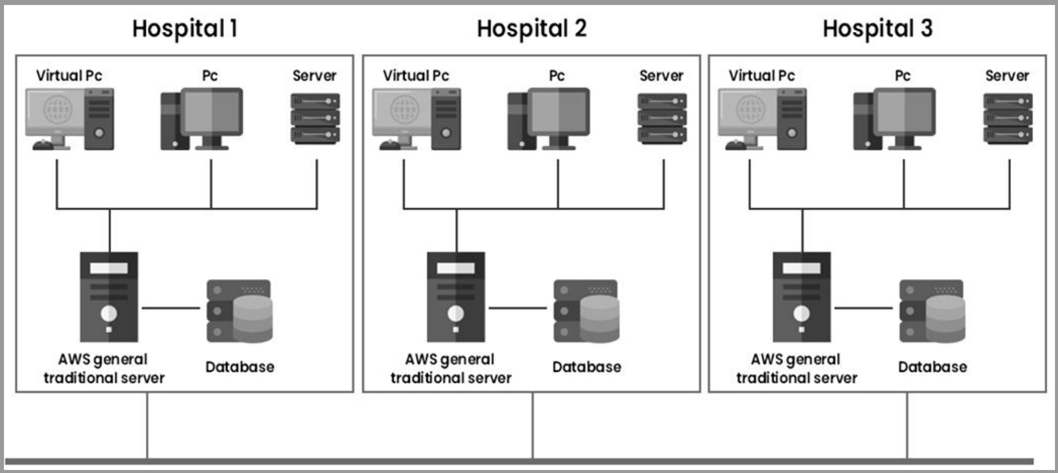
- Increased infrastructure needed
- In distributed systems, it is challenging to provide adequate security because both the nodes and the connections must be protected.
- If every node in a distributed system tries to

Applications

- In many web applications functionalities such as Security, Search Engines, Database servers, web servers, proxy, and email.
- It is flexible to allocate work as small data tasks for processing.
- Assist and help to solve complex computational problems
- Cluster computing can be used in weather modeling
- Earthquake, Nuclear, Simulation, and tornado forecast

(ii) Grid Computing

In grid computing, the subgroup consists of distributed systems, which are often set up as a network of computer systems, each system can belong to a different administrative domain and can differ greatly in terms of hardware, software, and implementation network technology.



The different department has a different computer with different OS to make the control node present which helps different computer with different OS to communicate with each other and transfer messages to work.

Advantages

- Can solve bigger and more complex problems in a shorter time frame. Easier collaboration with other organizations and better use of existing equipment
- Existing hardware is used to the fullest.
- Collaboration with organizations made easier

Disadvantages

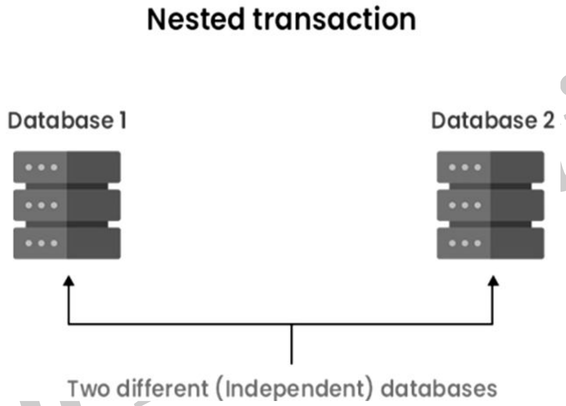
- Grid software and standards continue to evolve
- Getting started learning curve
- Non-interactive job submission
- You may need a fast connection between computer resources.
- Licensing on many servers can be prohibitive for some applications.

Applications of Grid Computing

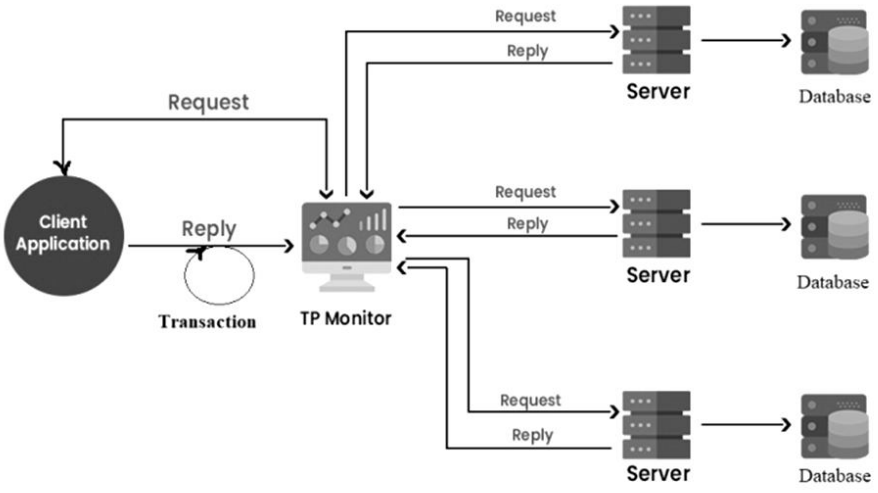
- Organizations that develop grid standards and practices for the guild line.
- Works as a middleware solution for connecting different businesses.
- It is a solution-based solution that can meet computing, data, and network needs.

2. Distributed Information System:

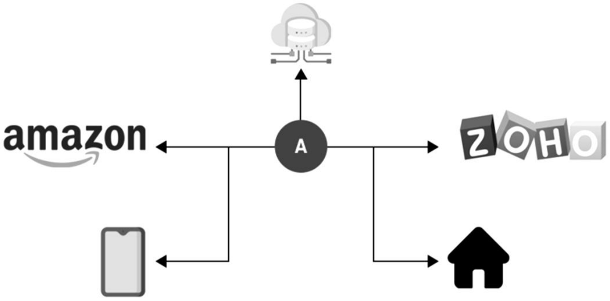
- **Distributed transaction processing:** It works across different servers using multiple communication models. The four characteristics that transactions have:
- **Atomic:** the transaction taking place must be indivisible for the others
- **Consistent:** The transaction should be consistent after the transaction has been done
- **Isolated:** A transaction must not interfere with another transaction
- **Durable:** Once an engaged transaction, the changes are permanent. Transactions are often constructed as several sub-transactions, jointly forming a nested transaction.



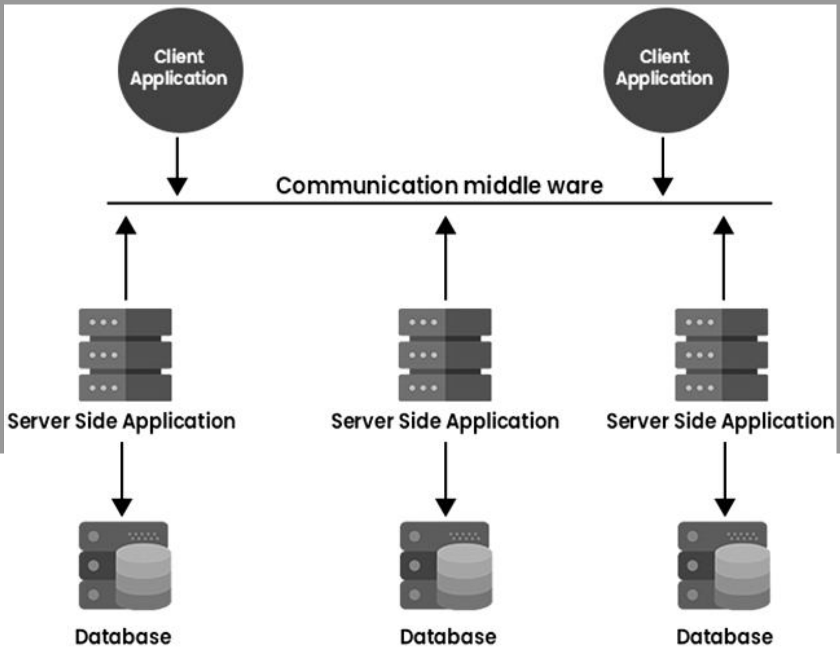
Each database can perform its own individual query containing data retrieval from two different databases to give one single result. In the company's middleware systems, the component that manages distributed (or nested) transactions has formed the application integration core at the server or database. This was referred to as the Transaction Processing Monitor(TP Monitor). Its main task was to allow an application to access multiple servers/databases by providing a transactional programming model. Many requests are sent to the database to get the result, to ensure each request gets successfully executed and deliver result to each request, this work is handled by the TP Monitor.



- **Enterprise application integration:** Enterprise Application Integration (EAI) is the process of bringing different businesses together. The databases and workflows associated with business applications ensure that the business uses information consistently and that changes in data done by one business application are reflected correctly in another's. Many organizations collect different data from different plate forms in the internal systems and then they use those data are used in the Trading system /physical medium.



- **RPC:** Remote Procedure Calls (RPC), a software element that sends a request to every other software element with the aid of using creating a nearby method name and retrieving the data Which is now known as remote method invocation (RMI). An app can have a different database for managing different data and then they can communicate with each other on different platforms. Suppose, if you login into your android device and watch you're video on YouTube then you go to your laptop and open YouTube you can see the same video is in your watch list. RPC and RMI have the disadvantage that the sender and receiver must be running at the time of communication.



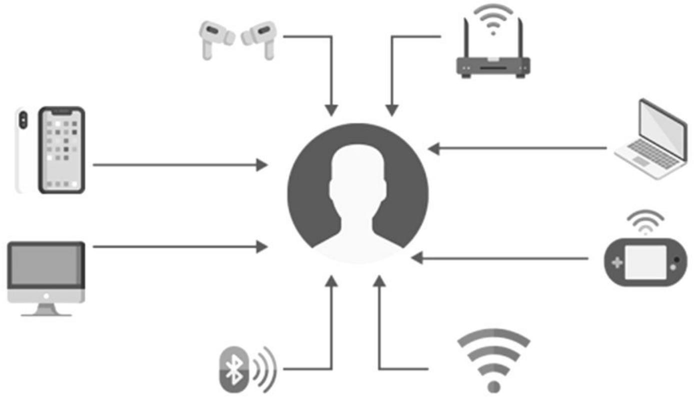
Purposes

- Targets the application rules and implements them in the EAI system so that even if one of the lines of business applications is replaced by the application of another vendor.
- An EAI system can use a group of applications as a front end, provide only one, consistent access interface to those applications, and protect users from learning how to use different software packages.

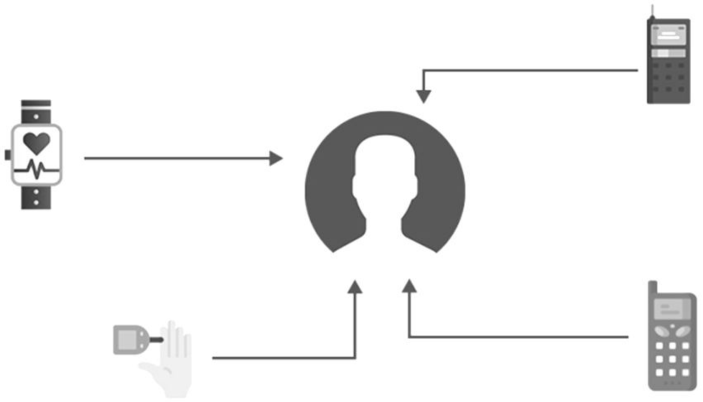
3. Distributed Pervasive System

Pervasive Computing is also abbreviated as ubiquitous (Changed and removed) computing and it is the new step towards integrating everyday objects with microprocessors so that this information can communicate. a computer system available anywhere in the company or as a generally available consumer system that looks like that same everywhere with the same functionality but that operates from computing power, storage, and locations across the globe.

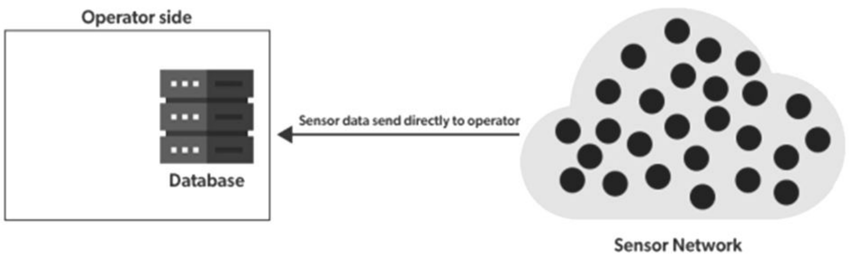
- **Home system:** Nowadays many devices used in the home are digital so we can control them from anywhere and effectively.



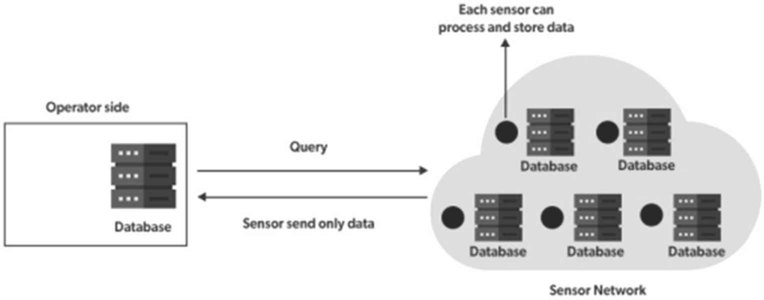
- **Electronic health system:** Nowadays smart medical wearable devices are also present through which we can monitor our health regularly.



- **Sensor network (IoT devices):** Internet devices only send data to the client to act according to the data send to the device.



- Before sensory devices only send and send data to the client but now, they can store and process the data to manage it efficiently.



1.2 Trends in Distributed Systems

Q3. What are the emerging trends in distributed systems? Explain.

Ans : **(Imp.)**

Distributed systems are undergoing a period of significant change and this can be traced back to a number of influential trends:

- The emergence of pervasive networking technology;
- The emergence of ubiquitous computing coupled with the desire to support user mobility in distributed systems;
- The increasing demand for multimedia services;
- The view of distributed systems as a utility.

(i) Internet

The modern Internet is a vast interconnected collection of computer networks of many different types, with the range of types increasing all the time and now including, for example, a wide range of wireless communication technologies such as WiFi, WiMAX, Bluetooth and third-generation mobile phone networks. The net result is that networking has become a pervasive resource and devices can be connected (if desired) at any time and in any place.

The Internet is also a very large distributed system. It enables users, wherever they are, to make use of services such as the World Wide Web, email and file transfer.

(ii) Intranet

- A portion of the Internet that is separately administered and has a boundary that can be configured to enforce local security policies
- Composed of several LANs linked by backbone connections
- Be connected to the Internet via a router

Main issues in the design of components for the use in intranet

- File services
- Firewall
- The cost of software installation and support

(iii) Mobile and Ubiquitous Computing

Technological advances in device miniaturization and wireless networking have led increasingly to the integration of small and portable computing devices into distributed systems. These devices include:

- Laptop computers.
- Handheld devices, including mobile phones, smart phones, GPS-enabled devices, pagers, personal digital assistants (PDAs), video cameras and digital cameras.
- Wearable devices, such as smart watches with functionality similar to a PDA.
- Devices embedded in appliances such as washing machines, hi-fi systems, cars and refrigerators.

The portability of many of these devices, together with their ability to connect conveniently to networks in different places, makes mobile computing possible. Mobile computing is the performance of computing tasks while the user is on the move, or visiting places other than their usual environment. In mobile computing, users who are away from their 'home' intranet They can continue to access the Internet; they can continue to access resources in their home intranet; and there is increasing provision for users to utilize resources such as printers or even sales points that are conveniently nearby as they move around. The latter is also known as location-aware or context-aware computing. Mobility introduces a number of challenges for distributed systems, including the need to deal with variable connectivity and indeed disconnection, and the need to maintain operation in the face of device mobility.

(iv) Portable and handheld devices in a distributed system

Ubiquitous computing is the harnessing of many small, cheap computational devices that are present in users' physical environments, including the home, office and even natural settings. The term 'ubiquitous' is intended to suggest that small computing devices will eventually become so pervasive in everyday objects that they are scarcely noticed. That is, their computational behaviour will be transparently and intimately tied up with their physical function.

The presence of computers everywhere only becomes useful when they can communicate with one another. For example, it may be convenient for users to control their washing machine or their entertainment system from their phone or a 'universal remote control' device in the home. Equally, the washing machine could notify the user via a smart badge or phone when the washing is done.

Ubiquitous and mobile computing overlap, since the mobile user can in principle benefit from computers that are everywhere. But they are distinct, in general. Ubiquitous computing could benefit users while they remain in a single environment such as the home or a hospital. Similarly, mobile computing has advantages even if it involves only conventional, discrete computers and devices such as laptops and printers.

1.3 Focus on Resource Sharing

Q4. Explain about Resource Sharing in Distributed System

Ans : **(Imp.)**

Resource Sharing is basically how the existing resource in the distributed system can be shared and accessed across different computer system. The resource shared can be software, hardware or any data in the distributed system. The resource of distributed system are made available in following ways:

1. Data Migration

The process in which data is transferred from one location to another location in the system. Data is brought to the location of computation, that needs to be accessed by a distributed system. We can also say that the data is migrated from destination part to source, after requesting source to destination.

2. Computation Migration

The process in which computation is transferred rather than the data across the system. The files cannot be transferred to the computation point due to large file size or big data file. So, the computation data is transferred to the data file point and processes the computation and the result is transferred to the computation point.

Advantages

- Increases computational speed.
- **Load Balancing: -** It helps spread the load across the distributed system in order to optimize resource sharing.

Here are some key considerations for resource sharing in distributed computing:

1. **Resource allocation**

   In distributed computing, resources such as processing power, memory, and storage are allocated across the network to meet the needs of various applications and users. Efficient resource allocation is crucial to ensure that each application or user has access to the resources they need, without overburdening any one computer or server.

2. **Load balancing**

   Resource sharing also involves load balancing, which is the process of distributing workloads across the network to optimize resource utilization. Load balancing helps to prevent overloading of individual computers or servers, which can cause performance issues and slow down the entire system.

3. **Resource discovery**

   In a distributed computing environment, resources may be located on different computers or servers across the network. Resource discovery tools are used to locate and access resources as needed, enabling efficient resource sharing across the network.

4. **Security**

   Resource sharing in distributed computing must be carefully managed to ensure that resources are only accessed by authorized users and applications. Access controls and authentication mechanisms are used to ensure that resources are protected from unauthorized access and misuse.

5. **Monitoring and management**

   Monitoring and management tools are used to track resource usage across the network and identify areas where resource sharing can be improved. These tools can help to optimize resource utilization and prevent performance issues.

Overall, efficient resource sharing is essential for maximizing the performance and scalability of distributed computing systems. By carefully managing resource allocation, load balancing, security, and monitoring, organizations can ensure that their distributed computing systems are optimized for maximum efficiency and productivity.

1.4 Challenges In Distributed Computing

Q5. Explain the challenges in distributed computing.

Ans : **(Imp.)**

Distributed computing, like any complex system, presents several challenges that need to be addressed to ensure its efficient and reliable operation. Here are some of the key challenges in distributed computing:

1. **Network connectivity**

   Distributed computing relies on a network of interconnected computers and servers, which can be vulnerable to connectivity issues such as latency, packet loss, and network congestion. These issues can cause delays, failures, and other performance problems that can affect the reliability of the distributed system.

2. **Data consistency and replication**

   Distributed computing often involves the replication of data across multiple servers, which can present challenges for ensuring data consistency and integrity. Synchronization mechanisms and other techniques are needed to ensure that data updates are propagated correctly across all nodes in the network.

3. **Fault tolerance and resilience**

   Distributed computing systems must be designed to be fault-tolerant and resilient, meaning that they can continue to operate even if one or more nodes in the network fail or become unavailable. This requires the use of redundancy and other techniques to ensure that critical functions and services can continue to operate even in the event of failures.

4. **Security and privacy**

   Distributed computing systems often involve the sharing of sensitive data and resources across the network, making them vulnerable to security threats such as hacking, data breaches, and other cyber attacks. Robust security measures and protocols must be implemented to ensure the confidentiality, integrity, and availability of data and resources.

5. **Resource management and optimization**

Distributed computing systems often involve the sharing and allocation of resources across the network, which can present challenges for efficient resource management and optimization. Techniques such as load balancing, resource allocation, and performance monitoring must be used to ensure that resources are used effectively and efficiently.

Q6. Explain the web challenges in distributed computing.

Ans :

Web Challenges in Distributed System

To get an ideal distributed system various web challenges need to be overcome. The major web challenges in distributed system are:

1. **Scalability**

The feature of distributed systems in which if the load of a system increases the performance of the system will not be degraded.

2. **Heterogeneity**

The ability to communication with different devices. For example, communication between computer to mobile or other peripheral devices.

3. **Security Challenges**

There are three types of Security Challenges, i.e. Privacy, Authentication, Availability.

- **Privacy:** Data Shared should maintain confidentiality.
- **Authentication:** Whoever shares the message should have proper identity for that system. Unauthorised users should not access the system.
- **Availability:** Data and Resources should be available.

4. **Handling of Failure**

There are three types of failures that can occur in a system which need to be fixed.

- **Tolerance:** If there is any error occurring while running the system then the system continues operating without any interruption.
- **Redundancy:** It means that any duplicacy or inconsistency should not be found in our system.
- **Exception Handling:** It is all about handling the error which may be found while running the system.

1.5 Case Study: World Wide Web

Q7. Write the case study on WWW.

Ans :

The World Wide Web is an evolving system for publishing and accessing resources and services across the Internet. Through commonly available web browsers, users retrieve and view documents of many types, listen to audio streams and view video streams, and interact with an unlimited set of services. The Web began life at the European requirement to organize their knowledge. This means that documents contain *links* (or *hyperlinks*) – references to other documents and resources that are also stored in the Web. It is fundamental to the user's experience of the Web that when they encounter a given image or piece of text within a document, this will frequently be accompanied by links to related documents and other resources. The structure of links can be arbitrarily complex and the set of resources that can be added is unlimited – the 'web' of links is indeed world-wide.

Bush [1945] conceived of hypertextual structures over 50 years ago; it was with the development of the Internet that this idea could be manifested on a world-wide scale.

The Web is an *open* system: it can be extended and implemented in new ways without disturbing its existing functionality.

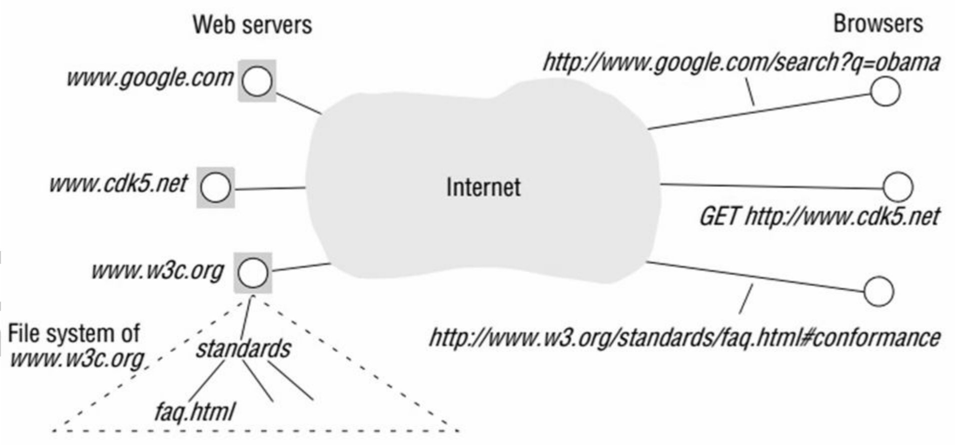
First, its operation is based on communication standards and document or content standards that are freelypublished and widely implemented. For example, there are many types of browser, each in many cases implemented on several platforms; and there are many implementations of web servers. Any conformant browser can retrieve resources from any conformant server. So users have access to browsers on the majority of the devices that they use, from mobile phones to desktop computers.

Second, the Web is open with respect to the types of resource that can be published and shared on it. At its simplest, a resource on the Web is a web page or some other type of *content* that can be presented to the user, such as media files and documents in

Portable Document Format. If somebody invents, say, a new image-storage format, then images in this format can immediately be published on the Web. Users require a means of viewing images in this new format, but browsers are designed to accommodate new content-presentation functionality in the form of 'helper' applications and 'plug-ins'. The Web has moved beyond these simple data resources to encompass services, such as electronic purchasing of goods. It has evolved without changing its basic architecture. The Web is based on three main standard technological components:

- The HyperText Markup Language (HTML), a language for specifying the contents and layout of pages as they are displayed by web browsers;
- Uniform Resource Locators (URLs), also known as Uniform Resource Identifiers (URIs), which identify documents and other resources stored as part of the Web;
- A client-server system architecture, with standard rules for interaction (the HyperText

Transfer Protocol – HTTP) by which browsers and other clients fetch documents and other resources from web servers. Figure shows some web servers, and browsers making requests to them. It is an important feature that users may locate and manage their own web servers anywhere on the Internet.


Web servers
Browsers
www.google.com
http://www.google.com/search?q=obama
www.cdk5.net
Internet
GET http://www.cdk5.net
www.w3c.org
http://www.w3.org/standards/faq.html#conformance
File system of www.w3c.org
standards
faq.html

Centre for nuclear research (CERN), Switzerland, in 1989 as a vehicle for exchanging documents between a community of physicists connected by the Internet [Berners-Lee 1999]. A key feature of the Web is that it provides a hypertext structure among the documents that it stores, reflecting the users'.

HTML

The HyperText Markup Language [www.w3.org II] is used to specify the text and images that make up the contents of a webpage and to specify how they are laid out and formatted for presentation to the user. A web page contains such structured items as headings, paragraphs,tablesand images. HTML is also used to specify links and which resources are associated with them.

Users may produce HTML by hand, using a standard text editor, but they more commonly use an HTML-aware 'wysiwyg' editor that generates HTML from a layout that they create graphically. A typical piece of HTML text follows.

1.6 System Model

Q8. Explain about distributed system model.

Ans : **(Imp.)**

System models describe common properties and design choices for distributed system in a single descriptive model. The system models of distributed systems are classified into the following

- **Physical models:** It is the most explicit way in which to describe a system in terms of hardware composition.
- **Architectural models**: They describe a system in terms of the computational and communication tasks performed by its computational elements.
- **Fundamental models**: They examine individual aspects of a distributed system.

They are again classified based on some parameters as follows:

- **Interaction models:** This deals with the structure and sequencing of the communication between the elements of the system
- **Failure models:** This deals with the ways in which a system may fail to operate correctly
- **Security models:** This deals with the security measures implemented in the system against attempts to interfere with its correct operation or to steal its data.

1. Physical Models

A physical model is a representation of the underlying hardware elements of a distributed system hides the details of the computer and networking technologies employed.

There are many physical models available from the primitive baseline model to the complex models that could handle cloud environments.

Baseline physical model

- This is a primitive model that describes the hardware or software components located at networked computers.
- The communication and coordination of their activities is done by passing messages.
- This is a minimal physical model of a distributed system.
- This model could be extended to a set of computer nodes interconnected by a computer network for the required passing of messages.
- This model helps us to categorize the three generations of distributed systems.

Early distributed systems

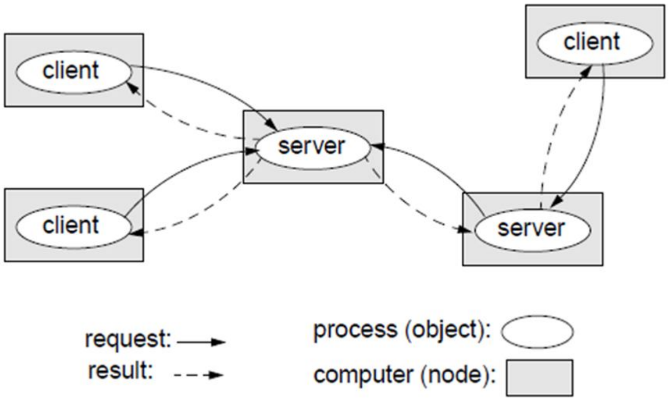
- The period is in the late 1970s and early 1980s.
- This was developed after the emergence LAN.

2. Architectural Models

Architectural model describes responsibilities distributed between system components and how are these components placed.

a) Client-server model

- The system is structured as a set of processes, called servers, that offer services to the users, called clients.
- The client-server model is usually based on a simple request/reply protocol, implemented with send/receive primitives or using remote procedure calls (RPC) or remote method invocation (RMI):
- The client sends a request (invocation) message to the server asking for some service;
- The server does the work and returns a result (e.g. the data requested) or an error code if the work could not be performed.

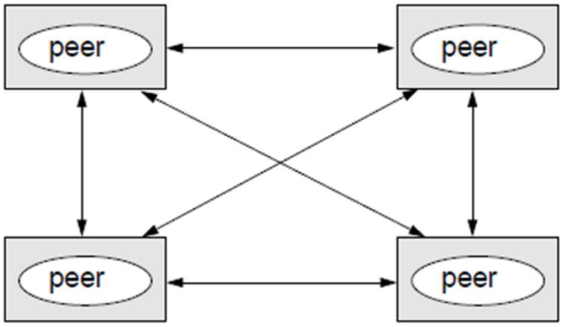


A server can itself request services from other servers; thus, in this new relation, the server itself acts like a client.

b) Peer-to-peer

All processes (objects) play similar role.

- Processes (objects) interact without particular distinction between clients and servers.
- The pattern of communication depends on the particular application.
- A large number of data objects are shared; any individual computer holds only a small part of the application database.
- Processing and communication loads for access to objects are distributed across many computers and access links.
- This is the most general and flexible model.



  - Peer-to-Peer tries to solve some of the above
  - It distributes shared resources widely -> share computing and communication loads.

Problems with peer-to-peer:

- High complexity due to
  - Cleverly place individual objects
  - Retrieve the objects
  - Maintain potentially large number of replicas.

3. Fundamental Models

They examine individual aspects of a distributed system. They are again classified based on some parameters as follows:

- **Client-Server Model:** In this model, the system consists of a set of clients that request services from a set of servers. The servers provide the requested services to the clients, and the clients do not interact directly with each other.
- **Peer-to-Peer Model:** In this model, all nodes in the system are equal and can act as both clients and servers. The nodes can interact directly with each other to share resources, data, and services.
- **Message-Passing Model:** In this model, nodes communicate with each other by exchanging messages. Each node can send a message to any other node in the system, and the nodes may or may not be aware of the existence of each other.
- **Event-Based Model:** In this model, nodes communicate by sending and receiving events. An event is a signal that indicates that a particular action has occurred, such as the arrival of a message or the completion of a task.
- **Publish-Subscribe Model:** In this model, nodes communicate by subscribing to and publishing events. A node can subscribe to a particular type of event, and any node that publishes an event of that type will send it to all subscribers.

4. Interaction Model

Interaction model are for handling time i. e. for process execution, message delivery, clock drifts etc.

- Synchronous distributed systems

Main features

- Lower and upper bounds on execution time of processes can be set.
- Transmitted messages are received within a known bounded time.
- Drift rates between local clocks have a known bound.

Important Consequences

1. In a synchronous distributed system there is a notion of global physical time (with a known relative precision depending on the drift rate).
2. Only synchronous distributed systems have a predictable behavior in terms of timing. Only such systems can be used for hard real-time applications.
3. In a synchronous distributed system it is possible and safe to use timeouts in order to detect failures of a process or communication link.

   It is difficult and costly to implement synchronous distributed systems.

Asynchronous Distributed Systems

Many distributed systems (including those on the Internet) are asynchronous. - No bound on process execution time (nothing can be assumed about speed, load, and reliability of computers). - No bound on message transmission delays (nothing can be assumed about speed, load, and reliability of interconnections) - No bounds on drift rates between local clocks.

Important Consequences

1. In an asynchronous distributed system there is no global physical time. Reasoning can be only in terms of logical time (see lecture on time and state).
2. Asynchronous distributed systems are unpredictable in terms of timing.
3. No timeouts can be used.

   Asynchronous systems are widely and successfully used in practice.

   In practice timeouts are used with asynchronous systems for failure detection.

   However, additional measures have to be applied in order to avoid duplicated messages, duplicated execution of operations, etc.

5. Fault Models

Failures can occur both in processes and communication channels. The reason can be both software and hardware faults.

Fault models are needed in order to build systems with predictable behavior in case of faults (systems which are fault tolerant) such a system will function according to the predictions, only as long as the real faults behave as defined by the “fault model”.

6. Security Models

This deals with the security measures implemented in the system against attempts to interfere with its correct operation or to steal its data.

- **Access Control Model:** This model specifies the rules and policies that determine who is allowed to access resources in a distributed system. It defines mechanisms for authentication, authorization, and accounting.
- **Confidentiality Model:** This model ensures that sensitive information is protected from unauthorized disclosure. It may involve encryption, decryption, or other techniques to ensure that only authorized users can access sensitive information.
- **Integrity Model:** This model ensures that data and resources are not modified or tampered with by unauthorized users. It may involve techniques such as digital signatures, checksums, and other mechanisms to ensure that data remains intact and has not been altered.
- **Availability Model:** This model ensures that resources are available to authorized users when they need them. It may involve techniques such as redundancy, fault tolerance, and load balancing to ensure that resources are always available.
- **Non-Repudiation Model:** This model ensures that a user cannot deny having performed a particular action. It may involve techniques such as digital signatures, timestamping, and other mechanisms to ensure that the actions of a user are recorded and cannot be denied later.

1.7 Inter Process Communication

Q9. Explain about inter process communication is distributed systems.

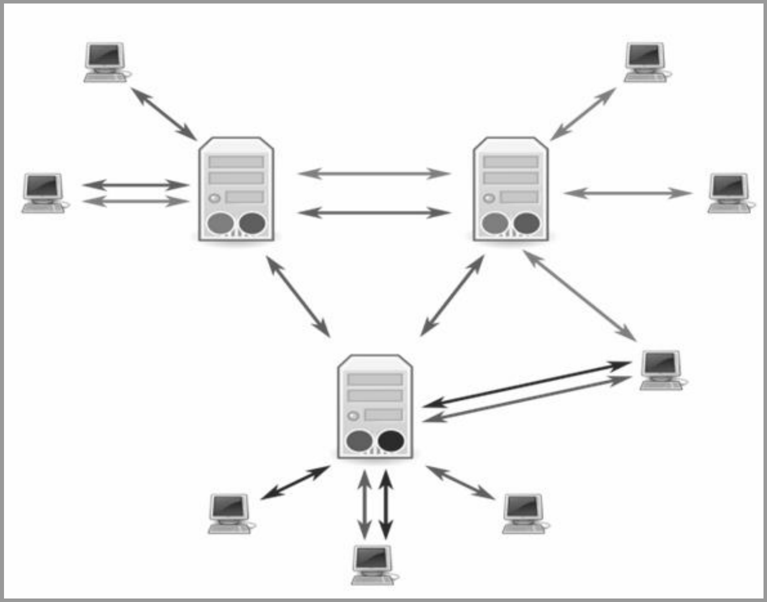
Ans : **(Imp.)**

Meaning

Interprocess Communication is a process of exchanging the data between two or more independent process in a distributed environment is called as Interprocess communication. Interprocess communi-cation on the internet provides both Datagram and stream communication.

Examples

1. N number of applications can communicate with the X server through network protocols.
2. Servers like Apache spawn child processes to handle requests.
3. Pipes are a form of IPC: grep foo file | sort



It has two functions:

1. **Synchronization**

   Exchange of data is done synchronously which means it has a single clock pulse.

2. **Message Passing**

   When processes wish to exchange information. Message passing takes several forms such as: pipes, FIFO, Shared Memory, and Message Queues.

Q10. State the Characteristics of Inter-process Communication

Ans :

There are mainly five characteristics of inter-process communication in a distributed environment/ system.

1. **Synchronous System Calls**

   In the synchronous system calls both sender and receiver use blocking system calls to transmit the data which means the sender will wait until the acknowledgment is received from the receiver and receiver waits until the message arrives.

2. **Asynchronous System Calls**

   In the asynchronous system calls, both sender and receiver use non-blocking system calls to transmit the data which means the sender doesn't wait from the receiver acknowledgment.

3. **Message Destination**

   A local port is a message destination within a computer, specified as an integer. Aport has exactly one receiver but many senders. Processes may use multiple ports from which to receive messages. Any process that knows the number of a port can send the message to it.

4. **Reliability**

   It is defined as validity and integrity.

5. **Integrity**

   Messages must arrive without corruption and duplication to the destination.

6. **Validity**

   Point to point message services are defined as reliable, If the messages are guaranteed to be delivered without being lost is called validity.

7. **Ordering**

   It is the process of delivering messages to the receiver in a particular order. Some applications require messages to be delivered in the sender order i.e., the order in which they were transmitted by the sender.

Q11. Explain the Methods in Interprocess Communication.

Ans :

These are the methods in IPC:

1. **Pipes (Same Process)**

   This allows flow of data in one direction only. Analogous to simplex systems (Keyboard). Data from the output is usually buffered until input process receives it which must have a common origin.

2. **Names Pipes (Different Processes)**

   This is a pipe with a specific name it can be used in processes that don't have a shared common process origin. E.g. is FIFO where the details written to a pipe is first named.

3. **Message Queuing**

   This allows messages to be passed between processes using either a single queue or several message queue. This is managed by system kernel these messages are coordinated using an API.

4. **Semaphores**

   This is used in solving problems associated with synchronization and to avoid race condition. These are integer values which are greater than or equal to 0.

5. **Shared Memory**

   This allows the interchange of data through a defined area of memory. Semaphore values have to be obtained before data can get access to shared memory.

6. **Sockets**

   This method is mostly used to communicate over a network between a client and a server. It allows for a standard connection which is computer and OS independent.

Q12. Explain message passing system in distributed system and write about the Issues in IPC By Message Passing in Distributed System.

Ans :

Message Passing

A message-passing system gives a collection of message-based IPC (Inter-Process Communication) protocols while sheltering programmers from the complexities of sophisticated network protocols and many heterogeneous platforms. The send() and receive()

communication primitives are used by processes for interacting with each other. For example, Process A wants to communicate with Process B then Process A will send a message with send() primitive and Process B will receive the message with receive() primitive.

Characteristics of a Good Message Passing System:

- Simplicity
- Uniform Semantics
- Efficiency
- Correctness
- Reliability
- Flexibility
- Security
- Portability

Issues in Message Passing

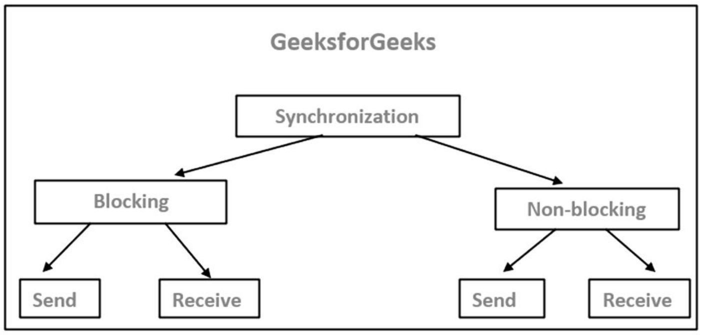
- Who is the message's sender?
- Who is the intended recipient?
- Is there a single receiver or several receivers?
- Is there any guarantee that the intended recipient received the message? Is it required to wait for the reply from the sender?
- Is there any strategy for handling a catastrophic event if it occurs during communication, such as a communication link failure or node crash?
- What should be done with the message if the receiver is not ready to take it? Whether it will be destroyed or kept in a buffer? What are the steps to follow if the buffer is also full in the case of buffering?
- Can a receiver be able to do the ordering of messages in case of several outstanding messages?

Q13. What is the process of synchronization in IPC? Explain.

Ans : **(Imp.)**

Synchronization

- The send() and receive() primitives are called whenever processes need to communicate.
- The communication primitives' synchronization of communicative processes is a critical issue in the communication structure.



Synchronization Semantics

The following are the two ways of message passing between processes:

- Blocking (Synchronous)
- Non-blocking (Asynchronous)

1. **Blocking:** The blocking semantics implies that when the call of a send () or receive() primitive blocks the invoker's current execution.

2. **Non-blocking:** The non-blocking semantics imply that when the call of a send () or receive() primitive does not block the invoker's current execution and the control immediately goes back to the invoker.

   - **Blocking send() primitive:** The blocking send() primitive refers to the blocking of sending process. The process remains blocking until it receives an acknowledgment from the receiver side that the message has been received after the execution of this primitive.
   - **Non-blocking send() primitive:** The non-blocking send() primitive refers to the non-blocking state of the sending process that implies after the execution of send() statement, the process is permitted to continue further with its execution immediately when the message has been transferred to a buffer.
   - **Blocking receive() primitive:** when they receive statement is executed, the receiving process is halted until a message is received.
   - **Nonblocking receive() primitive:** The non-blocking receive() primitive implies that the receiving process is not blocked after executing the receive() statement, control is returned immediately after informing the kernel of the message buffer's location.

The issue in a nonblocking receive() primitive is when a message arrives in the message buffer, how does the receiving process know? One of the following two procedures can be used for this purpose:

1. **Polling**

   In the polling method, the receiver can check the status of the buffer when a test primitive is passed in this method. The receiver regularly polls the kernel to check whether the message is already in the buffer.

2. **Interrupt**

   A software interrupt is used in the software interrupt method to inform the receiving process regarding the status of the message i.e. when the message has been stored into the buffer and is ready for usage by the receiver. So, here in this method, receiving process keeps on running without having to submit failed test requests.

In a blocked send() and receive() primitive, there is an issue:

Blocking send() primitive

The issue that can be raised here is that the sending process may become permanently halted if receiving process has crashed or the sent messages are lost because of communication failure. So, blocking send() primitives have set a fixed time value which when elapsed send operation is halted with an error status to avoid this problem. Users might be given the option to specify the timeout value as a parameter of the send primitive, or it could be set as a default.

Blocking receive() primitive

To avoid the receiving process from becoming halted indefinitely, a blocking receive() primitive might be connected with a fixed time value. This can happen if the prospective sending procedure fails or if the expected message is lost on the network owing to a communication breakdown.

1.8 The API For Internet Protocols

Q14. Explain about the API for the Internet protocols.

Ans : **(Imp.)**

The characteristics of inter process communication:

Message passing between a pair of processes can be supported by two message communication operations, send and receive, defined in terms of destinations and messages. To communicate, one process sends a message (a sequence of bytes) to a destination and another process at the destination receives the message. This activity involves the communication of data from the sending process to the receiving process and may involve the synchronization of the two processes

Synchronous and asynchronous communication

A queue is associated with each message destination. Sending processes cause messages to be added to remote queues and receiving processes remove messages from local queues. Communication between the sending and receiving processes may be either synchronous or asynchronous.

In the synchronousform of communication, the sending and receiving processes synchronize at very message. In this case, both send and receive are blocking operations. Whenever a send is issued the sending process (or thread) is blocked until the corresponding receive is issued. Whenever a receive is issued by a process (or thread), it blocks until a message arrives. In the asynchronous form of communication, the use of the send operation is nonblocking in that the sending process is allowed to proceed as soon as the message has been copied to a local buffer, and the transmission of the message proceeds in parallel with the sending process.

The receive operation can have blocking and non-blocking variants. In the non-blocking variant, the receiving process proceeds with its program after issuing a receive operation, which provides a buffer to be filled in the background, but it must separately receive notification that its buffer has been filled, by polling or interrupt.

1. Transmission Control Protocol (TCP)

TCP is a protocol that provides reliable and ordered delivery of data between nodes. It establishes a connection between two nodes and uses a flow control mechanism to ensure that data is transmitted at an appropriate rate. TCP segments data into packets, sends them, and waits for acknowledgments from the receiver. If an acknowledgment is not received, the data is retransmitted.

2. User Datagram Protocol (UDP)

UDP is a protocol that provides an unreliable and unordered delivery of data between nodes. Unlike TCP, UDP does not establish a connection before sending data. Instead, it sends packets called datagrams, which are not guaranteed to be delivered to the receiver.

3. Internet Protocol (IP)

IP is the protocol responsible for routing packets between nodes in a distributed system. It provides a unique address for each node on the network, called an IP address, which allows packets to be delivered to their intended destination. IP packets are sent in a best-effort manner, meaning that they may be lost, delayed, or delivered out of order.

4. Domain Name System (DNS)

DNS is a protocol used to translate human-readable domain names into IP addresses. When a user types a URL into a web browser, the browser sends a DNS query to a DNS server to retrieve the IP address associated with the domain name. This allows the browser to connect to the correct server to retrieve the requested content.

5. Simple Mail Transfer Protocol (SMTP)

SMTP is a protocol used for sending email messages between nodes in a distributed system. It defines the rules for how email messages are formatted, transmitted, and received. SMTP uses a store-and-forward mechanism to deliver email, meaning that messages are temporarily stored on intermediate servers until they can be delivered to their final destination.

6. Hypertext Transfer Protocol (HTTP)

HTTP is a protocol used for transferring data over the World Wide Web. It provides a request-response mechanism for fetching resources such as web pages, images, and videos. When a user clicks on a link or types a URL into a browser, the browser sends an HTTP request to the server hosting the resource. The server then sends an HTTP response containing the requested resource.

1.9 External Data Representation and Multicast Communication

Q15. Explain about External data Representation in Distributed Systems.

Ans : **(Imp.)**

External Data Representation (XDR) and Multicast communication are two important concepts in distributed systems. Here's an overview of each:

1. External Data Representation (XDR)

XDR is a protocol for encoding data in a platform-independent format that can be transmitted between different computer systems. It defines a standard way of representing data types such as integers, floating-point numbers, and strings, as well as more complex data structures such as arrays and structures. XDR allows data to be transmitted between systems that use different byte orders and word sizes, and is commonly used in distributed systems for interprocess communication and remote procedure calls.

2. Multicast Communication

Multicast is a communication method that allows data to be transmitted to multiple recipients simultaneously. Unlike unicast, which sends data to a single recipient, and broadcast, which sends data to all recipients on a network segment, multicast sends data to a specific group of recipients. Multicast is used in distributed systems for applications such as video streaming, online gaming, and group chat. It provides a more efficient use of network bandwidth than unicast, as the data is only transmitted once and is then distributed to all members of the multicast group.

In combination, XDR and Multicast can be used to implement efficient and scalable distributed systems. For example, in a video streaming application, XDR can be used to encode the video data in a platform-independent format, and multicast can be used to distribute the data to multiple recipients simultaneously. This reduces the network bandwidth required to stream the video, as the data is transmitted once and then distributed to all recipients. XDR and Multicast are just two of many tools available for building distributed systems, but they are important ones that can be used to optimize performance and scalability.

Short Question and Answers

1. What is distributed system?

Ans :

A distributed system is a network that consists of autonomous computers that are connected using a distribution middleware. They help in sharing different resources and capabilities to provide users with a single and integrated coherent network.

2. Trends in distributed systems.

Ans :

(i) Internet

The modern Internet is a vast interconnected collection of computer networks of many different types, with the range of types increasing all the time and now including, for example, a wide range of wireless communication technologies such as WiFi, WiMAX, Bluetooth and third-generation mobile phone networks. The net result is that networking has become a pervasive resource and devices can be connected (if desired) at any time and in any place.

The Internet is also a very large distributed system. It enables users, wherever they are, to make use of services such as the World Wide Web, email and file transfer.

(ii) Intranet

- ➢ A portion of the Internet that is separately administered and has a boundary that can be configured to enforce local security policies
- ➢ Composed of several LANs linked by backbone connections
- ➢ Be connected to the Internet via a router

Main issues in the design of components for the use in intranet

- ➢ File services
- ➢ Firewall
- ➢ The cost of software installation and support

(iii) Mobile and Ubiquitous Computing

Technological advances in device miniaturization and wireless networking have led increasingly to the integration of small and portable computing devices into distributed systems. These devices include:

- ➢ Laptop computers.
- ➢ Handheld devices, including mobile phones, smart phones, GPS-enabled devices, pagers, personal digital assistants (PDAs), video cameras and digital cameras.
- ➢ Wearable devices, such as smart watches with functionality similar to a PDA.
- ➢ Devices embedded in appliances such as washing machines, hi-fi systems, cars and refrigerators.

3. Resource Sharing in Distributed System

Ans :

i) Data Migration

The process in which data is transferred from one location to another location in the system. Data is brought to the location of computation, that needs to be accessed by a distributed system. We can also say that the data is migrated from destination part to source, after requesting source to destination.

ii) Computation Migration

The process in which computation is transferred rather than the data across the system. The files cannot be transferred to the computation point due to large file size or big data file. So, the computation data is transferred to the data file point and processes the computation and the result is transferred to the computation point.

Advantages

- ➢ Increases computational speed.
- ➢ **Load Balancing:** It helps spread the load across the distributed system in order to optimize resource sharing.

Here are some key considerations for resource sharing in distributed computing:

(a) Resource allocation

In distributed computing, resources such as processing power, memory, and storage are allocated across the network to meet the needs of various applications and users. Efficient resource allocation is crucial to ensure that each application or user has access to the resources they need, without overburdening any one computer or server.

(b) Load balancing

Resource sharing also involves load balancing, which is the process of distributing workloads across the network to optimize resource utilization. Load balancing helps to prevent overloading of individual computers or servers, which can cause performance issues and slow down the entire system.

(c) Resource discovery

In a distributed computing environment, resources may be located on different computers or servers across the network. Resource discovery tools are used to locate and access resources as needed, enabling efficient resource sharing across the network.

4. Challenges in distributed computing.

Ans :

i) Network connectivity

Distributed computing relies on a network of interconnected computers and servers, which can be vulnerable to connectivity issues such as latency, packet loss, and network congestion. These issues can cause delays, failures, and other performance problems that can affect the reliability of the distributed system.

ii) Data consistency and replication

Distributed computing often involves the replication of data across multiple servers, which can present challenges for ensuring data consistency and integrity. Synchronization mechanisms and other techniques are needed to ensure that data updates are propagated correctly across all nodes in the network.

iii) Fault tolerance and resilience

Distributed computing systems must be designed to be fault-tolerant and resilient, meaning that they can continue to operate even if one or more nodes in the network fail or become unavailable. This requires the use of redundancy and other techniques to ensure that critical functions and services can continue to operate even in the event of failures.

iv) Security and privacy

Distributed computing systems often involve the sharing of sensitive data and resources across the network, making them vulnerable to security threats such as hacking, data breaches, and other cyber attacks. Robust security measures and protocols must be implemented to ensure the confidentiality, integrity, and availability of data and resources.

5. Web challenges in distributed computing.

Ans :

i) Scalability

The feature of distributed systems in which if the load of a system increases the performance of the system will not be degraded.

ii) Heterogeneity

The ability to communication with different devices. For example, communication between computer to mobile or other peripheral devices.

iii) Security Challenges

There are three types of Security Challenges, i.e. Privacy, Authentication, Availability.

- **Privacy:** Data Shared should maintain confidentiality.
- **Authentication:** Whoever shares the message should have proper identity for that system. Unauthorised users should not access the system.
- **Availability:** Data and Resources should be available.

iv) Handling of Failure

There are three types of failures that can occur in a system which need to be fixed.

6. WWW.

Ans :

The World Wide Web is an evolving system for publishing and accessing resources and services across the Internet. Through commonly available web browsers, users retrieve and view documents of many types, listen to audio streams and view video streams, and interact with an unlimited set of services. The Web began life at the European requirement to organize their knowledge. This means that documents contain *links* (or *hyperlinks*) – references to other documents and resources that are also stored in the Web. It is fundamental to the user's experience of the Web that when they encounter a given image or piece of text within a document, this will frequently be accompanied by links to related documents and other resources. The structure of links can be arbitrarily complex and the set of resources that can be added is unlimited – the 'web' of links is indeed world-wide.

7. Inter process communication

Ans :

Interprocess Communication is a process of exchanging the data between two or more independent process in a distributed environment is called as Interprocess communication. Interprocess communi-cation on the internet provides both Datagram and stream communication.

8. Characteristics of Inter-process Communication

Ans :

i) Synchronous System Calls

In the synchronous system calls both sender and receiver use blocking system calls to transmit the data which means the sender will wait until the acknowledgment is received from the receiver and receiver waits until the message arrives.

ii) Asynchronous System Calls

In the asynchronous system calls, both sender and receiver use non-blocking system calls to transmit the data which means the sender doesn't wait from the receiver acknowledgment.

iii) Message Destination

A local port is a message destination within a computer, specified as an integer. Aport has exactly one receiver but many senders. Processes may use multiple ports from which to receive messages. Any process that knows the number of a port can send the message to it.

iv) Reliability

It is defined as validity and integrity.

v) Integrity

Messages must arrive without corruption and duplication to the destination.

9. External Data Representation

Ans :

XDR is a protocol for encoding data in a platform-independent format that can be transmitted between different computer systems. It defines a standard way of representing data types such as integers, floating-point numbers, and strings, as well as more complex data structures such as arrays and structures. XDR allows data to be transmitted between systems that use different byte orders and word sizes, and is commonly used in distributed systems for interprocess communication and remote procedure calls.

10. Multicast Communication

Ans :

Multicast is a communication method that allows data to be transmitted to multiple recipients simultaneously. Unlike unicast, which sends data to a single recipient, and broadcast, which sends data to all recipients on a network segment, multicast sends data to a specific group of recipients. Multicast is used in distributed systems for applications such as video streaming, online gaming, and group chat. It provides a more efficient use of network bandwidth than unicast, as the data is only transmitted once and is then distributed to all members of the multicast group.

Choose the Correct Answers

1. In which type of distributed system does each node perform the same function and there is no concept of a central server? [b]

   (a) Client-server architecture (b) Peer-to-peer architecture

   (c) Centralized architecture (d) Hybrid architecture

2. Which of the following is a characteristic of a distributed system? [d]

   (a) Decentralization (b) Scalability

   (c) Fault tolerance (d) All of the above

3. Which type of distributed system is best suited for applications that require high availability and fault tolerance? [a]

   (a) Client-server architecture (b) Peer-to-peer architecture

   (c) Centralized architecture (d) Hybrid architecture

4. What is the primary challenge in implementing resource sharing? [a]

   (a) Ensuring fair distribution of resources (b) Maximizing resource utilization

   (c) Maintaining resource compatibility (d) Minimizing resource availability

5. Which of the following is not a benefit of resource sharing? [d]

   (a) Cost savings (b) Improved performance

   (c) Reduced complexity (d) Increased security risks

6. What is a common challenge in fault tolerance in distributed systems? [a]

   (a) Detecting the fault (b) Preventing the fault

   (c) Ignoring the fault (d) All of the above

7. In which distributed system model does the server provide services to multiple clients? [a]

   (a) Client-server model (b) Peer-to-peer model

   (c) Hierarchical model (d) Mesh model

8. Which of the following is not an example of IPC? [d]

   (a) Pipes (b) Sockets

   (c) Shared memory (d) Remote procedure calls

9. Which of the following is not a type of message passing used in IPC? [c]

   (a) Synchronous message passing (b) Asynchronous message passing

   (c) Buffered message passing (d) Unbuffered message passing

10. Which of the following is not a type of message passing in distributed systems? [d]

    (a) Unicast (b) Multicast

    (c) Broadcast (d) Polling

Fill in the Blanks

1. A ________ is a network that consists of autonomous computers that are connected using a distribution middleware.

2. A collection of connected computers that work together as a unit to perform operations together, functioning in a single system is called as ________

3. Sharing computing resources like memory and CPU is called ________

4. ________ helps spread the load across the distributed system in order to optimize resource sharing.

5. ________ describe common properties and design choices for distributed system in a single descriptive model.

6. A ________ is a representation of the underlying hardware elements of a distributed system hides the details of the computer and networking technologies employed.

7. ________ describes responsibilities distributed between system components and how are these components placed.

8. ________ used in solving problems associated with synchronization and to avoid race condition

9. ________ allows processes running on different hosts to call procedures on each other as if they were running on the same host.

10. ________ is a protocol for encoding data in a platform-independent format that can be transmitted between different computer systems

Answers

1. Distributed system
2. Cluster computing
3. Resource sharing
4. Load Balancing
5. System models
6. Physical model
7. Architectural model
8. Semaphores
9. RPC
10. XDR

UNIT II

Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation and Objects: Remote Invocation – Introduction – Request-reply protocols – Remote procedure call - Remote method invocation.

Case study: Java RMI - Group communication - Publish-subscribe systems - Message queues - Shared memory approaches - Distributed objects - Case study: Enterprise Java Beans - from objects to components.

2.1 Network Virtualization

2.1.1 Overlay Networks

Q1. What is an overlay network? Explain its applications.

Ans : **(Imp.)**

Meaning

An overlay network is a logical network that is built on top of an existing physical network infrastructure. It is used to provide additional functionality or services beyond what is available in the underlying network, such as increased security, better performance, or enhanced routing capabilities.

In an overlay network, each node in the logical network is associated with one or more nodes in the physical network. The overlay nodes communicate with each other using the physical network as a transport mechanism, but they also use additional protocols and mechanisms to implement the overlay network functionality.

One of the key advantages of overlay networks is that they can be designed and deployed independently of the underlying physical network infrastructure. This means that new overlay networks can be added or modified without requiring any changes to the underlying network, which can be especially useful in environments where the physical network is managed by a different organization or is subject to strict security or regulatory requirements.

Overlay networks are commonly used in virtualization environments, where multiple virtual machines or containers need to communicate with each other in a secure and isolated manner. They are also used in peer-to-peer networking, content distribution networks, and other distributed systems where nodes need to communicate and collaborate across a large-scale network.

Applications

Overlay networks are used in a variety of applications where additional functionality or services are needed beyond what is available in the underlying physical network infrastructure. Some common applications of overlay networks include:

1. **Virtualization**

   Overlay networks are used in virtualization environments to provide secure and isolated communication between virtual machines or containers.

2. **Content Delivery Networks (CDNs)**

   CDNs use overlay networks to distribute content across a large-scale network of servers to improve performance and availability.

3. **Peer-to-Peer (P2P) Networking**

   P2P networks use overlay networks to enable direct communication and file sharing between nodes without the need for a centralized server.

4. **Cloud Computing**

   Overlay networks are used in cloud computing environments to provide network isolation and security between different tenants or users.

5. **Software-Defined Networking (SDN)**

   SDN uses overlay networks to abstract the underlying physical network infrastructure and enable more flexible and programmable network management.

6. **Internet of Things (IoT)**

   Overlay networks can be used in IoT environments to enable secure and efficient communication between IoT devices and cloud-based services.

7. **Distributed Databases**

   Overlay networks can be used to provide replication and fault-tolerance in distributed database systems.

2.2 Case Study: Mpi

Q2. Explain about the case study on MPI.

Ans :

Message Passing Interface (MPI) is a standardized and portable message-passing system developed for distributed and parallel computing. MPI provides parallel hardware vendors with a clearly defined base set of routines that can be efficiently implemented. As a result, hardware vendors can build upon this collection of standard low-level routines to create higher-level routines for the distributed-memory communication environment supplied with their parallel machines.

MPI gives users the flexibility of calling a set of routines from C, C++, Fortran, C#, Java, or Python. The advantages of MPI over older message passing libraries are portability (because MPI has been implemented for almost every distributed memory architecture) and speed (because each implementation is in principle optimized for the hardware on which it runs)

The advantages of MPI over other message-passing framework is portability and speed. It has been implemented for almost every distributed memory architecture and each implementation is in principle optimized for the hardware on which it runs.

Even though there are options available for multiple languages, Python is the most preferred one due to its simplicity, and ease of writing the code. So, now, we will now look at how to install MPI on ubuntu 14.10.

Message Passing Interface (MPI) is a popular library used for parallel programming in distributed systems. It allows programs to communicate and coordinate across multiple computing nodes, which can be located on different machines connected by a network.

One example of the use of MPI in distributed systems is in weather forecasting. Weather simulations are complex and computationally intensive, and require large amounts of processing power to run efficiently. By using MPI, weather models can be divided into smaller sub-models that can be run on different computing nodes simultaneously. The results of each sub-model can then be combined to create a more accurate overall forecast.

Another example of MPI in distributed systems is in bioinformatics research. Many bioinformatics algorithms require large amounts of processing power and memory to analyze complex genetic data. By using MPI, these algorithms can be run on multiple computing nodes, allowing researchers to analyze data more quickly and efficiently.

MPI has also been used in high-performance computing applications, such as simulations of fluid dynamics, combustion, and other physical phenomena. In these applications, large-scale simulations can be divided into smaller sub-simulations that can be run on different computing nodes using MPI.

In conclusion, MPI is a powerful tool for parallel programming in distributed systems, and has been used in a wide range of applications, from weather forecasting to bioinformatics research and high-performance computing. By allowing programs to communicate and coordinate across multiple computing nodes, MPI enables faster and more efficient processing of complex data and simulations.

2.3 Remote Method Invocation and Objects

2.3.1 Remote Invocation

Q3. Explain about remote method invocation.

Ans :

Remote invocation, also known as remote procedure call (RPC), is a method used to enable communication between different processes or systems over a network. With remote invocation, a client can invoke a method or function on a remote server as if it were local, allowing the client and server to interact with each other seamlessly.

The client sends a message to the server, which includes the name of the method or function to be executed, as well as any necessary arguments. The server then executes the requested method or function and returns a response to the client, which can include a return value, an error message, or other information.

Remote invocation can be implemented using various protocols such as HTTP, TCP/IP, or UDP, and can be used in a variety of applications including distributed systems, client-server architectures, and web services. It enables the building of complex systems that can be distributed across multiple locations and platforms, making it an essential technology in modern computing.

2.3.1.1 Introduction

Q4. Give the introduction about RPC and distributed objects?

Ans : **(Imp.)**

Distributed objects are objects that are distributed across different address spaces, either in multiple computers connected via a network or even indifferent processes on the same computer, but which work together by sharing data and invoking methods.

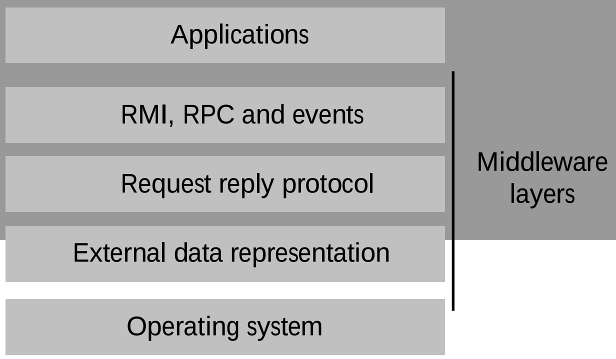


Fig. Middle ware layer

- This often involves location transparency, where remote objects appear the same as local objects.
- The main method of distributed object communication is with remote method invocation Invoking a method on a remote object is known as remote method invocation) generally by message-passing
- Message-passing: one object sends a message to another object in a remote machine or process to perform some task.

- The results are sent back to the calling object.
- The remote procedure call (RPC) approach extends the common programming Abstraction of the procedure call to distributed environments, allowing a calling Process to call a procedure in a remote node as if it is local.
- Remote method invocation (RMI) is similar to RPC but for distributed objects,with Added benefits in terms of using object-oriented programming concepts in Distributed systems and also extending the concept of an object reference to the Global distributed environments, and allowing the use of object references as Parameters in remote invocations.
- Remote procedure call – client calls the procedures in a server program that is running in a different process
- Remote method invocation (RMI) – an object in one process can invoke methods of objects in another process.
- Event notification – objects receive notification of events at other objects for which they have registered Middleware Roles provide high-level abstractions such as RMI enable location transparency free from specifics of communication protocols.

Communication between Distributed Objects and Other Objects

- **Life cycle**

  Creation, migration and deletion of distributed objects is different from local objects
- **Reference**

  Remote references to distributed objects are more complex than simple pointers to memory addresses.
- **Request Latency**

  A distributed object request is orders of magnitude slower than local method invocation Object
- **Activation**

  Distributed objects may not always be available to serve an object request at any point in time.
- **Parallelism**

  Distributed objects may be executed in parallel.
- **Communication**

  There are different communication primitives available for distributed objects requests
- **Failure**

  Distributed objects have far more points of failure than typical local objects.
- **Security**

  Distribution makes them vulnerable to attack.

2.3.1. 2 Request Reply Protocols

Q5. Explain about communication protocols for RPC.

Ans :

Communication Protocols for Remote Procedure Calls

The following are the communication protocols that are used:

- Request Protocol
- Request/Reply Protocol
- The Request/Reply/Acknowledgement-Reply Protocol

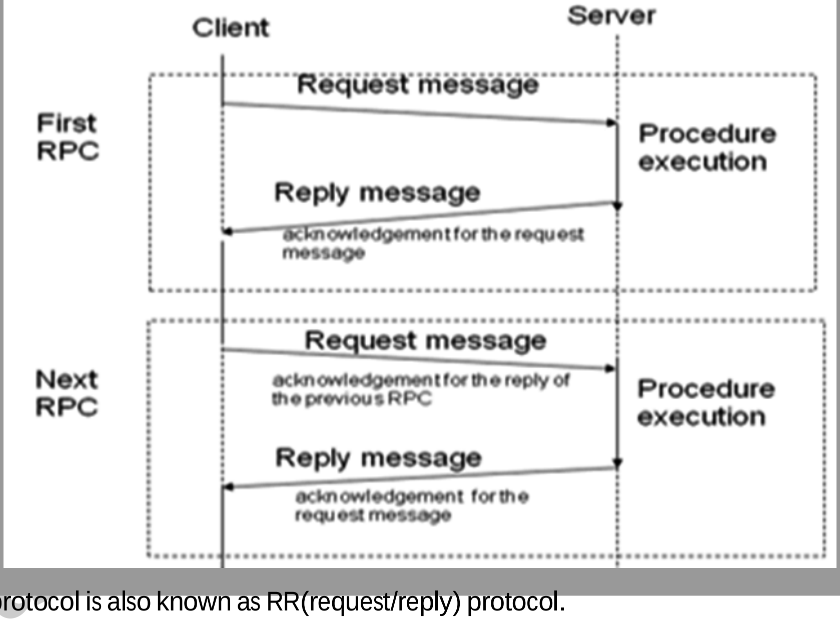
(i) Request Protocol

Based on the needs of different systems, several communication protocols have been proposed for use in RPC which are mentioned below:

- This protocol is also called as R (request) protocol.
- It is used in RPC when a called procedure has nothing to return as a result of execution and the requirement of client confirmation about procedure execution is not needed.
- As no acknowledgement or reply message is involved, only single message is transmitted from client to server.

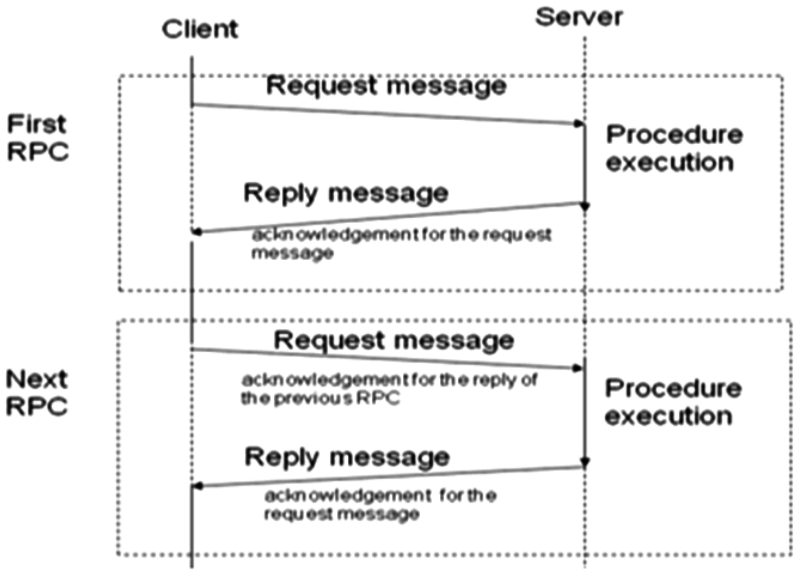
- The client proceeds after the request message is sent as there is no reply message.
- This protocol provides May-be call semantics and does not need retransmission of request messages.
- RPC that uses the R protocol is known as asynchronous RPC which helps to improve the combined performance of the client and server. This is done because the client does not ait for a reply and server does not need to send a reply.
- For an asynchronous RPC, the RPCRuntime does not retry a request in case of communication failure. TCP is better alternative then UDP since no retransmission is required and it is connection oriented.
- Asynchronous RPC with unreliable transport protocol are generally used in implementing periodic update services. Distributed system window is one of its applications.

(ii) The Request/Reply Protocol

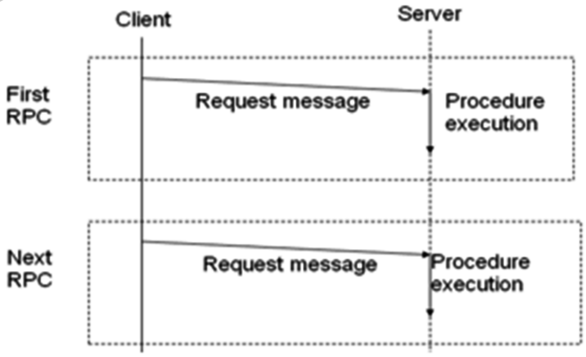

Client
Server
First RPC
Request message
Procedure execution
Reply message
acknowledgement for the request message
Next RPC
Request message
acknowledgement for the reply of the previous RPC
Procedure execution
Reply message
acknowledgement for the request message

- This protocol is also known as RR(request/reply) protocol.
- It is useful for designing systems which involve simple RPCs.
- In a simple RPC all the arguments and result fit in a single packet buffer while the call duration and intervals between calls are short.
- This protocol is based on the idea of using implicit acknowledgement to eliminate explicit acknowledgement messages.
- In this protocol a server reply is considered as an ACK for a clients request and a subsequent call from a client is considered as ACK of the client’s previous call.
- Timeout-and-retires technique is used with RR protocol for failure handling. Retransmission of request message is done when there is no response.
- RR protocol and timeout technique provides at-least-once call semantics on if duplicate requests are not filtered out.
- Exactly once semantics are supported by servers using reply cache which stores replies.

(iii) The Request/Reply/Acknowledgement-Reply Protocol


Client
Server
First RPC
Request message
Procedure execution
Reply message
acknowledgement for the request message
Next RPC
Request message
acknowledgement for the reply of the previous RPC
Procedure execution
Reply message
acknowledgement for the request message

- This protocol is also known as RRA (request/reply/acknowledge-reply) protocol.
- RR protocol implements exactly once semantics which requires storage of a lot of information in the server cache and can lead to loss of replies that have not been delivered.
- To overcome the limitations or RR protocol, RSA protocol is used.
- In this clients acknowledge the receipt of reply messages and the server deletes information from its cache only after it receives an acknowledgement from client.
- Sometimes the reply acknowledgement message may get lost therefore RRA protocol needs a unique ordered message identifiers. This keeps a track of the acknowledgement series sent.



2.3.1.3 Remote Procedure Call

Q6. Explain the working procedure of remote procedure call.

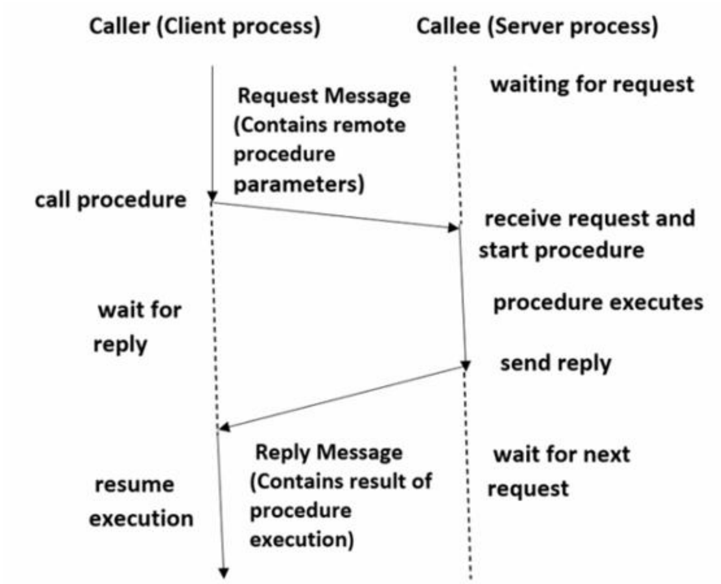
Ans :

Remote Procedure Call (RPC) is a communication technology that is used by one program to make a request to another program for utilizing its service on a network without even knowing the network's details. A function call or a subroutine call are other terms for a procedure call.

It is based on the client-server concept. The client is the program that makes the request, and the server is the program that gives the service. An RPC, like a local procedure call, is based on the synchronous operation that requires the requesting application to be stopped until the remote process returns its results. Multiple RPCs can be executed concurrently by utilizing lightweight processes or threads that share the same address space. Remote Procedure Call program as often as possible utilizes the Interface Definition Language (IDL), a determination language for describing a computer program component's Application Programming Interface (API). In this circumstance, IDL acts as an interface between machines at either end of the connection, which may be running different operating systems and programming languages.

Working Procedure for RPC Model

- The process arguments are placed in a precise location by the caller when the procedure needs to be called.
- Control at that point passed to the body of the method, which is having a series of instructions.
- The procedure body is run in a recently created execution environment that has duplicates of the calling instruction's arguments.
- At the end, after the completion of the operation, the calling point gets back the control, which returns a result.
- The call to a procedure is possible only for those procedures that are not within the caller's address space because both processes (caller and callee) have distinct address space and the access is restricted to the caller's environment's data and variables from the remote procedure.
- The caller and callee processes in the RPC communicate to exchange information via the message-passing scheme.
- The first task from the server-side is to extract the procedure's parameters when a request message arrives, then the result, send a reply message, and finally wait for the next call message.
- Only one process is enabled at a certain point in time.
- The caller is not always required to be blocked.
- The asynchronous mechanism could be employed in the RPC that permits the client to work even if the server has not responded yet.
- In order to handle incoming requests, the server might create a thread that frees the server for handling consequent requests.



Q7. Explain different types of RPC.

Ans : **(Imp.)**

(i) Callback RPC

In a Callback RPC, a P2P (Peer-to-Peer)paradigm opts between participating processes. In this way, a process provides both client and server functions which are quite helpful. Callback RPC's features include:

- The problems encountered with interactive applications that are handled remotely
- It provides a server for clients to use.
- Due to the callback mechanism, the client process is delayed.
- Deadlocks need to be managed in callbacks.
- It promotes a Peer-to-Peer (P2P) paradigm among the processes involved.

(ii) RPC for Broadcast

A client's request that is broadcast all through the network and handled by all servers that possess the method for handling that request is known as a broadcast RPC. Broadcast RPC's features include:

- You have an option of selecting whether or not the client's request message ought to be broadcast.
- It also gives you the option of declaring broadcast ports.
- It helps in diminishing physical network load.

(iii) Batch-mode RPC

Batch-mode RPC enables the client to line and separate RPC inquiries in a transmission buffer before sending them to the server in a single batch over the network. Batch-mode RPC's features include:

- It diminishes the overhead of requesting the server by sending them all at once using the network.
- It is used for applications that require low call rates.
- It necessitates the use of a reliable transmission protocol.

(iv) Local Procedure Call Vs Remote Procedure Call

- Remote Procedure Calls have disjoint address space i.e. different address space, unlike Local Procedure Calls.
- Remote Procedure Calls are more prone to failures due to possible processor failure or communication issues of a network than Local Procedure Calls.
- Because of the communication network, remote procedure calls take longer than local procedure calls.

Q8. State the advantages and disadvantages of RPC?

Ans :

Advantages

- The technique of using procedure calls in RPC permits high-level languages to provide communication between clients and servers.
- This method is like a local procedure call but with the difference that the called procedure is executed on another process and a different computer.
- The thread-oriented model is also supported by RPC in addition to the process model.
- The RPC mechanism is employed to conceal the core message passing method.
- The amount of time and effort required to rewrite and develop the code is minimal.
- The distributed and local environments can both benefit from remote procedure calls.
- To increase performance, it omits several of the protocol layers.
- Abstraction is provided via RPC. To exemplify, the user is not known about the nature of message-passing in network communication.
- RPC empowers the utilization of applications in a distributed environment.

Disadvantages of Remote Procedure Calls

- In Remote Procedure Calls parameters are only passed by values as pointer values are not allowed.
- It involves a communication system with another machine and another process, so this mechanism is extremely prone to failure.

- The RPC concept can be implemented in a variety of ways, hence there is no standard.
- Due to the interaction-based nature, there is no flexibility for hardware architecture in RPC.
- Due to a remote procedure call, the process's cost has increased.

2.3.1.4 Remote Method Invocation

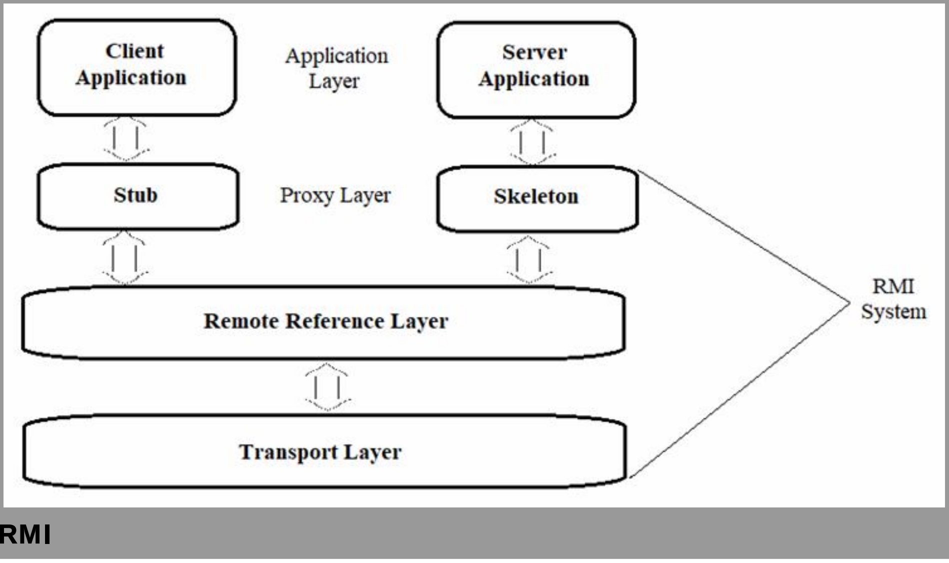
Q9. Explain briefly about RMI state in architecture and working mechanism.

Ans : **(Imp.)**

- The RMI stands for Remote Method Invocation is an API mechanism.
- This mechanism allows an object residing in one system (JVM) to access an object running on another JVM.
- This mechanism generally creates distributed applications in java.
- The RMI provides remote communication between the java applications using two objects called stub and skeleton.

The Architecture of Remote Method Invocation (RMI)

- A remote object is an object whose method can be invoked from another JVM.
- Java RMI application contains two types of programs such as server program and client program.
- In the server-side program, a remote object is created, and a reference of that remote object is made available for the client-side using registry.
- The client-side program requests the remote objects on the server and tries to invoke its methods.



Working of RMI

- When the client makes a call to the remote object, it is received by the stub which eventually passes this request to the RRL.
- When the client-side RRL receives the request, it invokes a method called invoke() of the object remoteRef. It passes the request to the RRL on the server-side.
- The RRL on the server-side passes the request to the Skeleton (proxy on the server) which finally invokes the required object on the server.
- The result is passed all the way back to the client.

Q10. State the Components of RMI Architecture

Ans :

(i) Stub

- The stub is an object, acts as a gateway for the client-side.
- All the outgoing requests are routed through it.
- It resides at the client-side and represents the remote object.
- When the caller invokes a method on the stub object, it does the following tasks:
- It initiates a connection with remote Virtual Machine (JVM).
- It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM).
- It waits for the result.
- It reads (unmarshals) the return value or exception, and
- It finally, returns the value to the caller.

(ii) Skeleton

- The skeleton is an object, acts as a gateway for the server-side object.
- All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:
- It reads the parameter for the remote method
- It invokes the method on the actual remote object, and
- It writes and transmits (marshals) the result to the caller.

(iii) Transport Layer

- This layer connects the client and the server.
- It manages the existing connection and also sets up new connections.
- It is responsible for the transportation of data from one machine to another.
- The default connection is set up only in the TCP/IP protocol.

(iv) RRL(Remote Reference Layer)

- It is the layer that manages the references made by the client to the remote object.
- This layer gets a stream-oriented connection from the transport layer.
- It is responsible for dealing with the semantics of remote invocations.
- It is also responsible for handling duplicated objects and for performing implementation-specific tasks with remote objects.

2.4 Case Stydt: Java Rmi

Q11. Write a Case Study on JAVA RMI.

Ans : **(Imp.)**

Java RMI (Remote Method Invocation) is a Java API that provides a mechanism for creating distributed Java applications. It allows Java objects to invoke methods on remote Java objects located on different JVMs (Java Virtual Machines) running on different machines. In this case study, we will look at how Java RMI was used to develop a distributed application.

Problem Statement

A company has a distributed system with several nodes running on different machines. The system is used to manage inventory for their retail stores. The system has several modules, including a billing module, a stock management module, and a customer module. The company wants to make some changes to the stock management module to improve its performance and scalability. The existing module is a monolithic application, and the company wants to refactor it into a distributed application.

Solution

The company decided to refactor the stock management module into a distributed application using Java RMI. The new application consists of a server and multiple clients. The server runs on a dedicated machine, and the clients run on different machines. The server exposes remote objects that can be accessed by the clients using Java RMI. The server maintains a local copy of the inventory data, which is shared with the clients. Whenever a client wants to modify the inventory data, it sends a request to the server, which updates the inventory data and sends the updated data back to all the clients.

To implement this solution, the company developed the following Java classes:

1. **StockManagementServer**

   This is the main class that starts the server. It creates an instance of the Stock Management Service Impl class and binds it to the RMI registry.

2. **Stock Management Service IMPL**

   This class implements the Stock Management Service interface, which defines the remote methods that can be called by the clients. The implementation maintains a local copy of the inventory data and provides methods to update the inventory data.

3. **Stock Management Service**

   This is the remote interface that defines the methods that can be called by the clients. It extends the java.rmi.Remote interface and declares the methods that can be called remotely.

4. **Stock Management Client**

   This class represents the client that interacts with the server. It obtains a reference to the remote StockManagementService object from the RMI registry and calls its methods to modify the inventory data.

The company tested the application using multiple clients running on different machines, and the results were satisfactory. The new distributed application improved the performance and scalability of the stock management module, and it was easier to maintain and update.

Conclusion

Java RMI is a powerful API that can be used to develop distributed Java applications. It provides a simple and efficient mechanism for invoking methods on remote objects. In this case study, we saw how Java RMI was used to refactor a monolithic stock management module into a distributed application. The new application improved the performance and scalability of the system, and it was easier to maintain and update.

2.4.1 Group Communication

Q12. Write a note on Group Communication in distributed systems.

Ans : **(Imp.)**

Meaning

Communication between two processes is essential to transfer data between the processes, such as code or a file.

Group Communication occurs when a single source process simultaneously attempts to communicate with numerous functions.

A group is an abstract collection of interrelated operations. This abstraction hides the message passing such that the communication seems to be a standard procedure call. Group communication also enables processes from separate hosts to collaborate and conduct activities in a coordinated way, improving overall system performance.

Different areas of Group Communication

Types of group communication

As illustrated above, there are three types of group communication. They are defined below:

1. **Broadcast Communication**

   This occurs when the host simultaneously attempts to communicate with all the processes in a distributed system. It is helpful when a consistent information stream must be supplied to all methods effectively. Communication is highly rapid compared to other means of communication since it does not require processing. It doesn't, however, support many operations and cannot address each function independently.

   P1 process communicating with every process in the system

2. **Multicast Communication**

   The host process attempts to simultaneously interact with a specific set of operations in a distributed system. This approach is mainly used to discover solutions to the high burden on the host system and duplicate information from system processes. Multitasking can considerably reduce the time required to handle messages.

   P1 process communicating with only a group of the process in the system.

3. Unicast Communication

This occurs when the host process attempts to interact with a single operation in a distributed system simultaneously. It works well for two processes interacting since it treats one way. However, it incurs costs since it must first determine the specific procedure and then communicate information.

P1 process communicating with only P5 process

Q13. State characteristics of Group Communication.

Ans :

Atomicity, often known as an all-or-nothing quality, is a crucial property in the group communication mechanism. If one or more group members have a problem receiving the message, the process that delivers it to them will get an error notice.

The ordering attribute of the messages is in charge of managing the order in which messages are delivered. Message ordering types include:

- No order means message sending happens without regard for the order to the group.
- FIFO order means messages are shown in the order they are sent.
- Casual order means messages are shipped in a random order after receiving another message.
- Total order means all communications are sent to all group members in the same order.

Group Organization

Group communication systems can be classified as either closed or open. Only members of the closed group can send messages to the group. Users who are not group members can send messages to each member separately. Non-members in the open group can send messages to the group. The program's objective determines the use of a closed or open group.

The group's internal structure can be determined based on its organization. All decisions in egalitarian groupings are made collaboratively. In the event of a failure, the group proceeds without a procedure. The coordinator makes decisions in hierarchical clusters. The loss of the coordinator brings all processes to a standstill.

Q14. Explain group communication in message passing.

Ans :

- A group is a collection of processes that act together in some system or user specified way.
- The key property that all groups have that when a message is sent to the group; itself all the members of the group receive it.
- It is a form of one to many communication.

One – to - many

a) Group management

b) Group Addressing

c) Message delivery to receiving process

d) Buffered and unbuffered multicast

e) Flexible reliability in multicast communication

f) Atomic multicast

g) Group communication primitives

- Single sender and multiple receiver is also known as multicast communication
- Broadcast is a special case of multicast communication.

a) Group Management

- In case of group communication the communicating processes forms a group.
- Such a group may of either of 2 types.
- Closed group is the one, in which only member can send message, outside process cannot sned message to the group as whole but can send to single member of a group.
- Open group is one in which any process in the system can send the message to group as a whole.
- Message passing provides flexibility and create groups dynamically and to allow process join or leave group dynamically.
- Centralized group server is used for this purpose but suffers from poor reliability and scalability.

b) Group Addressing

- Two level naming scheme is used for group addressing.
- Multicast address is a special n/w address, packet is delivered automatically here to all m/cs listening to address.
- It broadcast address the all m/cs has to aheck if packet is intended for it or else simply discard so broadcast is less efficient.
- If network doesn't support any addressing among two then one to one communication is used.
- First two methods send single packet but one to one sends many, creating much traffic.

c) Message Delivery to Receiving Process

- User application uses high level group names in program.
- Centralized group server maintains a mapping of high level group names to their low level names, when sender sends message to the group. With high level name, kernel of server m/c asks the group servers low level name and list of process identifiers.
- When packet reaches m/c kernel, that m/c extract list of process IDs and forwards message to those processes belonging to its own m/c.
- If none of ID's is matching packet is discarded.

d) Buffered and unbuffered multicast

- **Send to all semantics:** A copy of message is sent to each process of multicast group and message is buffered until it is accepted by process.
- **Bulletin board semantics:** Message to be multicast is addressed to channel instead of each process.
  - Channel plays as a bulletin board.

e) Flexible Reliability in Multicast communication

- In multicast mechanism there is flexibility for user definable reliability.
- Degree of reliability is expressed in 4 forms.

  a) O-Reliable: - No response is expected at all.

  b) I - Reliable: - Sender expects response from any one of the receiver.

  c) Mount of n reliable :- Sender expects response from m of n receiver.

  d) All reliable : Sender expects response from all.

f) Atomic Multicast

- All or nothing property all reliable form requires atomic multicast facility.
- Message passing system should support both atomic and non- atomic multicast facility.
- It should provide flexibility to sender to specify whether atomicity is required or not.
- It is difficult to implement atomic multicast if sender or receiver fails.
- Solution to this is fault tolerated atomic multicast protocol.
- In this protocol, each message has message identifier field to distinguish message and one field to indicate data message in atomic multicast message.

g) Group communication primitives

- In one to one and one to many send or has to specify destination address and pointer to data so some send primitives can be used for both.
- But some systems use send group primitives because

  1) It simplifies design and implementation

  2) It provides greater flexibilities.

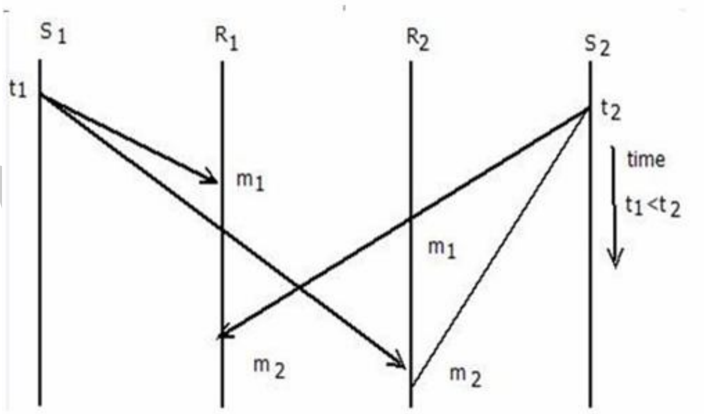
Many – to – one

- Multiple sender single receiver
- Single receiver may be selective or non-selective.
- Selective receiver specifies unique sender message exchange takes place only if the sender sends the message.
- Non-selective receiver specifies set of sender if any from that sends message then message exchange takes place.
- No determinism issue need to be handled here.
- Rest all factors are same as for one – to – many communication.

Many –to – many

- One to many, many to one, all issues are induced in this.
- Ordered message delivery/ event ordering.
  1) Absolute ordering
  2) Consistent ordering
  3) Casual

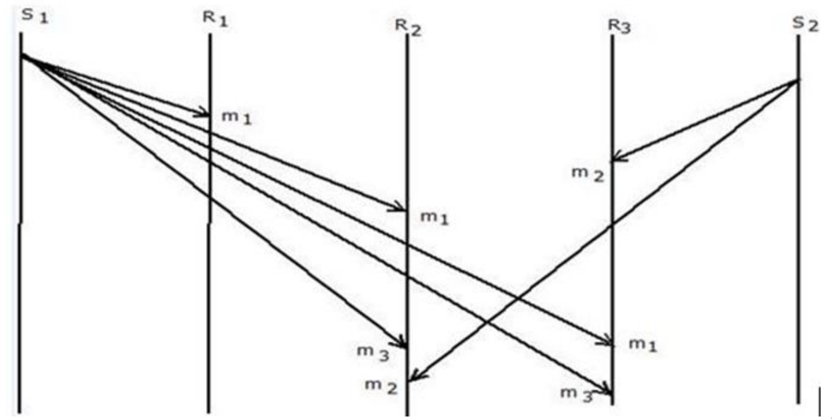
1. Absolute Ordering



- It ensures that all message delivered to all receiver in exact in which they are sent.
- Used to implement is to use global timestamps as message id.
- System is assumed to have clock synchronization and when sender message timestamps is taken as id of message and embedded in message.
- Kernel of receiver m/c, saves all incoming message in separate queue.
- A sliding window is used to periodically deliver message from receiver i.e fixed time interval is selected as window size.
- Window size and properly chosen by considering maximum time for message to kernel from one m/c to other in n/w.

2. Casual Ordering

- For some application weaker semantics acceptable.
- Such semantic is called casual ordering.



- This semantics ensures that is the sending one message is casually related to event of sending another message, two messages are delivered to all receiver in correct order, otherwise these may be delivered in any order.
- Two message sending events are said to be casually related if they are correlated by happened before relation.
- Two message sending events are casually related if there is any possibilities of second one being influenced in a way by first one.
- For 8 implementing casual ordering semantics is the CACAST protocol:

  1. Each member process of group maintaining a vector of n components, where n is the total no. of members in group.

     Each member is assigned a sequence no. from 0 to n, and the ith component of vector corresponds to member with sequence no. i

  2. To send message, a process increments he value of its own component vector and sends vector as a part of message.

  3. When message arrives at receiver size, it is buffered by runtime system.

     a) Runtime system, test two conditions given below to user process or must be delayed to ensure casual ordering.

     b) The two conditions are:

        $S [i] = R [i] + 1$ and $s [i]\ d \leq R [i]$ for all $j \neq i$.

     c) 1st condition ensures that receiver has not missed any message from sender and 2nd condition ensures that sender has not get received any message that receiver has not get received.

     d) This test is needed to make sure that sender's message is not casually related to message missed by receiver.

     e) If message passes these tests runtime system delivers it to user process otherwise message is buffered.

2.4.2 Publish Subscribe Systems

Q15. Explain the concept of Publish-Subscribe?

Ans : **(Imp.)**

The publish/subscribe model, also known as pub/sub model, is a messaging pattern used in cloud computing to facilitate communication between different services, applications, or components in a distributed system. In this model, publishers send messages to a topic or channel, and subscribers receive those messages that are of interest to them from that topic or channel.

The pub/sub model is commonly used in cloud computing architectures because it decouples the sender and receiver, allowing for more flexible and scalable communication. The sender does not need to know anything about the receivers, and the receivers do not need to know anything about the sender, as they only need to subscribe to the relevant topic or channel. This makes it easier to add new components or services to the system without affecting the existing ones.

The pub/sub model can be implemented using various cloud services such as Amazon SNS (Simple Notification Service), Google Cloud Pub/Sub, and Azure Service Bus. These services provide APIs for publishers and subscribers to send and receive messages, and also provide features such as message filtering, message ordering, and message durability to ensure reliable communication. Overall, the pub/sub model is a powerful and flexible way to facilitate communication in cloud computing systems, enabling components to communicate asynchronously and enabling new components to be added to the system with ease.

The architecture of the pub/sub model in cloud computing typically consists of three main components: publishers, subscribers, and a message broker. Here is a high-level overview of the architecture:

1. **Publishers**

   Publishers are the components or applications that generate messages and send them to the message broker. Publishers do not need to know the identity of subscribers or how many subscribers there are; they simply send messages to a specific topic or channel.

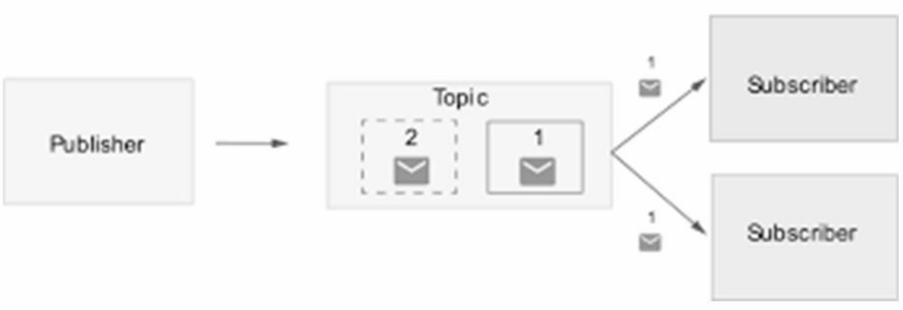
2. **Subscribers**

   Subscribers are the components or applications that receive messages from the message broker. Subscribers subscribe to specific topics or channels, and they receive messages from the message broker only if those messages match their subscription criteria.

3. **Message Broker**

   The message broker is the central component that receives messages from publishers and distributes them to the appropriate subscribers. The message broker acts as an intermediary between publishers and subscribers, and it manages the routing of messages based on topic or channel subscriptions.

PUB/SUB Model



The message broker is the core of the pub/sub model. It maintains a list of active subscribers and their subscriptions, and it forwards messages to the appropriate subscribers based on their subscriptions. The message broker can be implemented as a standalone service or as part of a larger messaging system.

In addition to these three core components, pub/sub architectures may also include additional components such as message filters, message queues, and message processors, depending on the specific requirements of the system.

Examples

Sure, here are some examples of the pub/sub model in cloud computing:

1. **Amazon SNS**

Amazon SNS (Simple Notification Service) is a pub/sub messaging service provided by Amazon Web Services (AWS). It allows publishers to send messages to a topic, and subscribers can receive those messages through various protocols such as email, SMS, HTTP/S, and more. SNS supports fan-out, which means that a single message can be delivered to multiple subscribers in parallel.

2. **Google Cloud Pub/Sub**

Google Cloud Pub/Sub is a messaging service provided by Google Cloud Platform. It allows publishers to send messages to a topic, and subscribers can receive those messages through various protocols such as HTTP/S, gRPC, or by pulling messages from a subscription. Pub/Sub supports scalable and reliable messaging, and can be used for event-driven computing, messaging-oriented middleware, and other use cases.

3. **Azure Service Bus**

Azure Service Bus is a messaging service provided by Microsoft Azure. It allows publishers to send messages to a topic or a queue, and subscribers can receive those messages through various protocols such as AMQP, MQTT, or HTTPS. Service Bus supports advanced messaging features such as message sessions, dead-lettering, and duplicate detection.

4. **Apache Kafka**

Apache Kafka is a distributed streaming platform that can be used for pub/sub messaging. It allows publishers to write messages to a topic, and subscribers can consume those messages from the topic in real-time. Kafka is known for its high-throughput and low-latency messaging, and is widely used for real-time data processing, log aggregation, and other use cases.

These are just a few examples of the pub/sub model in cloud computing. Many other cloud providers and open-source projects offer similar messaging services that use the pub/sub model.

2.4.3 Message Queues

Q16. What is Message Queue? Explain.

Ans :

A message queue is a messaging destination that uses the queue data structure to facilitate asynchronous communication between two services, commonly used in serverless and microservices architectures. Messages are stored on the queue until they are processed and deleted. Each message is processed only once by a single consumer. And MQ helps to decouple heavyweight processing, buffer or batch work, and smooth spiky workloads.

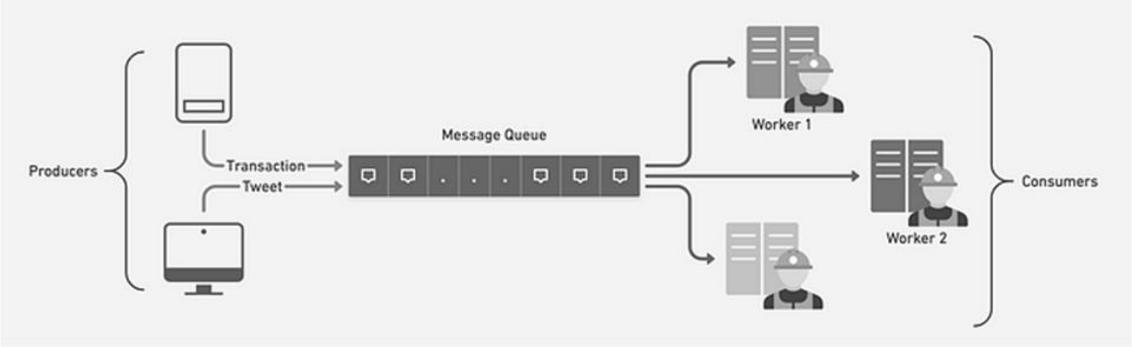
The message queue is comprised of two terms:

1. **Message**

This is the object passed from the producer to the Consumer. The object can be requests, information, meta-data, etc.

2. **Queue**

This is a temporary buffer that stores messages. It uses the First-In-First-Out method to pass the messages from producer to Consumer.



Basic Message Queue

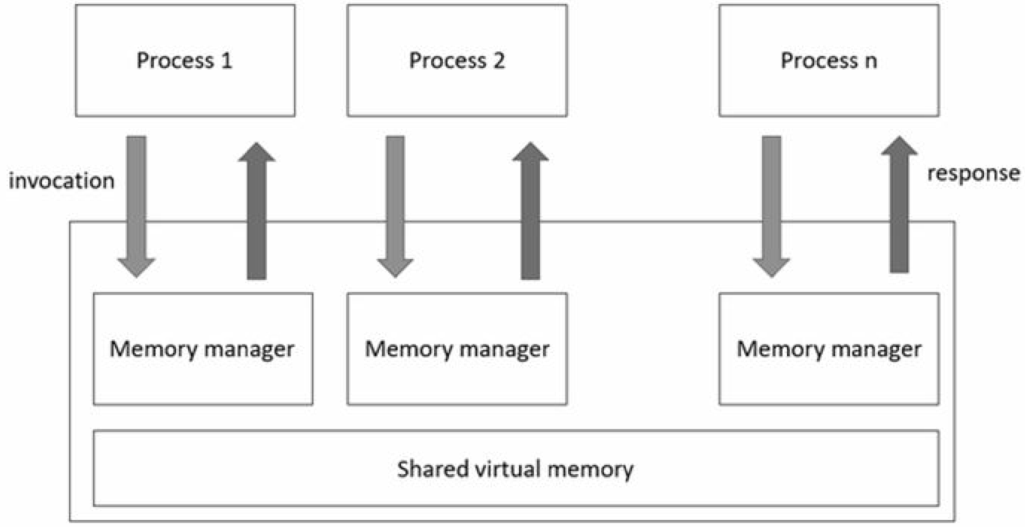
1. The producer creates the message and sends it to the message queue if the Consumer is busy processing it immediately, the queue stores it until the Consumer is available.
2. The Consumer retrieves the message from the queue and starts processing it.
3. The message queue then temporarily locks the message to prevent it from being read by another consumer.
4. After the Consumer completes the message processing, it deletes the message from the queue to prevent it from being read by other consumers.

2.4.4 Shared Memory Approaches

Q17. What is Distributed shared memory and its advantages.

Ans : **(Imp.)**

DSM is a mechanism that manages memory across multiple nodes and makes inter-process communications transparent to end-users. The applications will think that they are running on shared memory. DSM is a mechanism of allowing user processes to access shared data without using inter-process communications. In DSM every node has its own memory and provides memory read and write services and it provides consistency protocols. The distributed shared memory (DSM) implements the shared memory model in distributed systems but it doesn't have physical shared memory. All the nodes share the virtual address space provided by the shared memory model. The Data moves between the main memories of different nodes.



Types of Distributed Shared Memory

On-Chip Memory

- The data is present in the CPU portion of the chip.
- Memory is directly connected to address lines.
- On-Chip Memory DSM is expensive and complex.

Bus-Based Multiprocessors:

- A set of parallel wires called a bus acts as a connection between CPU and memory.
- Accessing of same memory simultaneously by multiple CPUs is prevented by using some algorithms
- Cache memory is used to reduce network traffic.

Ring-Based Multiprocessors

- There is no global centralized memory present in Ring-based DSM.
- All nodes are connected via a token passing ring.
- In ring-bases DSM a single address line is divided into the shared area.

Advantages

- **Simpler Abstraction**

  Programmer need not concern about data movement, As the address space is the same it is easier to implement than RPC.

- **Easier Portability**

  The access protocols used in DSM allow for a natural transition from sequential to distributed systems. DSM programs are portable as they use a common programming interface.

- **Locality of Data**

  Data moved in large blocks i.e. data near to the current memory location that is being fetched, may be needed future so it will be also fetched.

- **On-demand data Movement**

  It provided by DSM will eliminate the data exchange phase.

- **Larger Memory Space**

  It provides large virtual memory space, the total memory size is the sum of the memory size of all the nodes, paging activities are reduced.

- **Better Performance**

  DSM improve performance and efficiency by speeding up access to data.

- **Flexible Communication Environment**

  They can join and leave DSM system without affecting the others as there is no need for sender and receiver to existing,

- **Process Migration Simplified**

  They all share the address space so one process can easily be moved to a different machine.

Apart from the above-mentioned advantages, DSM has furthermore advantages like:

- Less expensive when compared to using a multiprocessor system.
- No bottlenecks in data access.
- Scalability i.e. Scales are pretty good with a large number of nodes.

2.5 Distributed Objects

Q18. Explain about distributed objects.

Ans : **(Imp.)**

A distributed object is an object that can be accessed remotely.

This means that a distributed object can be used like a regular object, but from anywhere on the network. Distributed objects might be used :

1. To share information across applications or users.
2. To synchronize activity across several machines.
3. To increase performance associated with a particular task.
4. Work together by sharing data and invoking methods.

5. This often involves location transparency, where remote objects appear the same as local objects.

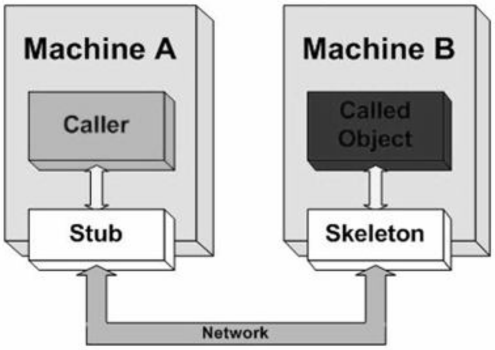
6. The main method of distributed object communication is with remote method invocation, generally by message-passing: one object sends a message to another object in a remote machine or process to perform some task. The results are sent back to the calling object.

Q19. How Distributed Object Communicate ?

Ans :

The widely used approach on how to implement the communication channel is realized by using stubs and skeletons. They are generated objects whose structure and behavior depends on chosen communication protocol, but in general provide additional functionality that ensures reliable communication over the network.

In RMI, a stub (which is the bit on the client) is defined by the programmer as an interface. The rmic (rmi compiler) uses this to create the class stub. The stub performs type checking. The skeleton is defined in a class which implements the interface stub.



When a caller wants to perform a remote call on the called object, it delegates requests to its stub which initiates communication with the remote skeleton. Consequently, the stub passes caller arguments over the network to the server skeleton. The skeleton then passes received data to the called object, waits for a response and returns the result to the client stub. Note that there is no direct communication between the caller and the called object.

In more details, the communication consists of several steps:

1. caller calls a local procedure implemented by the stub.
2. stub marshalls call type and the input arguments into a request message.
3. client stub sends the message over the network to the server and blocks the current execution thread.
4. server skeleton receives the request message from the network.
5. skeleton unpacks call type from the request message and looks up the procedure on the called object.
6. skeleton unmarshalls procedure arguments
7. skeleton executes the procedure on the called object.
8. called object performs a computation and returns the result.
9. skeleton packs the output arguments into a response message.
10. skeleton sends the message over the network back to the client.
11. client stub receives the response message from the network.
12. stub unpacks output arguments from the message.
13. stub passes output arguments to the caller, releases execution thread and caller then continues in execution.

The advantage of this architecture is that neither the caller nor the called object has to implement network related logic. This functionality, that ensures reliable communication channel over the network, has been moved to the stub and the skeleton layer.

Stub

The client side object participating in distributed object communication is known as a stub or proxy, and is an example of a proxy object. The stub acts as a gateway for client side objects and all outgoing requests to server side objects that are routed through it. The stub wraps client object functionality and by adding the network logic ensures the reliable communication channel between client and server. The stub can be written up manually or generated automatically depending on chosen communication protocol.

The stub is responsible for:

- initiating the communication towards the server skeleton.
- translating calls from the caller object.
- marshalling of the parameters.
- informing the skeleton that the call should be invoked.
- passing arguments to the skeleton over the network.
- unmarshalling of the response from the skeleton.
- informing the caller that the call is complete.

Skeleton

The server side object participating in distributed object communication is known as a skeleton .

A skeleton acts as gateway for server side objects and all incoming clients requests are routed through it. The skeleton wraps server object functionality and exposes it to the clients, moreover by adding the network logic ensures the reliable communication channel between clients and server. Skeletons can be written up manually or generated automatically depending on chosen communication protocol. The skeleton is responsible for:

- translating incoming data from the stub to the correct up-calls to server objects
- unmarshalling of the arguments from received data
- passing arguments to server objects
- marshalling of the returned values from server objects
- passing values back to the client stub over the network

2.6 Case Study: Enterprise Java Beans

Q20. Write a case study on Java Beans.

Ans : **(Imp.)**

Enterprise Java Beans (EJB) is a technology used for building distributed business applications in the Java platform. EJB is a component model that provides a set of services for developing, deploying, and managing distributed applications. In this case study, we will look at how EJB can be used to build a simple banking application.

The banking application will consist of the following components:

1. A web-based front-end for the customers to access their accounts and perform transactions.
2. A back-end server that manages the customer accounts and transactions.
3. A database to store the customer data and transaction history.

Here is an overview of how EJB can be used to implement each component:

1. Web-based front-end

The front-end can be implemented using JavaServer Faces (JSF), which is a web application framework that integrates with EJB. The front-end can access the back-end services through EJB remote interfaces, which are Java interfaces that define the methods and data types used by the back-end components. The front-end can use Java Naming and Directory Interface (JNDI) to look up the remote interfaces of the back-end components and establish a connection.

2. Back-end Server

The back-end server can be implemented using EJB session beans, which are components that provide a set of business logic services. The session beans can be stateful or stateless, depending on the type of service provided. For example, a session bean can provide services such as account creation, balance inquiry, funds transfer, and transaction history retrieval. The session beans can interact with the database through Java Persistence API (JPA), which is a set of APIs that provide an object-relational mapping (ORM) framework.

3. Database

The database can be implemented using a relational database management system (RDBMS) such as MySQL or Oracle. The database can store the customer data and transaction history in tables that correspond to the entities defined in the JPA entity classes. The session beans can access the database through JPA EntityManager, which is an interface that provides a set of methods for performing CRUD (create, read, update, delete) operations on the entities.

The EJB container provides a set of services for deploying, configuring, and managing the EJB components. The EJB container can manage the life cycle of the EJB components, including instantiation, pooling, activation, passivation, and removal. The EJB container also provides security, transaction management, and resource management services, such as connection pooling and data source configuration.

In summary, EJB can be used to implement a distributed banking application by providing a component model that separates the presentation, business logic, and data persistence layers. The use of EJB can simplify the development, deployment, and management of the application, while providing a scalable and robust architecture.

2.6.1 From Objects To Components

Q21. Explain, the evolution of components from objects.

Ans : **(Imp.)**

The evolution from objects to components reflects a shift in software development paradigms, from a focus on code reusability to a focus on modular and reusable software systems. While objects and components share some similarities, such as encapsulation, abstraction, and inheritance, there are some key differences between the two concepts.

Objects are instances of classes that encapsulate data and behavior, and they can communicate with each other through method calls. Objects are typically designed to solve a specific problem or task, and they can be reused by instantiating them in different contexts.

Components, on the other hand, are self-contained units of software that provide a specific functionality or service. Components are designed to be modular and composable, and they can be reused in different applications or systems. Components can have well-defined interfaces, which specify the inputs, outputs, and behavior of the component, and they can be connected to other components to form a larger system.

The shift from objects to components reflects a change in software development practices, from monolithic and tightly-coupled systems to distributed and loosely-coupled systems. Components enable software developers to create more flexible and adaptable systems by separating concerns and providing well-defined interfaces. Components can be developed independently of each other, and they can be combined to form new applications or systems by connecting them through well-defined interfaces.

Some of the benefits of component-based software development include:

1. **Reusability**

   Components can be reused across different applications or systems, which can save development time and reduce costs.

2. **Interoperability**

   Components can be designed to work with other components, regardless of the programming language or platform used to develop them.

3. **Modularity**

   Components can be designed to be modular, which makes them easier to test, maintain, and evolve.

4. **Scalability**

   Components can be deployed on distributed systems, which enables them to scale horizontally by adding more instances of the component.

5. **Robustness**

   Components can be designed to be resilient to failures and errors, which can improve the reliability and availability of the system.

In summary, the shift from objects to components reflects a change in software development practices, from a focus on code reusability to a focus on modular and reusable software systems. Components enable software developers to create more flexible, adaptable, and scalable systems by separating concerns and providing well-defined interfaces.

Short Question and Answers

1. What is an overlay network.

Ans :

An overlay network is a logical network that is built on top of an existing physical network infrastructure. It is used to provide additional functionality or services beyond what is available in the underlying network, such as increased security, better performance, or enhanced routing capabilities.

In an overlay network, each node in the logical network is associated with one or more nodes in the physical network. The overlay nodes communicate with each other using the physical network as a transport mechanism, but they also use additional protocols and mechanisms to implement the overlay network functionality.

2. Applications of overlay networks include:

Ans :

i) **Virtualization:** Overlay networks are used in virtualization environments to provide secure and isolated communication between virtual machines or containers.

ii) **Content Delivery Networks (CDNs):** CDNs use overlay networks to distribute content across a large-scale network of servers to improve performance and availability.

iii) **Peer-to-Peer (P2P) Networking:** P2P networks use overlay networks to enable direct communication and file sharing between nodes without the need for a centralized server.

iv) **Cloud Computing:** Overlay networks are used in cloud computing environments to provide network isolation and security between different tenants or users.

3. MPI

Ans :

Message Passing Interface (MPI) is a standardized and portable message-passing system developed for distributed and parallel computing. MPI provides parallel hardware vendors with a clearly defined base set of routines that can be efficiently implemented. As a result, hardware vendors can build upon this collection of standard low-level routines to create higher-level routines for the distributed-memory communication environment supplied with their parallel machines.

4. Remote procedure call.

Ans :

Remote Procedure Call (RPC) is a communication technology that is used by one program to make a request to another program for utilizing its service on a network without even knowing the network's details. A function call or a subroutine call are other terms for a procedure call.

It is based on the client-server concept. The client is the program that makes the request, and the server is the program that gives the service. An RPC, like a local procedure call, is based on the synchronous operation that requires the requesting application to be stopped until the remote process returns its results. Multiple RPCs can be executed concurrently by utilizing lightweight processes or threads that share the same address space. Remote Procedure Call program as often as possible utilizes the Interface Definition Language (IDL), a determination language for describing a computer program component's Application Programming Interface (API). In this circumstance, IDL acts as an interface between machines at either end of the connection, which may be running different operating systems and programming languages.

5. Callback RPC

Ans :

In a Callback RPC, a P2P (Peer-to-Peer)paradigm opts between participating processes. In this way, a process provides both client and server functions which are quite helpful. Callback RPC's features include:

- The problems encountered with interactive applications that are handled remotely
- It provides a server for clients to use.
- Due to the callback mechanism, the client process is delayed.
- Deadlocks need to be managed in callbacks.
- It promotes a Peer-to-Peer (P2P) paradigm among the processes involved.

6. Batch-mode RPC

Ans :

Batch-mode RPC enables the client to line and separate RPC inquiries in a transmission buffer before sending them to the server in a single batch over the network. Batch-mode RPC's features include:

- It diminishes the overhead of requesting the server by sending them all at once using the network.
- It is used for applications that require low call rates.
- It necessitates the use of a reliable transmission protocol.

7. Disadvantages of Remote Procedure Calls

Ans :

- In Remote Procedure Calls parameters are only passed by values as pointer values are not allowed.
- It involves a communication system with another machine and another process, so this mechanism is extremely prone to failure.
- The RPC concept can be implemented in a variety of ways, hence there is no standard.
- Due to the interaction-based nature, there is no flexibility for hardware architecture in RPC.
- Due to a remote procedure call, the process's cost has increased.

8. Components of RMI Architecture

Ans :

(i) Stub

- The stub is an object, acts as a gateway for the client-side.
- All the outgoing requests are routed through it.
- It resides at the client-side and represents the remote object.
- When the caller invokes a method on the stub object, it does the following tasks:
- It initiates a connection with remote Virtual Machine (JVM).
- It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM).
- It waits for the result.
- It reads (unmarshals) the return value or exception, and
- It finally, returns the value to the caller.

(ii) Skeleton

- The skeleton is an object, acts as a gateway for the server-side object.
- All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:
- It reads the parameter for the remote method
- It invokes the method on the actual remote object, and
- It writes and transmits (marshals) the result to the caller.

9. Write a Case Study on JAVA RMI.

Ans :

Java RMI (Remote Method Invocation) is a Java API that provides a mechanism for creating distributed Java applications. It allows Java objects to invoke methods on remote Java objects located on different JVMs (Java Virtual Machines) running on different machines. In this case study, we will look at how Java RMI was used to develop a distributed application.

10. Publish-Subscribe

Ans :

The publish/subscribe model, also known as pub/sub model, is a messaging pattern used in cloud computing to facilitate communication between different services, applications, or components in a distributed system. In this model, publishers send messages to a topic or channel, and subscribers receive those messages that are of interest to them from that topic or channel.

The pub/sub model is commonly used in cloud computing architectures because it decouples the sender and receiver, allowing for more flexible and scalable communication. The sender does not need to know anything about the receivers, and the receivers do not need to know anything about the sender, as they only need to subscribe to the relevant topic or channel. This makes it easier to add new components or services to the system without affecting the existing ones.

Choose the Correct Answers

1. Which of the following is not a benefit of using an overlay network? [d]

   (a) Improved fault tolerance (b) Reduced latency

   (c) Improved scalability (d) Reduced complexity

2. What is MPI in the context of distributed systems? [b]

   (a) A messaging protocol for the internet

   (b) A programming interface for parallel computing

   (c) A communication protocol for wireless networks

   (d) A file transfer protocol for distributed systems

3. Which of the following is not a commonly used function in MPI? [c]

   (a) MPI_Send (b) MPI_Recv

   (c) MPI_Lock (d) MPI_Barrier

4. Which of the following is not a component of RPC? [d]

   (a) Stub (b) Skeleton

   (c) Transport protocol (d) Naming service

5. What is the purpose of an ID field in a request message? [a]

   (a) To identify the client that sent the message

   (b) To identify the server that should process the message

   (c) To provide information about the type of message being sent

   (d) To ensure that messages are delivered in order

6. Which of the following is not a commonly used request-reply protocol? [b]

   (a) HTTP (b) SMTP

   (c) DNS (d) Remote Procedure Call (RP(C)

7. RPC provides a(an) ________ on the client-side, a separate one for each remote procedure. [a]

   (a) Stub (b) Identifier

   (c) Name (d) Process identifier

8. The remote method invocation ________ [b]

   (a) Allows a process to invoke memory on a remote object

   (b) Allows a thread to invoke a method on a remote object

   (c) Allows a thread to invoke memory on a remote object

   (d) Allows a process to invoke a method on a remote object

9. Which of the following protocols is commonly used for group communication in distributed systems? [b]

   (a) TCP (b) UDP

   (c) HTTP (d) FTP

10. Which of the following is a type of group communication? [d]

    (a) Unicast (b) Broadcast

    (c) Multicast (d) All of the above

Fill in the Blanks

1. An ___________ is a logical network that is built on top of an existing physical network infrastructure.
2. ___________ is a standardized and portable message-passing system developed for distributed and parallel computing.
3. ___________ is a method used to enable communication between different processes or systems over a network
4. ___________ are objects that are distributed across different address spaces, either in multiple computers connected via a network or even indifferent processes on the same computer, but which work together by sharing data and invoking methods
5. A protocol for nodes to communicate by sending messages to each other is called ___________
6. A process that is based on IPC mechanism which executes on different systems and can communicate with other processes using message based communication, is called ___________
7. The ___________is an object, acts as a gateway for the server-side object.
8. ___________ communication occurs when the host simultaneously attempts to communicate with all the processes in a distributed system
9. A ___________ is a messaging destination that uses the queue data structure to facilitate asynchronous communication between two services
10. ___________ is a mechanism that manages memory across multiple nodes and makes inter-process communications transparent to end-users

Answers

1. Overlay network
2. Message Passing Interface (MPI)
3. Remote method Invokation
4. Distributed objects
5. Request reply protocol
6. RPC
7. Skeleton
8. Broadcast
9. Message queue
10. Distributed shared memory

UNIT III

Introduction to Cloud Computing: Cloud Computing in a Nutshell, System Models for Distributed and Cloud Computing, Roots of Cloud Computing, Grid and Cloud, Layers and Types of Clouds, Desired Features of a Cloud, Basic Principles of Cloud Computing, Challenges and Risks, Service Models.

3.1 Introduction to Cloud Computing

3.1.1 Cloud Computing in a Nutshell

Q1. Define Cloud Computing? Explain the benefits of cloud computing.

Ans : **(Imp.)**

Meaning

Cloud computing is a general term for the delivery of hosted services over the internet.

Cloud computing enables companies to consume a compute resource, such as a virtual machine (VM), storage or an application, as a utility — just like electricity - rather than having to build and maintain computing infrastructures in house.

Benefits

Cloud computing is a big shift from the traditional way businesses think about IT resources. There are 6 common reasons organisations are turning to cloud computing services:

1. **Cost**

   Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site datacenters—the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure. It adds up fast.

2. **Speed**

   Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

3. **Global Scale**

   The benefits of cloud computing services include the ability to scale elastically. In cloud speak, that means delivering the right amount of IT resources for example, more or less computing power, storage, bandwidth right when its needed and from the right geographic location.

4. **Productivity**

   On-site data centers typically require a lot of "racking and stacking" hardware set up, software patching and other time-consuming IT management chores. Cloud computing removes the need for many of these tasks, so IT teams can spend time on achieving more important business goals.

5. **Performance**

   The biggest cloud computing services run on a worldwide network of secure data centers, which are regularly upgraded to the latest generation of fast and efficient computing hardware. This offers several benefits over a single corporate data center, including reduced network latency for applications and greater economies of scale.

6. **Reliability**

   Cloud computing makes data backup, disaster recovery and business continuity easier and less expensive, because data can be mirrored at multiple redundant sites on the cloud provider's network.

Cons

- Higher ongoing operating costs. Could cloud systems work out more expensive?
- Greater dependency on service providers. Can you get problems resolved quickly, even with SLAs?
- Risk of being locked into proprietary or vendor-recommended systems? How easily can you migrate to another system or service provider if you need to?
- What happens if your supplier suddenly decides to stop supporting a product or system you've come to depend on?
- Potential privacy and security risks of putting valuable data on someone else's system in an unknown location?
- If lots of people migrate to the cloud, where they're no longer free to develop neat and whizzy new things, what does that imply for the future development of the Internet?
- Dependency on a reliable Internet connection.

Q2. Explain about the layers of computing.

Ans :

Three Layers of Computing

Different categories of computing facilities. These computing facilities can be segmented into three categories:

- Infrastructure
- Platform
- Application

These three categories of computing facilities form three layers in the basic architecture of computing. It represents the relationships between these three entities.

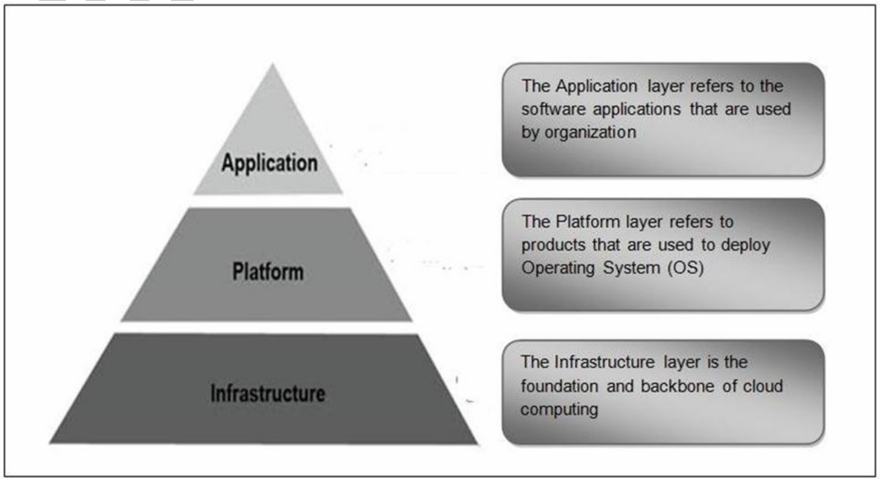

Application
Platform
Infrastructure
The Application layer refers to the software applications that are used by organization
The Platform layer refers to products that are used to deploy Operating System (OS)
The Infrastructure layer is the foundation and backbone of cloud computing

Fig 1.1: Relationship between entities

1. Infrastructure

The bottom layer or the foundation is the 'computing infrastructure' facility. This includes all physical computing devices or hardware components like the processor, memory, network, storage devices and other hardware appliances. Infrastructure refers to computing resources in their bare-metal form. This layer needs basic amenities like electric supply, cooling system etc.

2. Platform

In computing, platform consists of the physical computing device (hardware) loaded with layer(s) of software where the program or application can run. The term 'computing platform' refers to different abstract levels. It consists of:

- Certain hardware components, only.
- Hardware loaded with an operating system (OS).
- Hardware and OS, additionally, loaded with run-time libraries.

Hardware alone can be considered as the platform in case of embedded systems, where physical resources can be accessed without any operating system.

A fully configured physical computer loaded with an operating system is considered as a platform for computing. Different platforms can be installed over the same computing infrastructure. Linux or Windows operating systems installed over the same physical computer (computing infrastructure) can provide two different computing platforms.

The platform layer is also the place where software developers work. Hence Integrated Development Environments (IDEs) and runtimes are part of this layer. Java Development Kit (JDK) or .NET are examples of popular computing frameworks. Software applications can be developed and run over these platforms.

3. Application

Applications (application software) constitute the topmost layer of this layered architecture. This layer generally provides interfaces for interaction with external systems (human or machine) and is accessed by *end users* of computing. A user actually works on the application layer to edit a document, play a game or use the calculator in a computer. At this layer, organizations access enterprise applications using application interfaces to run their business.

Different types of people work at different layers of computing. They need to have different skill-sets and knowledge.

Q3. Write about the challenges of cloud computing.

Ans : **(Imp.)**

1. Data Security and Privacy

Data security is a major concern when switching to cloud computing. User or organizational data stored in the cloud is critical and private. Even if the cloud service provider assures data integrity, it is your responsibility to carry out user authentication and authorization, identity management, data encryption, and access control. Security issues on the cloud include identity theft, data breaches, malware infections, and a lot more which eventually decrease the trust amongst the users of your applications.

2. Cost Management

When there is under optimization of the resources, let's say that the servers are not being used to their full potential, add up to the hidden costs. If there is a degraded application performance or sudden spikes or overages in the usage, it adds up to the overall cost. Unused resources are one of the other main reasons why the costs go up. If you turn on the services or an instance of cloud and forget to turn it off during the weekend or when there is no current use of it, it will increase the cost without even using the resources.

3. Multi-Cloud Environments

Due to an increase in the options available to the companies, enterprises not only use a single cloud but depend on multiple cloud service providers. Most of these companies use hybrid cloud tactics and close to 84% are dependent on multiple clouds. This often ends up being hindered and difficult to manage for the infrastructure team. The process most of the time ends up being highly complex for the IT team due to the differences between multiple cloud providers.

4. **Performance Challenges**

Performance is an important factor while considering cloud-based solutions. If the performance of the cloud is not satisfactory, it can drive away users and decrease profits. Even a little latency while loading an app or a web page can result in a huge drop in the percentage of users. This latency can be a product of inefficient load balancing, which means that the server cannot efficiently split the incoming traffic so as to provide the best user experience. Challenges also arise in the case of fault tolerance, which means the operations continue as required even when one or more of the components fail.

5. **Interoperability and Flexibility**

When an organization uses a specific cloud service provider and wants to switch to another cloud-based solution, it often turns up to be a tedious procedure since applications written for one cloud with the application stack are required to be re-written for the other cloud. There is a lack of flexibility from switching from one cloud to another due to the complexities involved. Handling data movement, setting up the security from scratch and network also add up to the issues encountered when changing cloud solutions, thereby reducing flexibility.

6. **High Dependence on Network**

Since cloud computing deals with provisioning resources in real-time, it deals with enormous amounts of data transfer to and from the servers. This is only made possible due to the availability of the high-speed network. Although these data and resources are exchanged over the network, this can prove to be highly vulnerable in case of limited bandwidth or cases when there is a sudden outage. It is therefore a major challenge for smaller enterprises that have to maintain network bandwidth that comes with a high cost.

7. **Lack of Knowledge and Expertise**

Due to the complex nature and the high demand for research working with the cloud often ends up being a highly tedious task. It requires immense knowledge and wide expertise on the subject. Although there are a lot of professionals in the field they need to constantly update themselves. Cloud computing is a highly paid job due to the extensive gap between demand and supply. Therefore, there is a need for up skilling so these professionals can actively understand, manage and develop cloud-based applications with minimum issues and maximum reliability.

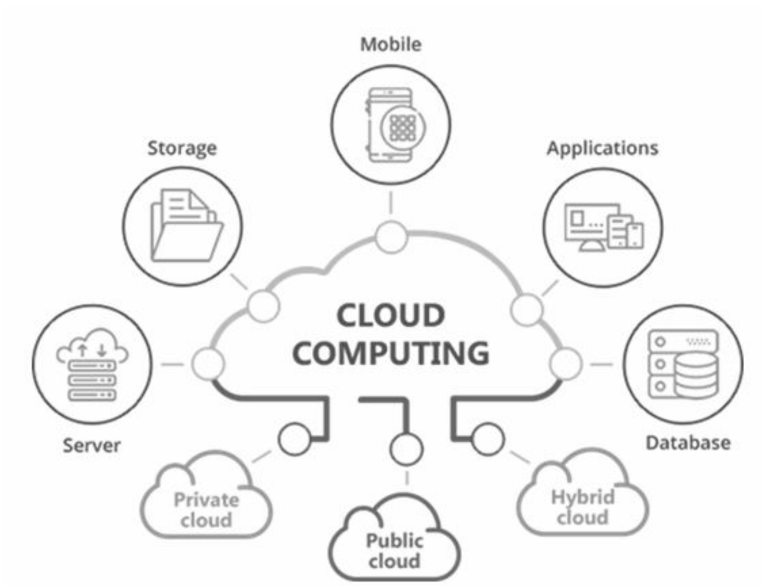
3.2 System Models for Distributed and Cloud Computing

Q4. Explain about system models for distributed and cloud computing.

Ans : **(Imp.)**

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. Massive systems are considered highly scalable, and can reach web-scale connectivity, either physically or logically.



Massive systems are classified into four groups:

1. Clusters

A distributed systems cluster is a group of machines that are virtually or geographically separated and that work together to provide the same service or application to clients. It is possible that many of the services you run in your network today are part of a distributed systems Cluster Distributed Services:

- Domain Naming System
- Windows Internet Naming Service
- Active Directory

2. P2P Networks

In a P2P system, every node acts as both a client and a server, providing part of the system resources. Peer machines are simply client computers connected to the Internet. All client machines act autonomously to join or leave the system freely. This implies that no master-slave relationship exists among the peers. No central coordination or central database is needed. The system is self-organizing with distributed control.

3. Computing Grids

This is the use of widely distributed computer resources to reach a common goal. A computing grid can be thought of as a distributed system with non-interactive workloads that involve many files. Grid computing is distinguished from conventional high-performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application. Grid computers also tend to be more heterogeneous and geographically dispersed than cluster computers.

4. Internet Clouds

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 Virtualized resources from data centers to form an Internet cloud, provisioned with hardware, software, storage, network, and services for paid users to run their applications.

5. **Physical Network**

- The participating peers form the physical network at any time.
- Unlike the cluster or grid, a P2P network does not use a dedicated interconnection network.
- The physical network is simply an ad hoc network formed at various Internet domains randomly using the TCP/IP and NAI protocols

6. **Overlay Network**

- Based on communication or file-sharing needs, the peer IDs form an overlay network at the logical level.
- This overlay is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping.
- When a new peer joins the system, its peer ID is added as a node in the overlay network and is removed from the overlay network automatically when it leaves.
- Therefore, it is the P2P overlay network that characterizes the logical connectivity among the peers.
- Two types of overlay networks:
  - unstructured and structured
- An unstructured overlay network is characterized by a random graph.
- There is no fixed route to send messages or files among the nodes.
- Often, flooding is applied to send a query to all nodes in an unstructured overlay, thus resulting in heavy network traffic and nondeterministic search results.
- Structured overlay networks follow certain connectivity topology and rules for inserting and removing nodes (peer IDs) from the overlay graph.

3.3 Roots of Cloud Computing

Q5. Explain about roots of cloud computing.

Ans :

Cloud computing has its roots in several technological and business trends that emerged in the late 20th century. Here are some of the key roots of cloud computing:

1. **Distributed Computing**

The idea of using multiple computers to work together as a single system dates back to the 1960s. Distributed computing allowed large-scale problems to be solved more efficiently by breaking them down into smaller tasks that could be distributed across multiple machines.

2. **Virtualization**

In the 1990s, virtualization technology began to emerge, allowing multiple virtual machines to run on a single physical machine. This made it possible to use hardware resources more efficiently, and provided a foundation for the development of cloud computing.

3. **Utility Computing**

In the early 2000s, companies began to offer computing resources on a pay-per-use basis, similar to the way that utilities like electricity or water are charged. This idea of "utility computing" laid the groundwork for the pricing models used in cloud computing today.

4. **Web 2.0**

The emergence of Web 2.0 technologies, such as social media, online collaboration tools, and web-based applications, led to an explosion in demand for computing resources that could scale quickly and easily.

5. **Infrastructure as a Service (IaaS)**

In 2006, Amazon Web Services (AWS) launched its Elastic Compute Cloud (EC2) service, which allowed users to rent virtual servers on an hourly basis. This marked the beginning of the era of

Infrastructure as a Service (IaaS) in cloud computing.

6. **Platform as a Service (PaaS)**

In 2008, Google launched its App Engine platform, which allowed developers to build and deploy web applications without worrying about the underlying infrastructure. This was the beginning of the Platform as a Service (PaaS) model in cloud computing.

7. **Software as a Service (SaaS)**

SaaS has its roots in the application service provider (ASP) model, which dates back to the 1990s. However, the emergence of cloud computing made it easier and more cost-effective for companies to offer software applications as a service over the internet.

These are some of the key roots of cloud computing, which have converged to create the powerful and flexible computing infrastructure that we have today.

3.4 Grid and Cloud Layers

Q6. Write about grid and cloud layers.

Ans :

Both grid computing and cloud computing are distributed computing models that allow users to access computing resources over a network. However, there are some key differences between these models, including the layers involved. Here are the layers typically involved in grid computing and cloud computing:

Grid Computing Layers

1. **Application Layer**

This is the layer where users interact with the grid computing system and submit their computing tasks.

2. **Middleware Layer**

This layer is responsible for managing the resources of the grid computing system, including authentication, data management, scheduling, and job execution.

3. **Resource Layer**

This is the layer where the computing resources are located, such as computers, storage systems, and networks.

Cloud Computing Layers

1. **Application Layer**

This is the layer where users interact with the cloud computing system and access applications and services.

2. **Platform Layer**

This layer provides a platform for developing and deploying applications in the cloud, including databases, operating systems, and development tools.

3. **Infrastructure Layer**

This is the layer where the physical and virtual resources of the cloud computing system are located, including servers, storage, and networks.

In summary, the grid computing layers typically focus on managing and utilizing resources, while the cloud computing layers focus on providing a platform for developing and deploying applications and services. However, there is some overlap between these layers, and different implementations of grid and cloud computing may have different layer structures.

3.5 Types of Clouds

Q7. Explain different types of clouds.

Ans : **(Imp.)**

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

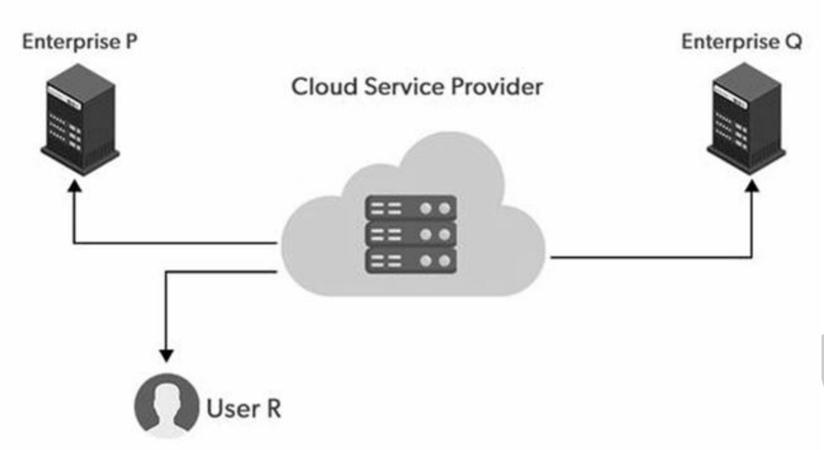
(i) **Public Cloud**

Public clouds are managed by third parties which provide cloud services over the internet to the public, these services are available as pay-as-you-go billing models.

They offer solutions for minimizing IT infrastructure costs and become a good option for handling peak loads on the local infrastructure.

Public clouds are the go-to option for small enterprises, which can start their businesses without large upfront investments by completely relying on public infrastructure for their IT needs.

The fundamental characteristics of public clouds are multitenancy. A public cloud is meant to serve multiple users, not a single customer. A user requires a virtual computing environment that is separated, and most likely isolated, from other users.



Public cloud

Advantages of using a Public cloud are:

1. High Scalability
2. Cost Reduction
3. Reliability and flexibility
4. Disaster Recovery

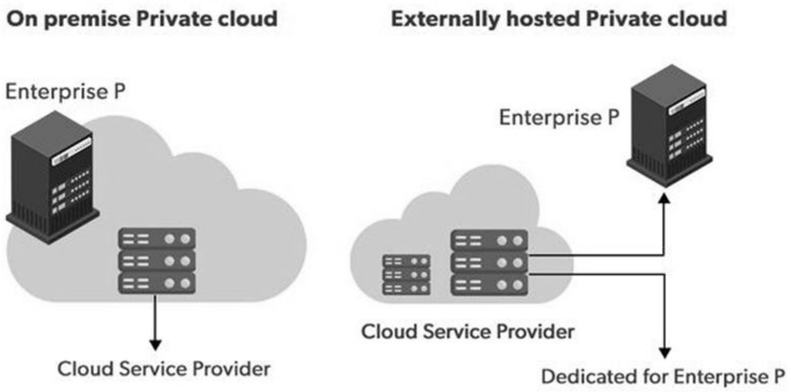
Disadvantages of using a Public cloud are:

1. Loss of control over data
2. Data security and privacy
3. Limited Visibility
4. Unpredictable cost

(ii) Private Cloud

Private clouds are distributed systems that work on private infrastructure and provide the users with dynamic provisioning of computing resources.

Instead of a pay-as-you-go model in private clouds, there could be other schemes that manage the usage of the cloud and proportionally billing of the different departments or sections of an enterprise. Private cloud providers are HP Data Centers, Ubuntu, Elastic-Private cloud, Microsoft, etc.



Private Cloud

The advantages of using a private cloud are as follows:

1. **Customer Information Protection**

   In the private cloud security concerns are less since customer data and other sensitive information do not flow out of private infrastructure.

2. **Infrastructure Ensuring SLAs**

   Private cloud provides specific operations such as appropriate clustering, data replication, system monitoring, and maintenance, disaster recovery, and other uptime services.

3. **Compliance with Standard Procedures and Operations**

   Specific procedures have to be put in place when deploying and executing applications according to third-party compliance standards. This is not possible in the case of the public cloud.

Disadvantages of using a private cloud are:

1. **The restricted area of Operations**

   Private cloud is accessible within a particular area. So the area of accessibility is restricted.

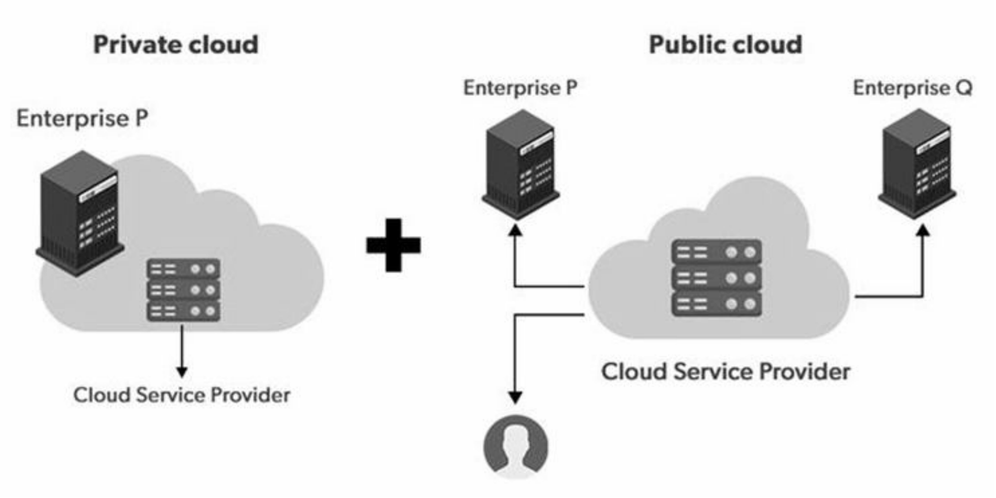
2. **Expertise Requires**

   In the private cloud security concerns are less since customer data and other sensitive information do not flow out of private infrastructure. Hence skilled people are required to manage & operate cloud services.

(iii) Hybrid Cloud

A hybrid cloud is a heterogeneous distributed system formed by combining facilities of the public cloud and private cloud. For this reason, they are also called heterogeneous clouds.

A major drawback of private deployments is the inability to scale on-demand and efficiently address peak loads. Here public clouds are needed. Hence, a hybrid cloud takes advantage of both public and private clouds.



Hybrid Cloud

Advantages of using a Hybrid cloud are:

1. **Cost**

   Available at a cheap cost than other clouds because it is formed by a distributed system.

2. **Speed**

   It is efficiently fast with lower cost, It reduces the latency of the data transfer process.

3. **Security**

   Most important thing is security. A hybrid cloud is totally safe and secure because it works on the distributed system network.

Disadvantages of using a Hybrid cloud are:

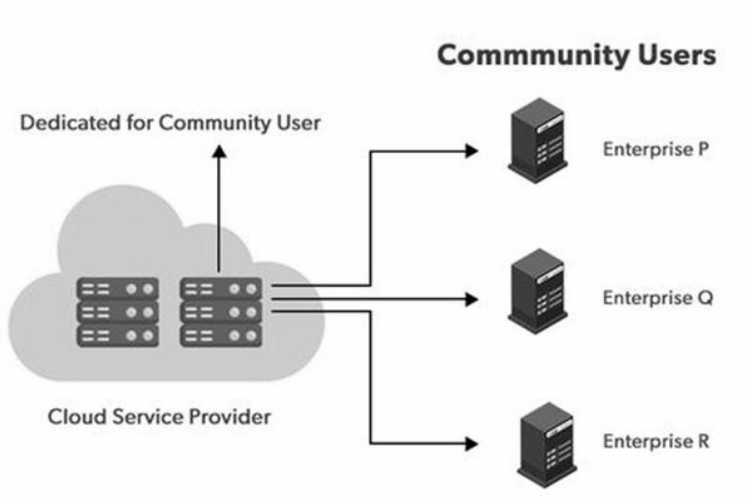
1. It's possible that businesses lack the internal knowledge necessary to create such a hybrid environment. Managing security may also be more challenging. Different access levels and security considerations may apply in each environment.

2. Managing a hybrid cloud may be more difficult. With all of the alternatives and choices available today, not to mention the new PaaS components and technologies that will be released every day going forward, public cloud and migration to public cloud are already complicated enough. It could just feel like a step too far to include hybrid.

(iv) Community Cloud

Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector. But sharing responsibilities among the organizations is difficult.

In the community cloud, the infrastructure is shared between organizations that have shared concerns or tasks. An organization or a third party may manage the cloud.



Community Cloud

Advantages of using Community cloud are:

1. Because the entire cloud is shared by numerous enterprises or a community, community clouds are cost-effective.
2. Because it works with every user, the community cloud is adaptable and scalable. Users can alter the documents according to their needs and requirements.
3. Public cloud is less secure than the community cloud, which is more secure than private cloud.
4. Thanks to community clouds, we may share cloud resources, infrastructure, and other capabilities between different enterprises.

Disadvantages of using Community cloud are:

1. Not all businesses should choose community cloud.
2. gradual adoption of data
3. It's challenging for corporations to share duties.

Sectors that use community clouds are:

1. Media Industry

Media companies are looking for quick, simple, low-cost ways for increasing the efficiency of content generation. Most media productions involve an extended ecosystem of partners. In particular, the creation of digital content is the outcome of a collaborative process that includes the movement of large data, massive compute-intensive rendering tasks, and complex workflow executions.

2. Healthcare Industry

In the healthcare industry community clouds are used to share information and knowledge on the global level with sensitive data in the private infrastructure.

3. Energy and Core Industry

In these sectors, the community cloud is used to cluster a set of solution which collectively addresses the management, deployment, and orchestration of services and operations.

4. **Scientific Research**

   In this organization with common interests in science share a large distributed infrastructure for scientific computing.

(v) Multicloud

Multicloud is the use of multiple cloud computing services from different providers, which allows organizations to use the best-suited services for their specific needs and avoid vendor lock-in.

This allows organizations to take advantage of the different features and capabilities offered by different cloud providers.

Advantages of using Multi-cloud

1. **Flexibility**

   Using multiple cloud providers allows organizations to choose the best-suited services for their specific needs, and avoid vendor lock-in.

2. **Cost-effectiveness**

   Organizations can take advantage of the cost savings and pricing benefits offered by different cloud providers for different services.

3. **Improved Performance**

   By distributing workloads across multiple cloud providers, organizations can improve the performance and availability of their applications and services.

4. **Increased Security**

   Organizations can increase the security of their data and applications by spreading them across multiple cloud providers and implementing different security strategies for each.

Disadvantages of using multi-cloud

1. **Complexity**

   Managing multiple cloud providers and services can be complex and require specialized knowledge and expertise.

2. **Increased Costs**

   The cost of managing multiple cloud providers and services can be higher than using a single provider.

3. **Compatibility Issues**

   Different cloud providers may use different technologies and standards, which can cause compatibility issues and require additional resources to resolve.

4. **Limited Interoperability**

   Different cloud providers may not be able to interoperate seamlessly, which can limit the ability to move data and applications between them.

3.6 Desired Features of A Cloud

Q8. Describe briefly about desired features of a Cloud.

Ans : **(Imp.)**

A cloud computing system is a network of remote servers that provide various computing services, including storage, processing power, and applications, over the internet.

The following are some of the desired features of a cloud:

1. **Scalability**

   One of the most signiûcant beneûts of cloud computing is its ability to scale resources up or down based on demand. A cloud infrastructure should be capable of adding or removing computing resources without any disruption to service.

2. **Availability**

   A cloud system should be highly available, providing consistent access to resources and services. The cloud infrastructure should be designed to minimize downtime, and services should be distributed across multiple servers and data centers to prevent outages.

3. **Security**

Cloud providers must ensure the security and confidentiality of customer data. The cloud infrastructure should implement robust security measures, such as encryption, access control, and regular security audits, to protect against data breaches and other threats.

4. **Reliability**

A cloud system must be reliable, providing consistent performance and availability. The infrastructure should be designed to handle large volumes of traffic and should provide failover capabilities to ensure that services remain available in the event of a system failure.

5. **Flexibility**

A cloud system should be flexible, providing the ability to customize services and resources to meet the needs of individual customers. Cloud providers should offer a range of service plans and options, allowing customers to tailor their services to their specific requirements.

6. **Cost-effectiveness**

A cloud system should be cost-effective, offering affordable and transparent pricing. Customers should be able to choose from a range of pricing options, such as pay-as-you-go or subscription-based pricing, to meet their budgetary needs.

7. **Performance**

A cloud system should be optimized for performance, providing high-speed connectivity, low-latency, and fast data processing capabilities. The cloud infrastructure should be designed to handle a large number of requests and data-intensive workloads efficiently.

8. **Manageability**

A cloud system should be easy to manage, with intuitive interfaces and tools for managing resources, deploying applications, and monitoring performance. The cloud provider should offer robust management tools to enable customers to configure and manage their services easily.

9. **Interoperability**

A cloud system should be designed to support interoperability, enabling customers to integrate their applications and services with other cloud-based and on-premises systems. The cloud infrastructure should support standard APIs and protocols to facilitate integration and data exchange.

10. **Sustainability**

A cloud system should be designed with sustainability in mind, minimizing its environmental impact by reducing energy consumption, carbon emissions, and waste. The cloud provider should adopt eco-friendly practices, such as using renewable energy, optimizing data center cooling, and recycling equipment.

11. **Compliance**

A cloud system should comply with relevant regulations and standards, such as GDPR, HIPAA, PCI-DSS, and ISO. The cloud provider should offer compliance certifications and provide customers with the necessary tools and features to meet regulatory requirements.

3.7 Basic Principles of Cloud Computing

Q9. Explain the basic principles of cloud compuitng.

Ans : **(Imp.)**

Basic Principle of Cloud Computing

➢ **Federation**

A cloud computing environment must be capable of providing federated service providers which means that, these providers, must be capable of collaborating and resource sharing at any point irrespective of their type. This is usually needed when an organization extends its computing paradigm from the private to the public cloud. Moreover, This federation must be kept transparent so that the virtual application can be used on all the sites. This makes the application be handled remotely and allows it to migrate from one site to another. Apart from this, the federation must be carried out in a secure and independent way.

- **Independence**

  The user of cloud computing services must be independent of the provider's specific tool and the type of service. According to this principle, a user must be allowed the required virtual resource irrespective of the type of provider. Moreover, it is the responsibility of service providers to handle infrastructure while hiding confidential information.

- **Isolation**

  According to this principle, a service provider must ensure the user with respect to the isolation of their data from others. Even the data in the same cloud must be separated from different users and therefore should not be accessed.

- **Elasticity**

  The user of cloud computing must be provided with ease of accessing and releasing the resources as required. This is typically referred to as elasticity. The rules associated with elasticity must be included within the contract made between consumers and services providers.

- **Business Orientation**

  To develop a more efficient computing environment, an efficient platform must be developed before the applications are included in the cloud. This typically ensures the quality of services and assist SLA (Service-Level-Agreement).

- **Trust**

  To build a successful cloud computing environment, one of the major factors is trust between consumers and service providers. Therefore, effective mechanisms must be included to develop a trustworthy computing environment.

3.8 Challenges and Risks

Q10. Explain the challenges and risks of cloud computing.

Ans : **(Imp.)**

Cloud computing offers numerous benefits, such as scalability, cost savings, and flexibility. However, there are also several challenges and risks associated with cloud computing that organizations should be aware of. Here are some of the major challenges and risks:

1. Security

Security is a top concern for cloud computing. Organizations need to ensure that their data is protected from unauthorized access, data breaches, and other security threats. This requires implementing appropriate security controls, such as encryption, access controls, and monitoring.

2. Compliance

Organizations need to comply with various regulations and standards, such as GDPR, HIPAA, and PCI DSS, when storing and processing data in the cloud. Compliance requirements can vary by region and industry, and failure to comply can result in legal and financial penalties.

3. Data Privacy

Cloud computing can raise concerns about data privacy, especially if the data is stored or processed in a different country with different privacy laws. Organizations need to ensure that they have appropriate data privacy policies and controls in place.

4. Dependence on Service Providers

Organizations that rely on cloud computing are dependent on their service providers to ensure that their services are available, reliable, and secure. This means that organizations need to carefully select their service providers and have contingency plans in place in case of service disruptions or failures.

5. **Integration**

Integrating cloud computing services with existing systems and applications can be challenging, especially if the services are provided by different vendors. This requires careful planning and coordination to ensure that the integration is successful and does not introduce new security or performance issues.

6. **Data Loss**

Cloud computing services can experience data loss due to various reasons such as hardware failure, cyber attacks, and natural disasters. To mitigate this risk, organizations need to have backup and disaster recovery plans in place.

In summary, cloud computing offers many benefits, but it also poses several challenges and risks that organizations need to address to ensure that their data and systems are secure and compliant.

3.9 Service Models

Q11. Explain about different cloud service models.

Ans : **(Imp.)**

Cloud Service Models

There are the following three types of cloud service models:

1. **Infrastructure as a Service (IaaS)**

IaaS is also known as Hardware as a Service (HaaS). It is a computing infrastructure managed over the internet. The main advantage of using IaaS is that it helps users to avoid the cost and complexity of purchasing and managing the physical servers.

Characteristics of IaaS

There are the following characteristics of IaaS -

- Resources are available as a service
- Services are highly scalable.
- Dynamic and flexible
- GUI and API-based access
- Automated administrative tasks

Example:

DigitalOcean, Linode, Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE), Rackspace, and Cisco Metacloud.

2. **Platform as a Service (PaaS)**

PaaS cloud computing platform is created for the programmer to develop, test, run, and manage the applications.

Characteristics of PaaS

There are the following characteristics of PaaS

- Accessible to various users via the same development application.
- Integrates with web services and databases.
- Builds on virtualization technology, so resources can easily be scaled up or down as per the organization's need.
- Support multiple languages and frameworks.
- Provides an ability to "Auto-scale".

Example:

AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, Magento Commerce Cloud, and OpenShift.

3. Software as a Service (SaaS)

SaaS is also known as "on-demand software". It is a software in which the applications are hosted by a cloud service provider. Users can access these applications with the help of internet connection and web browser.

Characteristics of SaaS

There are the following characteristics of SaaS

- Managed from a central location
- Hosted on a remote server
- Accessible over the internet
- Users are not responsible for hardware and software updates. Updates are applied automatically.
- The services are purchased on the pay-as-per-use basis.

Example:

BigCommerce, Google Apps, Salesforce, Dropbox, ZenDesk, Cisco WebEx, ZenDesk, Slack, and GoToMeeting.

Short Question and Answers

1. Define Cloud Computing

Ans :

Meaning

Cloud computing is a general term for the delivery of hosted services over the internet.

Cloud computing enables companies to consume a compute resource, such as a virtual machine (VM), storage or an application, as a utility — just like electricity - rather than having to build and maintain computing infrastructures in house.

2. Benefits of cloud computing

Ans :

1. Cost

Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site datacenters—the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure. It adds up fast.

2. Speed

Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

3. Global Scale

The benefits of cloud computing services include the ability to scale elastically. In cloud speak, that means delivering the right amount of IT resources for example, more or less computing power, storage, bandwidth right when its needed and from the right geographic location.

4. Productivity

On-site data centers typically require a lot of “racking and stacking” hardware set up, software patching and other time-consuming IT management chores. Cloud computing removes the need for many of these tasks, so IT teams can spend time on achieving more important business goals.

5. Performance

The biggest cloud computing services run on a worldwide network of secure data centers, which are regularly upgraded to the latest generation of fast and efficient computing hardware. This offers several benefits over a single corporate data center, including reduced network latency for applications and greater economies of scale.

3. Roots of cloud computing

Ans :

1. Distributed Computing

The idea of using multiple computers to work together as a single system dates back to the 1960s. Distributed computing allowed large-scale problems to be solved more efficiently by breaking them down into smaller tasks that could be distributed across multiple machines.

2. Virtualization

In the 1990s, virtualization technology began to emerge, allowing multiple virtual machines to run on a single physical machine. This made it possible to use hardware resources more efficiently, and provided a foundation for the development of cloud computing.

3. Utility Computing

In the early 2000s, companies began to offer computing resources on a pay-per-use basis, similar to the way that utilities like electricity or water are charged. This idea of “utility computing” laid the groundwork for the pricing models used in cloud computing today.

4. Web 2.0

The emergence of Web 2.0 technologies, such as social media, online collaboration tools, and web-based applications, led to an explosion in demand for computing resources that could scale quickly and easily.

4. Grid Computing Layers.

Ans :

1. Application Layer

This is the layer where users interact with the grid computing system and submit their computing tasks.

2. Middleware Layer

This layer is responsible for managing the resources of the grid computing system, including authentication, data management, scheduling, and job execution.

3. Resource Layer

This is the layer where the computing resources are located, such as computers, storage systems, and networks.

5. Cloud Computing Layers

Ans :

1. Application Layer

This is the layer where users interact with the cloud computing system and access applications and services.

2. Platform Layer

This layer provides a platform for developing and deploying applications in the cloud, including databases, operating systems, and development tools.

3. Infrastructure Layer

This is the layer where the physical and virtual resources of the cloud computing system are located, including servers, storage, and networks.

6. Features of a Cloud.

Ans :

The following are some of the desired features of a cloud:

1. Scalability

One of the most significant benefits of cloud computing is its ability to scale resources up or down based on demand. A cloud infrastructure should be capable of adding or removing computing resources without any disruption to service.

2. Availability

A cloud system should be highly available, providing consistent access to resources and services. The cloud infrastructure should be designed to minimize downtime, and services should be distributed across multiple servers and data centers to prevent outages.

3. Security

Cloud providers must ensure the security and confidentiality of customer data. The cloud infrastructure should implement robust security measures, such as encryption, access control, and regular security audits, to protect against data breaches and other threats.

4. Reliability

A cloud system must be reliable, providing consistent performance and availability. The infrastructure should be designed to handle large volumes of traffic and should provide failover capabilities to ensure that services remain available in the event of a system failure.

5. Flexibility

A cloud system should be ûexible, providing the ability to customize services and resources to meet the needs of individual customers. Cloud providers should offer a range of service plans and options, allowing customers to tailor their services to their speciûc requirements.

7. Principle of Cloud Computing

Ans :

- **Federation**

  A cloud computing environment must be capable of providing federated service providers which means that, these providers, must be capable of collaborating and resource sharing at any point irrespective of their type. This is usually needed when an organization extends its computing paradigm from the private to the public cloud. Moreover, This federation must be kept transparent so that the virtual application can be used on all the sites. This makes the application be handled remotely and allows it to migrate from one site to another. Apart from this, the federation must be carried out in a secure and independent way.

- **Independence**

  The user of cloud computing services must be independent of the provider's specific tool and the type of service. According to this principle, a user must be allowed the required virtual resource irrespective of the type of provider. Moreover, it is the responsibility of service providers to handle infrastructure while hiding confidential information.

- **Isolation**

  According to this principle, a service provider must ensure the user with respect to the isolation of their data from others. Even the data in the same cloud must be separated from different users and therefore should not be accessed.

8. Challenges and risks of cloud computing.

Ans :

1. Security

Security is a top concern for cloud computing. Organizations need to ensure that their data is protected from unauthorized access, data breaches, and other security threats. This requires implementing appropriate security controls, such as encryption, access controls, and monitoring.

2. Compliance

Organizations need to comply with various regulations and standards, such as GDPR, HIPAA, and PCI DSS, when storing and processing data in the cloud. Compliance requirements can vary by region and industry, and failure to comply can result in legal and financial penalties.

3. Data Privacy

Cloud computing can raise concerns about data privacy, especially if the data is stored or processed in a different country with different privacy laws. Organizations need to ensure that they have appropriate data privacy policies and controls in place.

9. Software as a Service.

Ans :

SaaS is also known as "on-demand software". It is a software in which the applications are hosted by a cloud service provider. Users can access these applications with the help of internet connection and web browser.

Characteristics of SaaS

There are the following characteristics of SaaS

- Managed from a central location
- Hosted on a remote server
- Accessible over the internet
- Users are not responsible for hardware and software updates. Updates are applied automatically.
- The services are purchased on the pay-as-per-use basis.

10. Platform as a Service (PaaS)

Ans :

PaaS cloud computing platform is created for the programmer to develop, test, run, and manage the applications.

Characteristics of PaaS

There are the following characteristics of PaaS

- Accessible to various users via the same development application.
- Integrates with web services and databases.
- Builds on virtualization technology, so resources can easily be scaled up or down as per the organization's need.
- Support multiple languages and frameworks.
- Provides an ability to "Auto-scale".

Example:

AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, Magento Commerce Cloud, and OpenShift.

Choose the Correct Answers

1. Among the following which layer consists of physical computing devices [a]

   (a) Platform (b) Application

   (c) Infrastructure (d) All the above

2. Which layer of cloud computing provides access to applications over the internet? [c]

   (a) Infrastructure as a Service (IaaS) (b) Platform as a Service (PaaS)

   (c) Software as a Service (SaaS) (d) Network as a Service (NaaS)

3. Which layer of cloud computing is responsible for providing data storage services? [d]

   (a) Infrastructure as a Service (IaaS) (b) Platform as a Service (PaaS)

   (c) Software as a Service (SaaS) (d) Data as a Service (DaaS)

4. Which layer of cloud computing is responsible for providing virtualization services? [a]

   (a) Infrastructure as a Service (IaaS) (b) Platform as a Service (PaaS)

   (c) Software as a Service (SaaS) (d) Anything as a Service (XaaS)

5. Which type of cloud deployment model is suitable for highly sensitive data or applications? [b]

   (a) Public cloud (b) Private cloud

   (c) Hybrid cloud (d) Community cloud

6. Which type of cloud computing model involves the integration of multiple cloud services and deployment models? [c]

   (a) Public cloud (b) Private cloud

   (c) Hybrid cloud (d) Community cloud

7. Among the following which feature is needed when an organization extends its computing paradigm from the private to the public cloud [b]

   (a) Federation (b) Elasticity

   (c) Isolation (d) Trust

8. Which challenge in cloud computing refers to the need to ensure that data is protected from unauthorized access, use, disclosure, or destruction? [a]

   (a) Security (b) Data privacy

   (c) Compliance (d) Vendor lock-in

9. Which challenge in cloud computing refers to the need to comply with legal, regulatory, and contractual requirements?

   (a) Security (b) Data privacy

   (c) Compliance (d) Vendor lock-in

10. Which cloud service model provides customers with access to software applications over the internet? [c]

    (a) Infrastructure as a Service (IaaS) (b) Platform as a Service (PaaS)

    (c) Software as a Service (SaaS) (d) Function as a Service (FaaS)

Fill in the Blanks

1. ______ enables companies to consume a compute resource, such as a virtual machine (VM), storage or an application, as a utility — just like electricity — rather than having to build and maintain computing infrastructures in house.

2. ______ layer generally provides interfaces for interaction with external systems

3. A ______ is a group of machines that are virtually or geographically separated and that work together to provide the same service or application to clients.

4. ______ layer where the physical and virtual resources of the cloud computing system are located, including servers, storage, and networks.

5. ______ are distributed systems that work on private infrastructure and provide the users with dynamic provisioning of computing resources.

6. ______ are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector.

7. The ______ network is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping

8. ______ cloud service model provides customers with a complete platform for developing, deploying, and managing applications without the need for infrastructure management?

9. ______ is used for widely distributed computer resources to reach a common goal

10. In a ______ system, every node acts as both a client and a server, providing part of the system resources

Answers

1. Cloud computing
2. Appl;ication layer
3. Distributed systems cluster
4. Infrastructure layer
5. Private clouds
6. Community clouds
7. Overlay
8. Platform as a service
9. Computing grids
10. P2P

UNIT IV

Virtual Machines and Virtualization of Clusters and Data Centers: Levels of Virtualization, Virtualization Structures Tools and Mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Virtualization Data-Center Automation. Case studies: Xen Virtual machine monitors- Xen API. VMware - VMware products-Vmware Features.

4.1 Virtual Machines and Virtualization of Clusters and Data Centers

4.1.1 Levels Of Virtualization

Q1. Explain about the levels of virtualization.

Ans : **(Imp.)**

Levels of Virtualization

A traditional computer runs with a host operating system specially tailored for its hardware architecture, as shown in Figure(a). 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 as shown in Figure (b). This virtualization layer is known as hypervisor or virtual machine monitor. 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.

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 the instruction set architecture (ISA) level, hardware level, operating system level, library support level, and application level .

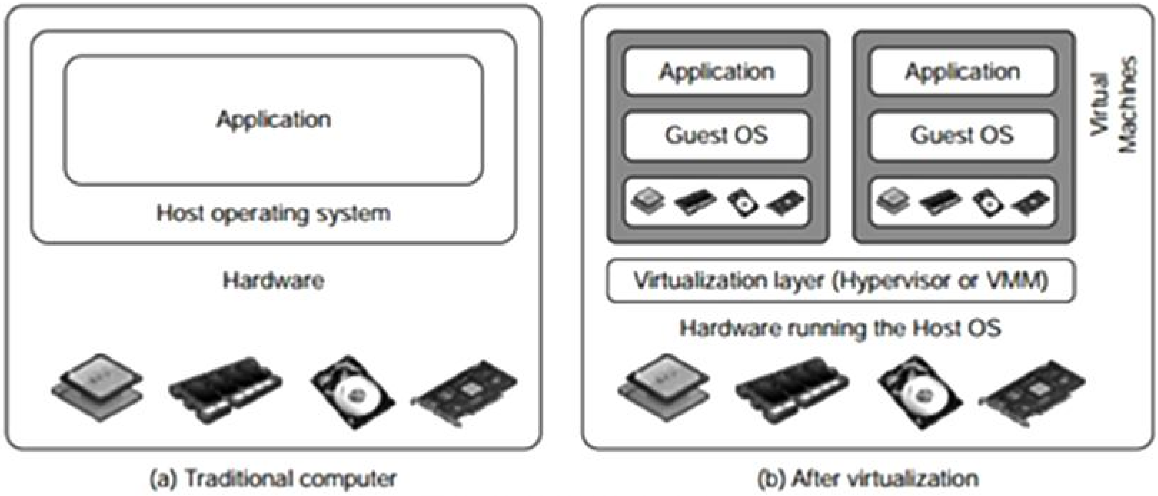


Fig.: The architecture of a computer system before and after virtualization

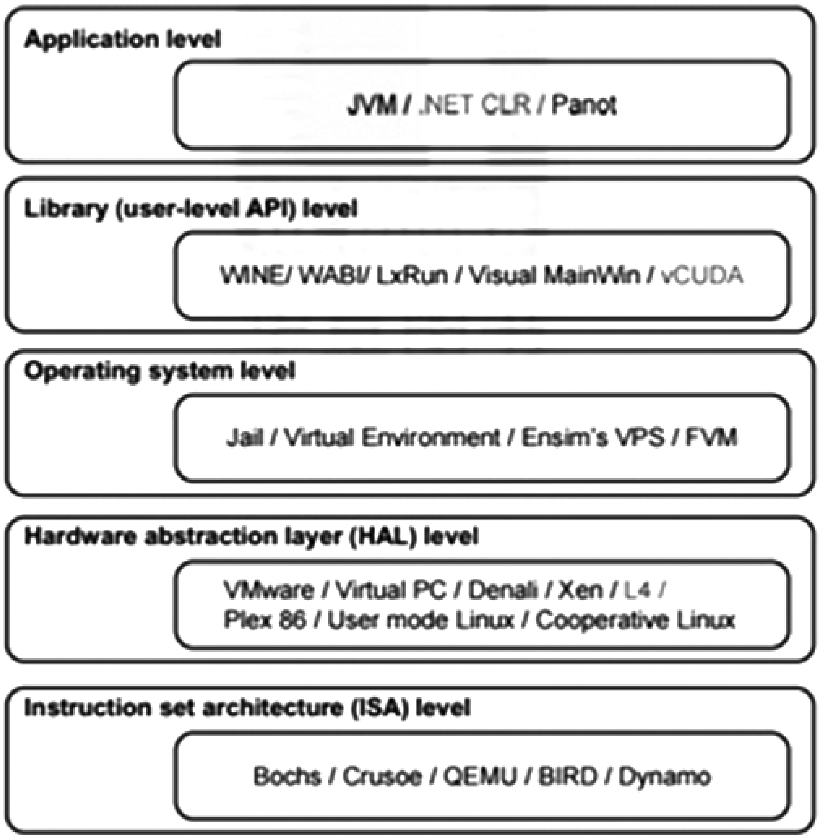

Application level
JVM / .NET CLR / Panot
Library (user-level API) level
WINE/ WABI/ LxRun / Visual MainWin / vCUDA
Operating system level
Jail / Virtual Environment / Ensim's VPS / FVM
Hardware abstraction layer (HAL) level
VMware / Virtual PC / Denali / Xen / L4 /
Plex 86 / User mode Linux / Cooperative Linux
Instruction set architecture (ISA) level
Bochs / Crusoe / QEMU / BIRD / Dynamo

Fig.: Virtualization ranging from hardware to applications

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.

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.

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 hard-ware 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.

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.

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.

4.1.2 Virtualization Structures Tools And Mechanisms

Q2. Explain about virtualization structures tools and mechanisms.

Ans : **(Imp.)**

- After virtualization a virtualization layer is inserted between the hardware and the OS.
- The virtualization layer converts portions of the real hardware into virtual ardware.
- Thus 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
- The hypervisor architecture
- Paravirtualization
- host-based virtualization.
- Hypervisor
- Hardware virtualization technique allowing multiple OS called guests to run on a host machine.
- Also called the Virtual Machine Monitor (VMM).
- Supports hardware-level virtualization on bare metal devices like CPU, memory, disk and network interfaces.
- Sits directly between the physical hardware and its OS.
- Provides hypercalls for the guest OSes and applications.
- Assumes a micro-kernel architecture like the Microsoft Hyper-V.
- Includes only the basic and unchanging functions (such as physical memory management and processor scheduling).
- The device drivers and other changeable components are outside the hypervisor.
- Or it can assume a monolithic hypervisor architecture like the VMware ESX for server virtualization.
- Implements all the above functions, including those of the device drivers.
- So the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithic hypervisor.
- A hypervisor must be able to convert physical devices into virtual resources dedicated for the deployed VM to use.

Xen Architecture

Microkernel hypervisor does not include any device drivers natively. “Virtualization Structures/Tools and Mechanisms” Before virtualization - OS manages the hardware. After virtualization - a virtualization layer is inserted between the hardware and the OS. The virtualization layer converts portions of the real hardware into virtual ardware.” Thus 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” the hypervisor architecture” paravirtualization” host-based virtualization.” Hypervisor” Hardware virtualization technique allowing multiple OS called guests to run on a host machine.” Also called the Virtual Machine Monitor (VMM).” Supports hardware-level virtualization on bare metal devices like CPU, memory, disk and networkinterfaces” Sits directly between the physical hardware and its OS.” Provides hypercalls for the guest OSes and applications.” Assumes a micro-kernel architecture like the Microsoft Hyper-V.” Includes only the basic and unchanging functions (such as physical memory management and processor scheduling).” The device drivers and other changeable components are outside the hypervisor” Or it can assume a monolithic hypervisor architecture like the VMware ESX for server virtualization.” implements all the above functions, including those of the device drivers.” So the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithichypervisor.” A hypervisor must be able to convert physical devices into virtual resources dedicated for the deployed VM to use. Xen Architecture” Microkernel hypervisor does not include any device drivers natively. Provides a mechanism by which a guest OS can have direct access to the physical.

- Before virtualization - OS manages the hardware.
- After virtualization - a virtualization layer is inserted between the hardware and the OS
- The virtualization layer converts portions of the real hardware into virtual hardware.
- Thus 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
    - the hypervisor architecture
    - paravirtualization
    - host-based virtualization.

Virtualization Structures/Tools and Mechanisms

- Before virtualization - OS manages the hardware.
- After virtualization - a virtualization layer is inserted between the hardware and the OS.
- The virtualization layer converts portions of the real hardware into virtual ardware.
- Thus 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
    - the hypervisor architecture
    - paravirtualization
    - host-based virtualization.

Hypervisor

- Hardware virtualization technique allowing multiple OS called guests to run on a host machine.
- Also called the Virtual Machine Monitor (VMM)
- Supports hardware-level virtualization on bare metal devices like CPU, memory, disk and networkinterfaces
- Sits directly between the physical hardware and its OS.” Provides hypercalls for the guest OSes and applications.
- Assumes a micro-kernel architecture like the Microsoft Hyper-V.
- Includes only the basic and unchanging functions (such as physical memory management and processor scheduling).
- The device drivers and other changeable components are outside the hypervisor
- Or it can assume a monolithic hypervisor architecture like the VMware ESX for server virtualization.
- Implements all the above functions, including those of the device drivers.

- So the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithichypervisor.
- A hypervisor must be able to convert physical devices into virtual resources dedicated for the deployed VM to use.

Xen Architecture

- Microkernel hypervisor does not include any device drivers natively.
  - Provides a mechanism by which a guest OS can have direct access to the physical devices. So the size of the Xen hypervisor is kept rather small.
- Xen provides a virtual environment located between the hardware and the OS.

Core components of a Xen system are:

- Hypervisor
- kernel applications
- Binary Translation with Full Virtualization

Depending on implementation technologies, hardware virtualization can be classified into two categories:

- full virtualization
- host-based virtualization

Full virtualization no need to modify the host OS.

Relies on binary translation to trap and to virtualize the execution of certain sensitive, non-virtualizable instructions.

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.

Full Virtualization

With full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software.

- Binary translation can incur a large performance overhead.
- Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do.
- Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.
- Binary Translation of Guest OS Requests Using a VMM" VMware: puts the VMM at Ring 0 and the guest OS at Ring 1.
- The VMM scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions.
- Once identified trapped into the VMM emulates the behaviour of these instructions.
- The method used in this emulation is called binary translation.
- Full virtualization combines binary translation and direct execution.
- The guest OS is completely decoupled from the underlying hardware and so is unaware that it is being virtualized.
- The performance of full virtualization may not be ideal, because it involves binary translation which is rather time-consuming.
- In particular, the full virtualization of I/O-intensive applications is a really a big challenge.
- Binary translation employs a code cache to store translated hot instructions to improve performance, but it increases the cost of memory usage.

Host Based Virtualization

- Alternative VM architecture install a virtualization layer on top of the host OS.
- This host OS is still responsible for managing the hardware.
- The guest OSes are installed and run on top of the virtualization layer.

- Dedicated applications may run on the VMs.
- Certainly, some other applications can also run with the host OS directly.

Advantages:

- The user can install this VM architecture without modifying the host OS.
- OS installs drivers and other low-level services simplify the VM design and ease its deployment.
- Host-based approach appeals to many host machine configurations.
- Para-Virtualization with Compiler Support
- 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.
- Para-virtualization attempts to reduce the virtualization overhead - performance is improved by modifying only the guest OS kernel.
- The guest operating systems are para-virtualized.
- And are assisted by an intelligent compiler to replace the nonvirtualizable OS instructions by hypercalls

4.1.3 Virtualization of CPU

Q3. What is CPU Virtualization? Explain different types of virtualization.

Ans :

Meaning

CPU Virtualization emphasizes running programs and instructions through a virtual machine, giving the feeling of working on a physical workstation. All the operations are handled by an emulator that controls software to run according to it. Nevertheless, CPU Virtualization does not act as an emulator. The emulator performs the same way as a normal computer machine does. It replicates the same copy or data and generates the same output just like a physical machine does. The emulation function offers great portability and facilitates working on a single platform, acting like working on multiple platforms.

With CPU Virtualization, all the virtual machines act as physical machines and distribute their hosting resources like having various virtual processors. Sharing of physical resources takes place to each virtual machine when all hosting services get the request. Finally, the virtual machines get a share of the single CPU allocated to them, being a single-processor acting as a dual-processor.

Types

The various types of CPU virtualization available are as follows.

1. Software-Based CPU Virtualization

This CPU Virtualization is software-based where with the help of it, application code gets executed on the processor and the privileged code gets translated first, and that translated code gets executed directly on the processor. This translation is purely known as Binary Translation (BT). The code that gets translated is very large in size and also slow at the same time on execution. The guest programs that are based on privileged coding runs very smooth and fast. The code programs or the applications that are based on privileged code components that are significant such as system calls, run at a slower rate in the virtual environment.

2. Hardware-Assisted CPU Virtualization

There is hardware that gets assistance to support CPU Virtualization from certain processors. Here, the guest user uses a different version of code and mode of execution known as a guest mode. The guest code mainly runs on guest mode. The best part in hardware-assisted CPU Virtualization is that there is no requirement for translation while using it for hardware assistance. For this, the system calls runs faster than expected. Workloads that require the updation of page tables get a chance of exiting from guest mode to root mode that eventually slows down the program's performance and efficiency.

3. Virtualization and Processor-Specific Behavior

Despite having specific software behavior of the CPU model, the virtual machine still helps in detecting the processor model on which the system runs. The processor model is different based on the CPU and the wide variety of features it offers, whereas the applications that produce the output generally utilize such features. In such cases, vMotion cannot be used to migrate the virtual machines that are running on feature-rich processors. Enhanced vMotion Compatibility easily handles this feature.

4. Performance Implications of CPU Virtualization

CPU Virtualization adds the amount of overhead based on the workloads and virtualization used. Any application depends mainly on the CPU power waiting for the instructions to get executed first. Such applications require the use of CPU Virtualization that gets the command or executions that are needed to be executed first. This overhead takes the overall processing time and results in an overall degradation in performance and CPU virtualisation execution.

Q4. Why CPU Virtualization is Important?

Ans :

CPU Virtualization is important in lots of ways, and its usefulness has been widespread in the cloud computing industry. I will brief regarding the advantages of using CPU Virtualization, stated as below:

- Using CPU Virtualization, the overall performance and efficiency are improved to a great extent because it usually takes virtual machines to work on a single CPU, sharing resources acting like using multiple processors at the same time. This saves cost and money.
- As CPU Virtualization uses virtual machines to work on separate operating systems on a single sharing system, security is also maintained by it. The machines are also kept separate from each other. Because of that, any cyber-attack or software glitch is unable to damage the system, as a single machine cannot affect another machine.
- It purely works on virtual machines and hardware resources. It consists of a single server where all the computing resources are stored, and processing is done based on the CPU's instructions that are shared among all the systems involved. Since the hardware requirement is less and the physical machine usage is absent, that is why the cost is very less, and timing is saved.
- It provides the best backup of computing resources since the data is stored and shared from a single system. It provides reliability to users dependent on a single system and provides greater retrieval options of data for the user to make them happy.
- It also offers great and fast deployment procedure options so that it reaches the client without any hassle, and also it maintains the atomicity. Virtualization ensures the desired data reach the desired clients through the medium and checks any constraints are there, and are also fast to remove it.

4.1.4 Memory and I/O Devices

Q5. Explain about Memory Virtualization.

Ans : **(Imp.)**

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

That means 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. Furthermore, MMU virtualization should be supported, which is transparent to the guest OS. The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. But the guest OS cannot directly access the actual machine memory. The VMM is responsible for mapping the guest physical memory to the actual machine memory. Figure: shows the two-level memory mapping procedure.

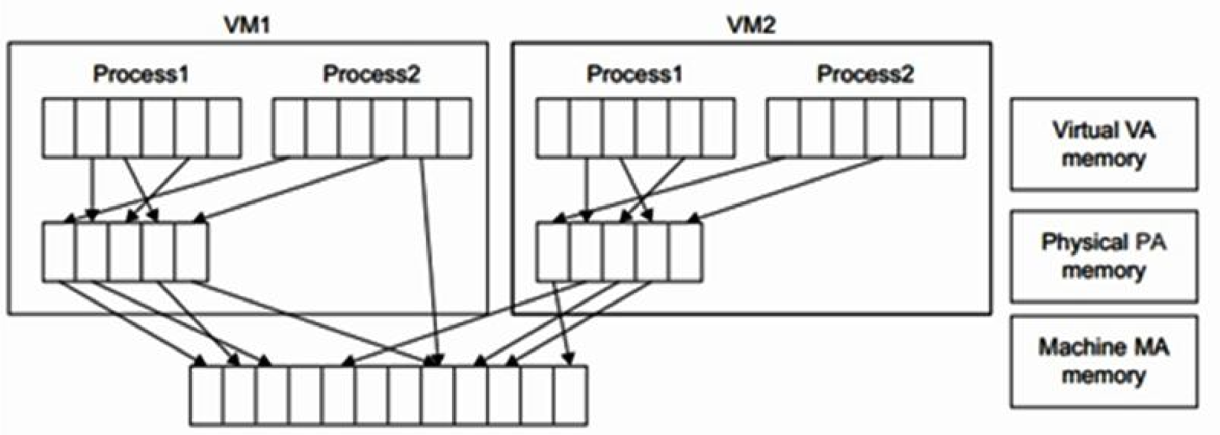


Fig. : Two level memory mappling

Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the shadow page table. Nested page tables add another layer of indirection to virtual memory. 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.

VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation. 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.

Q6. Write about I/O virtualization.

Ans :

I/O Virtualization

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

There are three ways to implement I/O virtualization:

(i) Full Device Emulation

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. The full device emulation approach is shown in Figure .

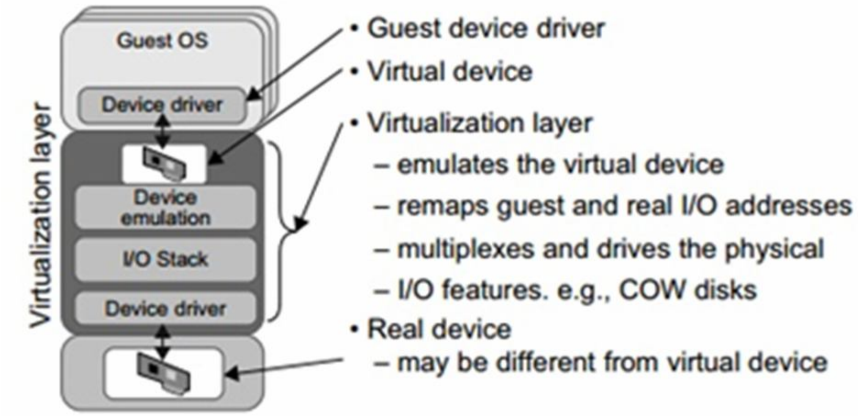

Guest OS
Device driver
Virtualization layer
Device emulation
I/O Stack
Device driver
• Guest device driver
• Virtual device
• Virtualization layer
– emulates the virtual device
– remaps guest and real I/O addresses
– multiplexes and drives the physical
– I/O features. e.g., COW disks
• Real device
– may be different from virtual device

Fig. : Device Emulation for I/O virtualization

(ii) Para virtualization

A single hardware device can be shared by multiple VMs that run concurrently. 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. Although para-I/O-virtualization achieves better device performance than full device emulation, it comes with a higher CPU overhead.

(iii) Direct I/O

Direct I/O virtualization lets the VM access devices directly. It can achieve close-to-native performance without high CPU costs. However, current direct I/O virtualization implementations focus on networking for mainframes. There are a lot of challenges for commodity hardware devices.

Another way to help I/O virtualization is via self-virtualized I/O]. 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.

4.1.5 Virtual Clusters and Resource Management

Q7. Explain about various virtual clusters.

Ans : **(Imp.)**

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

1 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.

- The virtual cluster boundaries are shown as distinct boundaries.
- The provisioning of VMs to a virtual cluster is done dynamically to have the following 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.
- Since system virtualization has been widely used, it is necessary to
  - effectively manage VMs running on a mass of physical computing nodes (also called virtual clusters) and
  - build a high-performance virtualized computing environment.
- This involves
  - virtual cluster deployment.
  - monitoring and management over large-scale clusters.
  - resource scheduling
  - load balancing
  - server consolidation
  - fault tolerance
- There are common installations for most users or applications, such as operating systems or user-level programming libraries.
- These software packages can be preinstalled as templates (called template VMs).
- With these templates, users can build their own software stacks.
- New OS instances can be copied from the template VM.
- User-specific components such as programming libraries and applications can be installed to those instances.
- Three physical clusters are shown on the left side of Figure.
- Four virtual clusters are created on the right, over the physical clusters.
- The physical machines are also called host systems.
- In contrast, the VMs are guest systems.
- The host and guest systems may run with different operating systems.
- Each VM can be installed on a remote server or replicated on multiple servers belonging to the same or different physical clusters.
- The boundary of a virtual cluster can change as VM nodes are added, removed, or migrated dynamically over time.

Fast Deployment and Effective Scheduling

- The system should have the capability of fast deployment.
- Here, deployment means two things:
  - to construct and distribute software stacks (OS, libraries, applications) to a physical node inside clusters as fast as possible,

  - to quickly switch runtime environments from one user's virtual cluster to another user's virtual cluster.
- If one user finishes using his system, the corresponding virtual cluster should shut down or suspend quickly to save the resources to run other VMs for other users.
- The concept of "green computing" has attracted much attention recently.

High-Performance Virtual Storage

- The template VM can be distributed to several physical hosts in the cluster to customize the VMs.
- Basically, there are 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.
- Every VM is configured with
  - a name,
  - disk image,
  - network setting, and
  - allocated CPU and memory.
- Each VM configuration is recorded into a file.
- Most configuration items use the same settings, while some of them, such as UUID, VM name, and IP address, are assigned with automatically calculated values.
- The deployment principle is to fulfill the VM requirement and to balance workloads among the whole host network.

2. Live VM Migration Steps and Performance Effects

- In a cluster built with mixed nodes of host and guest systems, the normal method of operation is to run everything on the physical machine.
- Virtual clusters can be applied in computational grids, cloud platforms, and high-performance computing (HPC) systems.
- Virtual clustering provides dynamic resources that can be quickly put together upon user demand or after a node failure.
- In particular, virtual clustering plays a key role in cloud computing.
- There are four ways to manage a virtual cluster.
  - Cluster manager resides on a guest system
  - Cluster manager resides on the host systems. The host-based manager supervises the guest systems and can restart the guest system on another physical machine.
  - Use an independent cluster manager on both the host and guest systems – issue – makes infrastructure management more complex.
  - Use an integrated cluster on the guest and host systems. This means the manager must be designed to distinguish between virtualized resources and physical resources.
  - A VM can be in one of the following four states.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.

3. Migration of Memory, Files, and Network Resources

- Since clusters have a high initial cost of ownership,
  - Including space, power conditioning, and cooling equipment.
  - Leasing or sharing access to a common cluster is an attractive solution when demands vary over time.

- Shared clusters offer economies of scale and more effective utilization of resources by multiplexing.
- Early configuration and management systems focus on expressive and scalable mechanisms for defining clusters for specific types of service, and physically partition cluster nodes among those types.
- When one system migrates to another physical node, we should consider the following issues.
  - Memory Migration
  - File System Migration
  - Network Migration
  - Live Migration of VM Using Xen

Memory Migration

- This is one of the most important aspects of VM migration.
- Moving the memory instance of a VM from one physical host to anotherdepend upon the characteristics of application/workloads supported by the guest OS.
- Memory migration can be in a range of hundreds of megabytes to a few gigabytes in a typical system today, andit needs to be done in an efficient manner.
- The Internet Suspend-Resume (ISR) technique exploits temporal locality as memory states are likely to have considerable overlap in the suspended and the resumed instances of a VM.
- Temporal locality refers to the fact that the memory states differ only by the amount of work done since a VM was last suspended before being initiated for migration.

File System Migration

To support VM migration, a system must provide each VM with a consistent, location-independent view of the file system that is available on all hosts.

Solution 1

- A simple way to achieve this is to provide each VM with its own virtual disk which the file system is mapped to and transport the contents of this virtual disk along with the other states of the VM.
- However, due to the current trend of high capacity disks, migration of the contents of an entire disk over a network is not a viable solution.

Network Migration

- A migrating VM should maintain all open network connections without relying on forwarding mechanisms on the original host or on support from mobility or redirection mechanisms.
- To enable remote systems to locate and communicate with a VM, each VM must be assigned a virtual IP address known to other entities.
- This address can be distinct from the IP address of the host machine where the VM is currently located.
- Each VM can also have its own distinct virtual MAC address.
- The VMM maintains a mapping of the virtual IP and MAC addresses to their corresponding VMs. (ARP Table)

Solution 1 – Virtual IP and MAC address

- In general, a migrating VM includes all the protocol states and carries its IP address with it.
- If the source and destination machines of a VM migration are typically connected to a single switched LAN, an unsolicited ARP reply from the migrating host is provided advertising that the IP has moved to a new location.
- This solves the open network connection problem by reconfiguring all the peers to send future packets to a new location.
- Although a few packets that have already been transmitted might be lost, there are no other problems with this mechanism.

Q8. Explain about resource management in cloud computing

Ans :

Resource Management Models in Cloud Computing

The term resource management refers to the operations used to control how capabilities provided by

Cloud resources and services are made available to other entities, whether users, applications, or services.

Types of Resources

- **Physical Resource**

  Computer, disk, database, network, etc.

- **Logical Resource**

  Execution, monitoring, and application to communicate

Resource Management in Cloud Computing Environment

On the Cloud Vendor's View

- Provision resources on an on-demand basis.
- Energy conservation and proper utilization is maintained in Cloud Data Centers

On the Cloud Service Provider's View

- To make available the best performance resources at the cheapest cost.
- QoS (Quality of Service) to their cloud users

On the Cloud User's View

- Renting resources at a low price without compromising performance.
- Cloud provider guarantees to provide a minimum level of service to the user.

Resource Management Models

Compute Model

Resource in the cloud is shared by all users at the same time. It allows the user to reserve the VM's memory to ensure that the memory size requested by the VM is always available to operate locally on clouds with a good enough level of QoS (Quality of Service) being delivered to the end user.

Grid Strictly manages the workload of computing mode. Local resource manager such as Portable Batch System, Condor, and Sun Grid Engine manages the compute resource for the Grid site. Identify the user to run the job.

Data Model

It is related to plotting, separating, querying, transferring, caching, and replicating data.

- **Data is Stored at an Un-Trusted Host**

  Although may not seem the best policy to store data and let others use the data without permission moving data off-premises increases the number of potential security risks.

- **Data Replication over Large Areas**

  Making sure data is available and durable whenever demanded is of utmost importance for cloud storage providers. Data availability and durability are typically achieved through under-the-covers replication i.e., data is automatically replicated without customer interference or requests.

- **Problems with Data Management**

  Transactional data management is one of the biggest data management problems. It is hard to ensure Atomicity, Consistency, Isolation, and Durability is maintained during data replication over large distances. It is also risky to store such sensitive data in untrusted storage.

- **Virtualization**

  It is the method by which we can create an emulation of software or hardware on our computer. It has further two components:-

- **Abstraction**

  Provides the necessary virtual versions of raw compute, storage, and network that can be unified as a pool of resources and resource overlay which includes data storage services, and a web hosting environment.

- **Encapsulation**

  A virtual machine can be represented as a single file. Virtualization configures, deploys, starts, migrates, suspends, resumes, and stops in each application. Provides better security, manageability, and isolation.

Monitoring

The challenge that virtualization brings is that users don't have a lot of control over the monitoring resource. It is a method of reviewing, observing, and managing the operation of a cloud-based IT infrastructure.

- **In Cloud**

  Different levels of services can be offered to end users. The user is only exposed to a limited Application Programming Interface. And lower-level resources are not revealed to the user (PaaS, SaaS level some providers may choose to expose monitoring information at these levels). The user does not have the capability to implement a new application of its own monitoring infrastructure. Limited information returned to users restricts their knowledge about the current status of the resource. Require to maintain business tracking, update, inspect and troubleshoot the servers of the cloud organization, monitor virtual machines, and maintain the functioning of the hardware.

- **In Grid**

  Have a different trust model in which users via their identity delegation can access and browse resources at different Grid sites and Grid resources are not highly abstracted and virtualized as in Clouds.

Programming Model

User-level programming languages are used for accessing and operating the cloud.

- **In Cloud**

  Makes use of Web Services where users have more control over the Cloud Services. The translation of data for the receiving system and real-time data exchange between systems without middleware of all the services and applications remain a big challenge.

- **In Grid**

  Makes use of parallel and distributed computing environment.

Challenges

1. Multiple service providers allow to access data to clients with little authorization or authentication.
2. Diversity in resources in turn affects the performance and stability.
3. Error handling in a continuously changing business environment.

Security Model

Allows users to control the security of their own data by maintaining passwords, and receiving any news regarding suspicious activity with their data via email.

Risks in the Security Model

1. Privileged use access
2. Regulatory compliance
3. Data location
4. Data partition
5. Recovery
6. Investigation support
7. Long-term durability

4.1.6 Virtualization

Q9. Write about some other types of virtualization techniques.

Ans :

Virtualization of computing infrastructure is not only about machine or server virtualization. Especially cloud-based service development requires each and every computing infrastructure in virtualized mode. In reality, virtualized network infrastructure, storage systems etc. are as important as the server virtualization.

1. Network Virtualization

The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.

Network virtualization provides a facility to create and provision virtual networks logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

2. Desktop Virtualization

Desktop virtualization allows the users' OS to be remotely stored on a server in the data centre. It allows the user to access their desktop virtually, from any location by a different machine. Users who want specific operating systems other than Windows Server will need to have a virtual desktop. Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates, and patches.

3. Storage Virtualization

Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren't aware of exactly where their data is stored, and instead function more like worker bees in a hive. It makes managing storage from multiple sources to be managed and utilized as a single repository. storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

4.1.7 Data Center Automation

Q10. Explain the concept of data centre automation.

Ans :

Data centre automation is a critical aspect of cloud computing, which refers to the use of automated tools and processes to manage and control data centre operations. Data centre automation in cloud computing environments is essential for managing the large and complex IT infrastructures that support cloud services. Here are some ways in which data center automation is used in cloud computing:

1. Provisioning

Provisioning refers to the process of setting up and configuring the hardware and software resources needed to support cloud services. Data center automation tools are used to automate the provisioning process, which can help to reduce the time and effort required to deploy new services and improve service delivery times.

2. Resource allocation

Data center automation tools are used to manage and allocate resources, such as CPU, memory, storage, and network bandwidth, to support cloud services. Automation can help to optimize resource utilization, reduce waste, and improve service performance.

3. Monitoring and Management

Data center automation tools can be used to monitor and manage the performance of cloud services, including resource usage, system health, and application performance. Automation can help to detect and resolve issues more quickly, reducing downtime and improving service availability.

4. Scaling

Data center automation tools can be used to automatically scale cloud services up or down based on demand. Automation can help to ensure that the right amount of resources are available to support the workload, improving service performance and reducing costs.

5. Security

Data center automation tools can be used to automate security processes, such as patch management, vulnerability scanning, and access control. Automation can help to reduce the risk of security breaches and improve compliance with industry and regulatory standards.

In summary, data center automation is a critical component of cloud computing, enabling organizations to manage and control the large and complex IT infrastructures that support cloud services. By automating provisioning, resource allocation, monitoring, scaling, and security processes, organizations can improve service delivery times, optimize resource utilization, reduce downtime, and improve service performance and security.

4.2 Case Studies: Xen Virtual Machine Monitors

4.2.1 XEN API

Q11. Explain XEN API with architecture diagram.

Ans : **(Imp.)**

➢ A Xen Project Toolstack that exposes the XAPI interface. When we refer to XAPI as a toolstack, we typically include all dependencies and components that are needed for XAPI to operate (e.g. xenopsd).

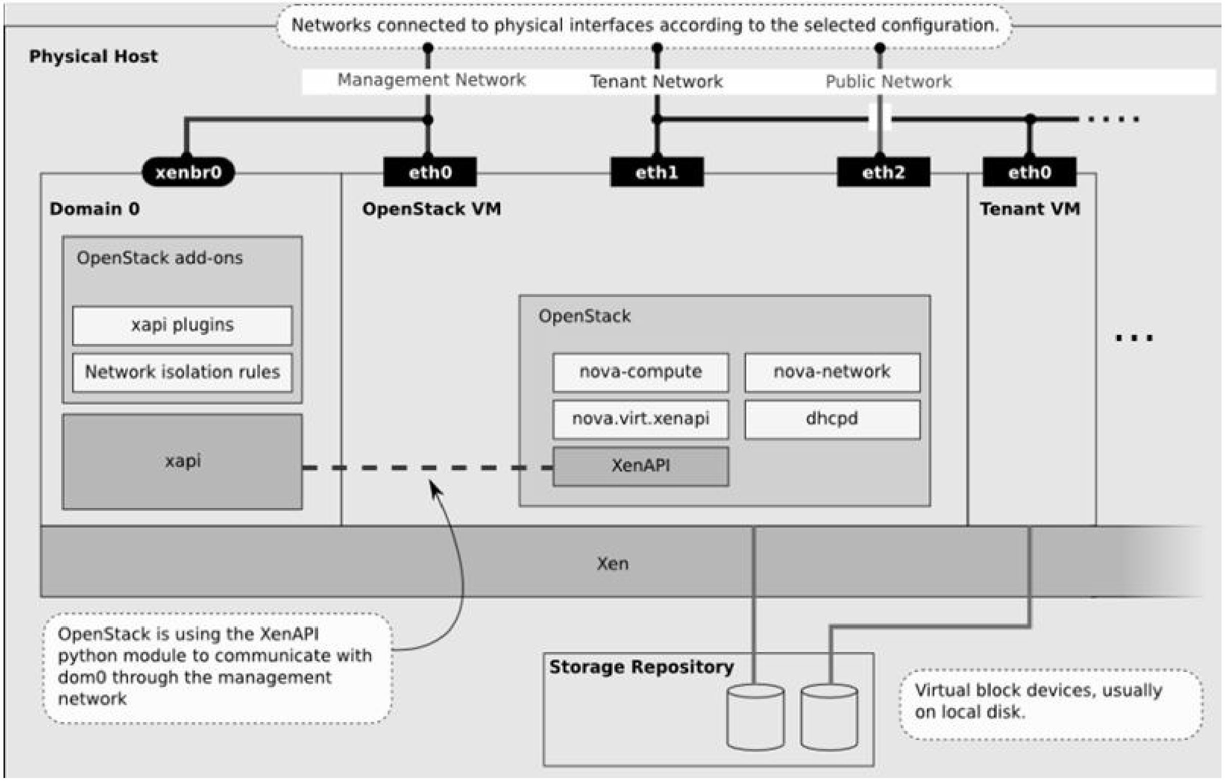
- An interface for remotely configuring and controlling virtualized guests running on a Xen-enabled host. XAPI is the core component of Citrix Hypervisor / XenServer, Magrana® Server, and XCP-ng.

  XAPI adds additional functionality compared to other Xen Project toolstacks, including:

- Extending the software to cover multiple hosts.
- Enhancing the VM lifecycle, including live snapshots, VM checkpointing, and VM migration.
- Enabling resource pools to include live migration, auto configuration, and disaster recovery.
- Allowing flexible storage and networking including integrated Open vSwitch support and storage XenMotion® live Migration (cross-pool migration, VDI migration).
- Enabling event tracking, with progress and notification.
- Creating upgrade and patching capabilities.
- Facilitating real-time performance monitoring and alerting.
- Integrations with cloud orchestration stacks.
- Built-in support and templates for Windows and Linux guests.

XenAPI Deployment Architecture

When you deploy OpenStack on XCP or XenServer, you get something similar to this:



Key things to note:

- The hypervisor: Xen
- Domain 0: runs xapi and some small pieces from OpenStack (some xapi plug-ins and network isolation rules). The majority of this is provided by XenServer or XCP (or yourself using Kronos).
- OpenStack VM: The nova-compute code runs in a paravirtualized virtual machine, running on the host under management. Each host runs a local instance of nova-compute. It will often also be running nova-network (depending on your network mode). In this case, nova-network is managing the addresses given to the tenant VMs through DHCP.
- Nova uses the XenAPI Python library to talk to xapi, and it uses the Management Network to reach from the domU to dom0 without leaving the host.

Some notes on the Networking

- The above diagram assumes FlatDHCnetworking (the DevStack default).
- There are three main OpenStack Networks:
  - Management network : RabbitMQ, MySQL, etc. Please note that the VM images are downloaded by the XenAPI plug-ins, so make sure that the images can be downloaded through the management network. It usually means binding those services to the management interface.
  - Tenant network : controlled by nova-network. The parameters of this network depend on the networking model selected (Flat, Flat DHCP, VLAN).
  - Public network - floating IPs, public API endpoints.
- The networks shown here must be connected to the corresponding physical networks within the data center. In the simplest case, three individual physical network cards could be used. It is also possible to use VLANs to separate these networks. Please note, that the selected configuration must be in line with the networking model selected for the cloud. (In case of VLAN networking, the physical channels have to be able to forward the tagged traffic.)

XenAPI pools

The host-aggregates feature enables you to create pools of XenServer hosts to enable live migration when using shared storage. However, you cannot configure shared storage.

4.3 VMware

Q12. Expalin about evolution of VMware.

Ans : **(Imp.)**

VMware is a software company that provides virtualization and cloud computing solutions to businesses of all sizes. One of the case studies that highlight the benefits of VMware's solutions is the case of the University of New Mexico Health Sciences Center (UNMHSC).

UNMHSC is a leading medical research institution that required a virtualization solution to manage its growing number of servers and applications. Prior to using VMware's solutions, the IT department at UNMHSC faced several challenges, including:

1. **Increasing Server Sprawl**

   As the number of applications and services increased, the organization's physical servers multiplied, leading to server sprawl.

2. **Difficulty in managing Multiple Operating Systems**

   UNMHSC's IT department was responsible for managing several different operating systems, including Windows, Linux, and Unix. This resulted in a lot of complexity and required significant effort to maintain and upgrade the servers.

3. **Downtime and Data Loss**

   UNMHSC faced downtime and data loss issues due to hardware failures and natural disasters.

To address these challenges, UNMHSC deployed VMware's virtualization solutions. VMware's solutions allowed UNMHSC to:

1. **Consolidate its Servers**

   By using VMware's virtualization solutions, UNMHSC was able to consolidate its servers and reduce server sprawl. This led to significant cost savings in terms of hardware, maintenance, and energy costs.

2. **Simplify Management**

   With VMware's solutions, UNMHSC's IT department was able to manage multiple operating systems from a single interface, reducing complexity and maintenance efforts.

3. **Ensure High availability and Disaster Recovery**

   VMware's virtualization solutions provided high availability and disaster recovery capabilities, ensuring that UNMHSC's critical applications remained available and data loss was minimized.

Overall, VMware's virtualization solutions helped UNMHSC to improve its IT infrastructure, reduce costs, and enhance its overall IT performance. The deployment of VMware's solutions has resulted in significant improvements in server utilization, performance, and availability, and has helped the organization to better meet the needs of its researchers and staff.

Q13. Explain the architecture of VMWare.

Ans :

VMware software is now commonly used in virtualized storage and networking, cloud management services, private data centers, desktop software, etc. VMware company also launched the VMWare ESX server and VMWare GSX Server. Mostly, VMWare virtualization software is for commercial use.

- **VMware ESXi Server**

This is an organization-level software built to provide better functionality than a freeware VMware Server to reduce system overhead. But, the service console in the VMware ESX server is replaced by BusyBox installation, which requires low disk space.

- **VMware ESX Server**

  This is similar to the VMware ESXi server where VMware ESX is combined with VMware vCenter produces additional solutions to improve the consistency and manageability of the server implementations.

  VMWare vSphere and VMWare Server are some of the important server virtualization software.

- **VMware vSphere**

  The vSphere is the best and well-known product from VMware. vSphere is the integration of well-known, commonly used vCenter, and ESXi hypervisor suite for applications.

- **VMware Server**

  It is an open-source (Freeware) software that can be used on any existing operating system like Ubuntu, LINUX, or Windows.

Physical Topology of Virtual Infrastructure (VI) Datacenter

A classic VMware vSphere data center contains (physical) building blocks like IP networks, desktop clients, storage networks and arrays, a management server, and x86 virtualization servers. The vSphere data center topology components are listed below:

- IP networks
- Management clients
- Compute servers
- Storage networks and arrays
- vCenter Server

- **IP networks**

  Any compute server can have one or more physical network adapters, and they provide reliable networking and high bandwidth to the whole VMware vSphere data center.

- **Management clients**

  They provide many interfaces to access a virtual machine, and for data center management. These interfaces incorporate either vSphere Command-Line Interface (vSphere CLI), or vSphere Web Client to access a web browser.

- **Compute servers**

  The x86 servers run ESXi on bare metal. Compute server provides resources for virtual machines to run. In the virtual environment, each computing server is called a standalone host.

  All servers configured with x86 are grouped together with a number of connections and can be connected to the same storage subsystems and network: A cluster (grouping of similarly configured servers creating a cumulative set of resources in the virtual environment) is formed.

- **Storage networks and arrays**

  NAS arrays, Fibre Channel SAN arrays, and iSCSI SAN arrays are the most frequently used storage technologies. VMware vSphere supports these techniques to meet data center storage requirements.

  Sharing of data between the group of servers is possible by connecting them through storage area networks. This aggregation provides more flexibility to virtual machines.

- **vCenter Server**

  The vCenter Server is a single point of control to the data center. It transfers essential data center services like configuration, access control, performance monitoring, etc. It merges all the resources from the individual computing servers and shares them with virtual machines in the whole data center.

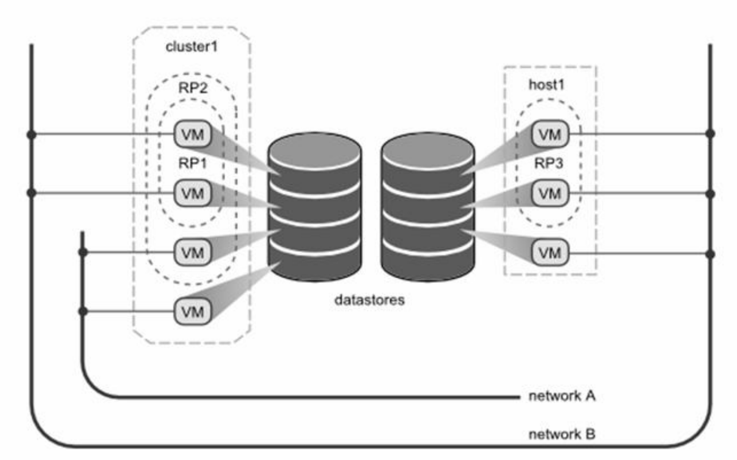
- vCenter Server manages all the tasks right from computing servers to virtual machines. The assignment of resources to the virtual machines is done by computing servers. The system administrator sets some policies based on which resources are assigned.

- When vCenter Server is not unreachable, still computing servers continue to function. For example, if the network is not severed, servers can be managed individually and continue to run the assigned virtual machines. After the connection is restored, the server functions within the network as usual.

Virtual Datacenter Architecture

The whole IT infrastructure including networks, servers, and storage is virtualized by VMware vSphere. These resources are accumulated and presented in a uniform set (elements) in the virtual environment.

The IT resources can be managed With VMware vSphere. The dynamic provision and shared utility are resources for different business projects.



In Virtual Datacenter Architecture, vSphere is used to configure, manage, and view the key elements. The key elements in a virtual data center are:

Hosts, resource pools, and clusters are regarded as memory and computing resources.

- Virtual machines.
- Datastores act as storage resources.
- Networks (Networking resources).

- **Host**

The virtual presentation of the memory resources and calculation of physical machines (PMs) running ESX/ESXi is called a Host.

- **Cluster**

When more than two PMs are connected together to manage and work as an entire entity, it is called cluster (The collaboration of computing and memory resources). Machines can be removed or added from a cluster dynamically.

- **Hierarchy of Resource Pools**

The partitioning from hosts and clusters into computing and memory resources is called resource pools.

- **Datastores**

The virtual representation of integrated physical storage resources in the datacenter.

These physical storage resources are:

- F-SAN (File Storage Area Network) disk arrays.
- SAN (Storage Area Network) disk arrays.
- NAS (Network Area Storage) arrays.
- SAS (Statistical Analysis System) disks of the server.

The virtual machines are connected to each other through the physical networks in the virtual environment outside of the virtual datacenter. Virtual machines are assigned to a particular cluster, resource pool, or host to create datastores.

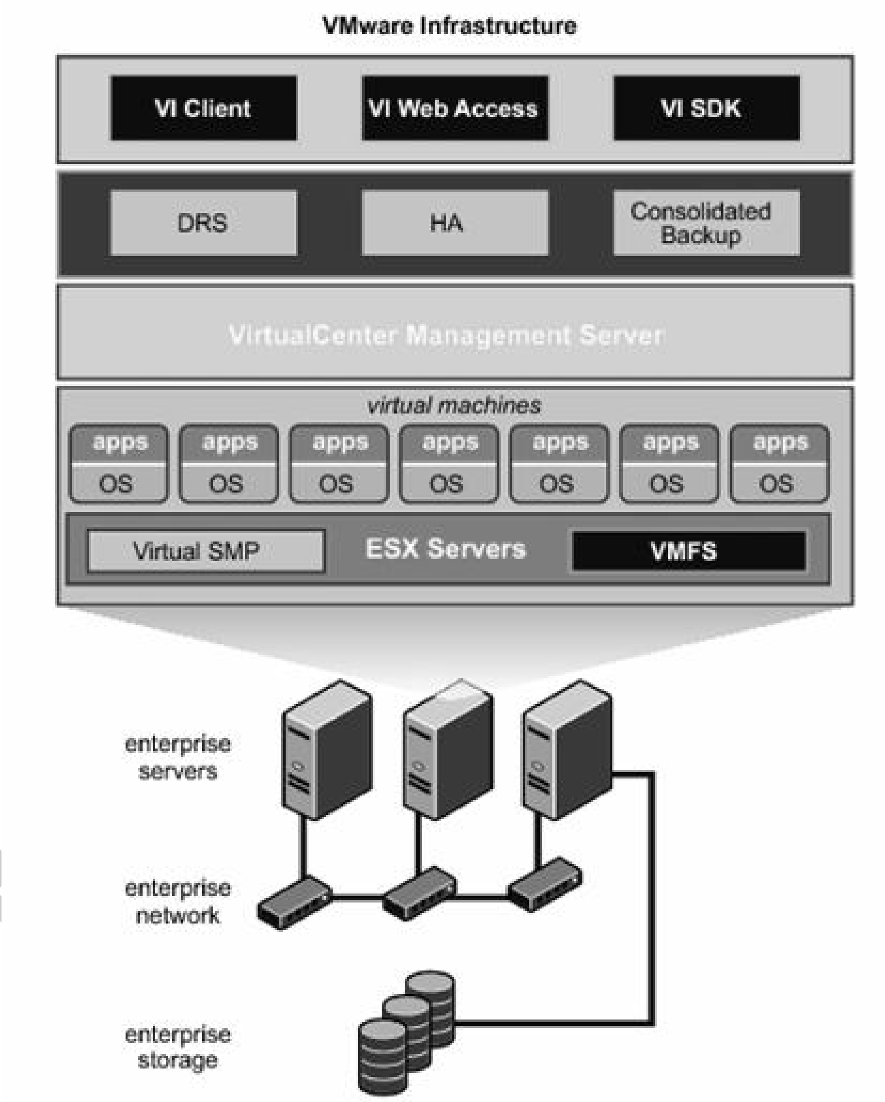
When a virtual machine is powered on, then workload increases, and it consumes resources dynamically. It reverts the resources dynamically as workload decreases. New virtual machines can be created in a fraction of a second, as it is much faster and easier than physical machines. An appropriate operating system and applications must be installed to alter the virtual machine to handle the workload.

Resources are allocated to virtual machines based on the principles set by the system administrator. Resources can be reserved for a particular virtual machine to guarantee its performance. Principles are given priority to set a differing portion of resources for every virtual machine.

VMware Infrastructure

VMware Infrastructure is a virtualization suite for infrastructure that provides a complete overview of application availability, virtualization, resource optimization, management, and operational automation. VMware Infrastructure integrates and virtualizes the hardware resources among two or more systems. In the virtual environment, it provides virtual resource pools to the data center.

In addition to this, VMware Infrastructure has a distributed architecture that provides excellent features like high availability, consolidated backup, fine-grain, policy-driven, and resource allocation of the entire virtual datacenter. To establish an IT enterprise, these distributed architecture services play a major role in meeting their service level agreements and production in a cost-effective manner.



- **Mware ESX Serve**r

  The virtualization layer which is production-proven, and robust is run on physical servers that abstract storage, memory, processor, and networking resources across multiple virtual machines (VMs).

- **VirtualCenter Management Server (VirtualCenter Server)**

  VirtualCenter Server is the central point for acceleration, managing, and configuring virtual environments.

- **VMware High Availability**

  Provides features that are cost-effective, easy-to-use, high availability to run applications on virtual machines.

- **Virtual Infrastructure Client (VI Client)**

  VirtualCenter Server or individual ESX Servers connect remotely from any Windows PC, and an interface is used by the client to connect.

- **Virtual Infrastructure Web Access**

  A virtual machine makes use of a web interface to manage and access remote consoles.

- **VMware Virtual Machine File System**

  ESX Server virtual machines have a high-performance file system cluster.

- **VMware Virtual Symmetric Multi-Processing (SMP)**

  SMP feature is enabled for a single virtual machine to convert into multiple physical processors to work parallelly.

- **VMware Virtual Motion**

  To run virtual machines from one physical server to another physical server, the VMotion feature is used, which allows live migration with complete transaction integrity, zero downtime, and continuous service availability.

  If the server fails, virtual machines automatically restart with production servers that have spare capacity.

- **VMware Backup**

  Consolidated Backup is easy to use. Backup can be simplified, and it reduces the overload on ESX Servers. Virtual machines have a centralized agent-free backup facility.

- **VMware Infrastructure SDK**

  SDK provides third-party access to the VMware Infrastructure and provides a standard interface for VMware.

- **Distributed Resource Scheduler (DRS)**

  To allocate resources for virtual machines, collection of hardware, and balancing the computing capacity dynamically done using DRS.

4.3.1 vmware products

Q14. What are different types of VMware products? Explain.

Ans : **(Imp.)**

VMware is a software company that provides a wide range of virtualization and cloud computing solutions to businesses of all sizes. Here are some of the key products offered by VMware:

1. VMware vSphere

VMware vSphere is a virtualization platform that allows organizations to consolidate their physical servers and run multiple virtual machines on a single server. It provides a range of features, including high availability, disaster recovery, and automated workload management.

2. VMware ESXi

VMware ESXi is a bare-metal hypervisor that provides a platform for running virtual machines. It is optimized for virtualization and provides a lightweight, secure, and scalable platform for running multiple virtual machines.

3. VMware vSAN

VMware vSAN is a software-defined storage solution that allows organizations to pool their storage resources and use them to run virtual machines. It provides high availability, data protection, and automated management.

4. VMware NSX

VMware NSX is a network virtualization and security platform that allows organizations to create virtual networks and micro-segmentation to secure their applications and data. It provides a range of network security features, including firewalling, intrusion detection, and distributed denial-of-service (DDoS) protection.

5. VMware Horizon

VMware Horizon is a virtual desktop infrastructure (VDI) solution that allows organizations to provide remote access to their applications and desktops. It provides a secure and scalable platform for delivering virtual desktops and applications to end-users.

6. VMware Cloud Foundation

VMware Cloud Foundation is a hybrid cloud infrastructure solution that allows organizations to run their applications and workloads on-premises or in the cloud. It provides a unified platform for managing hybrid cloud environments and automates the deployment and management of cloud infrastructure.

7. VMware Tanzu

VMware Tanzu is a portfolio of products and services for modernizing and running applications on Kubernetes. It provides a range of tools and services for building, deploying, and managing Kubernetes clusters and containerized applications.

These are some of the key products offered by VMware, and each product provides a range of features and capabilities to help organizations improve their IT infrastructure and operations.

4.3.2 Vmware Features

Q15. Explain the features of VMware?

Ans :

VMware is a software company that provides virtualization and cloud computing solutions to businesses of all sizes. Here are some of the key features of VMware's solutions:

1. Server Virtualization

VMware's server virtualization allows organizations to consolidate multiple physical servers into a single virtualized server, which leads to significant cost savings in terms of hardware, maintenance, and energy costs.

2. High Availability

VMware's solutions provide high availability and disaster recovery capabilities, ensuring that critical applications and data are always available and that downtime and data loss are minimized.

3. Scalability

VMware's solutions are designed to scale with the needs of the organization, allowing businesses to add resources as needed without disrupting existing operations.

4. Security

VMware's solutions provide a range of security features, including micro-segmentation, firewalls, intrusion detection, and distributed denial-of-service (DDoS) protection.

5. Automation

VMware's solutions automate many routine tasks, including workload management, resource allocation, and disaster recovery, reducing the workload of IT teams and improving efficiency.

6. Hybrid cloud Support

VMware's solutions support hybrid cloud environments, allowing businesses to run their applications and workloads on-premises or in the cloud, and providing a unified platform for managing hybrid cloud environments.

7. Compatibility

VMware's solutions are designed to work with a wide range of operating systems and applications, making it easier for businesses to migrate to virtualized environments without disrupting existing operations.

Overall, VMware's solutions provide a range of features that help businesses improve their IT infrastructure and operations, reduce costs, and enhance their overall IT performance.

Short Question and Answers

1. levels of virtualization.

Ans :

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.

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.

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 hard-ware 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.

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.

2. Host Based Virtualization

Ans :

- Alternative VM architecture install a virtualization layer on top of the host OS.
- This host OS is still responsible for managing the hardware.
- The guest OSes are installed and run on top of the virtualization layer.
- Dedicated applications may run on the VMs.
- Certainly, some other applications can also run with the host OS directly.

3. What is CPU Virtualization

Ans :

Meaning

CPU Virtualization emphasizes running programs and instructions through a virtual machine, giving the feeling of working on a physical workstation. All the operations are handled by an emulator that controls software to run according to it. Nevertheless, CPU Virtualization does not act as an emulator. The emulator performs the same way as a normal computer machine does. It replicates the same copy or data and generates the same output just like a physical machine does. The emulation function offers great portability and facilitates working on a single platform, acting like working on multiple platforms.

4. Explain different types of virtualization.

Ans :

1. Software-Based CPU Virtualization

This CPU Virtualization is software-based where with the help of it, application code gets executed on the processor and the privileged code gets translated first, and that translated code gets executed directly on the processor. This translation is purely known as Binary Translation (BT). The code that gets translated is very large in size and also slow at the same time on execution. The guest programs that are based on privileged coding runs very smooth and fast. The code programs or the applications that are based on privileged code components that are significant such as system calls, run at a slower rate in the virtual environment.

2. Hardware-Assisted CPU Virtualization

There is hardware that gets assistance to support CPU Virtualization from certain processors. Here, the guest user uses a different version of code and mode of execution known as a guest mode. The guest code mainly runs on guest mode. The best part in hardware-assisted CPU Virtualization is that there is no requirement for translation while using it for hardware assistance. For this, the system calls runs faster than expected. Workloads that require the updation of page tables get a chance of exiting from guest mode to root mode that eventually slows down the program's performance and efficiency.

3. Virtualization and Processor-Specific Behavior

Despite having specific software behavior of the CPU model, the virtual machine still helps in detecting the processor model on which the system runs. The processor model is different based on the CPU and the wide variety of features it offers, whereas the applications that produce the output generally utilize such features. In such cases, vMotion cannot be used to migrate the virtual machines that are running on feature-rich processors. Enhanced vMotion Compatibility easily handles this feature.

4. Performance Implications of CPU Virtualization

CPU Virtualization adds the amount of overhead based on the workloads and virtualization used. Any application depends mainly on the CPU power waiting for the instructions to get executed first. Such applications require the use of CPU Virtualization that gets the command or executions that are needed to be executed first. This overhead takes the overall processing time and results in an overall degradation in performance and CPU virtualisation execution.

5. Memory Virtualization.

Ans :

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

7. Network Virtualization

Ans :

The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.

Network virtualization provides a facility to create and provision virtual networks logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

8. Desktop Virtualization

Ans :

Desktop virtualization allows the users' OS to be remotely stored on a server in the data centre. It allows the user to access their desktop virtually, from any location by a different machine. Users who want specific operating systems other than Windows Server will need to have a virtual desktop. Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates, and patches.

9. Storage Virtualization

Ans :

Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren't aware of exactly where their data is stored, and instead function more like worker bees in a hive. It makes managing storage from multiple sources to be managed and utilized as a single repository. storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

10. XEN API

Ans :

- A Xen Project Toolstack that exposes the XAPI interface. When we refer to XAPI as a toolstack, we typically include all dependencies and components that are needed for XAPI to operate (e.g. xenopsd).
- An interface for remotely configuring and controlling virtualized guests running on a Xen-enabled host. XAPI is the core component of Citrix Hypervisor / XenServer, Magrana® Server, and XCP-ng.
- XAPI adds additional functionality compared to other Xen Project toolstacks, including:
- Extending the software to cover multiple hosts.
- Enhancing the VM lifecycle, including live snapshots, VM checkpointing, and VM migration.
- Enabling resource pools to include live migration, auto configuration, and disaster recovery.
- Allowing flexible storage and networking including integrated Open vSwitch support and storage XenMotion® live Migration (cross-pool migration, VDI migration).

11. Features of VMware?

Ans :

VMware is a software company that provides virtualization and cloud computing solutions to businesses of all sizes. Here are some of the key features of VMware's solutions:

1. **Server Virtualization**

   VMware's server virtualization allows organizations to consolidate multiple physical servers into a single virtualized server, which leads to significant cost savings in terms of hardware, maintenance, and energy costs.

2. **High Availability**

   VMware's solutions provide high availability and disaster recovery capabilities, ensuring that critical applications and data are always available and that downtime and data loss are minimized.

3. Scalability

VMware's solutions are designed to scale with the needs of the organization, allowing businesses to add resources as needed without disrupting existing operations.

12. Types of VMware products

Ans :

VMware is a software company that provides a wide range of virtualization and cloud computing solutions to businesses of all sizes. Here are some of the key products offered by VMware:

1. VMware vSphere

VMware vSphere is a virtualization platform that allows organizations to consolidate their physical servers and run multiple virtual machines on a single server. It provides a range of features, including high availability, disaster recovery, and automated workload management.

2. VMware ESXi

VMware ESXi is a bare-metal hypervisor that provides a platform for running virtual machines. It is optimized for virtualization and provides a lightweight, secure, and scalable platform for running multiple virtual machines.

3. VMware vSAN

VMware vSAN is a software-defined storage solution that allows organizations to pool their storage resources and use them to run virtual machines. It provides high availability, data protection, and automated management.

4. VMware NSX

VMware NSX is a network virtualization and security platform that allows organizations to create virtual networks and micro-segmentation to secure their applications and data. It provides a range of network security features, including firewalling, intrusion detection, and distributed denial-of-service (DDoS) protection.

5. VMware Horizon

VMware Horizon is a virtual desktop infrastructure (VDI) solution that allows organizations to provide remote access to their applications and desktops. It provides a secure and scalable platform for delivering virtual desktops and applications to end-users.

Choose the Correct Answers

1. Which of the following is not a type of virtualization? [d]

   (a) Server virtualization (b) Network virtualization

   (c) Storage virtualization (d) File virtualization

2. What is the purpose of virtualization in cloud computing? [d]

   (a) To improve security (b) To improve scalability

   (c) To improve reliability (d) All of the above

3. Which of the following is not a benefit of virtualization in cloud computing? [c]

   (a) Reduced hardware costs (b) Improved resource utilization

   (c) Increased complexity (d) Improved disaster recovery

4. Which of the following is a hypervisor used for server virtualization? [d]

   (a) VMware ESXi (b) Microsoft Hyper-V

   (c) Citrix XenServer (d) All of the above

5. An operating system running on a Type ______ VM is a full virtualization. [a]

   (a) 1 (b) 2

   (c) 3 (d) All of the mentioned

6. In a ______ scheme, the VM is installed as a Type 1 Hypervisor directly onto the hardware. [b]

   (a) Paravirtualization (b) Full virtualization

   (c) Emulation (d) None of the mentioned

7. What is the difference between type 1 and type 2 hypervisors? [a]

   (a) Type 1 hypervisors run on the host machine's operating system, while type 2 hypervisors run as an application on the host machine's operating system

   (b) Type 2 hypervisors are used for server virtualization, while type 1 hypervisors are used for desktop virtualization

   (c) Type 1 hypervisors provide better performance and security than type 2 hypervisors

   (d) Type 2 hypervisors provide better performance and security than type 1 hypervisors

8. Which of the following is a drawback of virtualization in cloud computing? [a]

   (a) Increased complexity (b) Reduced security

   (c) Reduced flexibility (d) Reduced scalability

9. What is the difference between full virtualization and para-virtualization? [a]

   (a) Full virtualization simulates all hardware resources, while para-virtualization simulates only some hardware resources

   (b) Full virtualization provides better performance than para-virtualization

   (c) Para-virtualization provides better isolation between virtual machines than full virtualization

   (d) Full virtualization requires hardware support, while para-virtualization does not

10. What is the purpose of a virtual machine monitor (VMM)? [a]

    (a) To manage virtual machines (b) To manage physical machines

    (c) To manage network resources (d) To manage storage resources

Fill in the Blanks

1. _________ s performed right on top of the bare hardware
2. _________ virtualization creates isolated containers on a single physical server and the OS instances to utilize the hard-ware and software in data centers.
3. The technology used to distribute service requests to resources is referred to as _________
4. _________ is the role of an API in virtualization in cloud computing.
5. _________ is used by Amazon Web Services to provide Amazon Machine Instances (AMIs).
6. Another name of Hypervisor is called _________
7. _________ involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
8. _________ allows the users' OS to be remotely stored on a server in the data centre.
9. _________ which refers to the use of automated tools and processes to manage and control data center operations.
10. _________ software is now commonly used in virtualized storage and networking, cloud management services, private data centers, desktop software, etc.

Answers

1. Hardware-level virtualization
2. OS-level virtualization
3. Load balancing
4. To manage software resources
5. Xen
6. Virtual Monitor Machine
7. I/O virtualization
8. Desktop virtualization
9. Data centre automation
10. VMware

UNIT V

Cloud computing architectures over Virtualized Data Centers: Data-Center design and Interconnection networks, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms, GAE, AWS, Azure, Inter-cloud Resource Management.

5.1 Cloud Computing Architectures Over Virtualized Data Centers

5.1.1 Data Center Design And Interconnection Networks

Q1. What is data centre? Explain different types of data centres.

Ans :

Meaning

A Data Center is an operational facility that organizations use to store, process, and distribute data and applications. The design is based on a network of computing and storage resources that enable the delivery of shared applications and data.

A data center is often built with a large number of servers through a huge interconnection network. The design of large-scale data centers and small modular data centers that can be housed in a 40-ft truck container.

Types

1. **Warehouse-Scale Data-Center Design**

   A data center that is as large as a shopping mall (11 times the size of a football field) under one roof. Such a data center can house 400,000 to 1 million servers. A small data center could have 1,000 servers. The approximate monthly cost to operate a huge 400-server data center is estimated by network cost $13/Mbps; storage cost $0.4/GB; and administration costs.

   A data center is often built with a large number of servers through a huge interconnection network. The design of large-scale data centers and small modular data centers that can be housed in a 40-ft truck container.

2. **Warehouse-Scale Data-Center Design**

   A data center that is as large as a shopping mall (11 times the size of a football field) under one roof. Such a data center can house 400,000 to 1 million servers. A small data center could have 1,000 servers. The approximate monthly cost to operate a huge 400-server data center is estimated by network cost $13/Mbps; storage cost $0.4/GB; and administration costs.

3. **Data-Center Construction Requirements**

   Most data centers are built with commercially available components. An off-the-shelf server consists of a number of processor sockets, each with a multicore CPU and its internal cache hierarchy, local shared and coherent DRAM, and a number of directly attached disk drives. The DRAM and disk resources within the rack are accessible through first-level rack switches and all resources in all racks are accessible via a cluster-level switch. Consider a data center built with 2,000 servers, each with 8 GB of DRAM and four 1 TB disk drives. Each group of 40 servers is connected through a 1 Gbps link to a rack-level switch that has an additional eight 1 Gbps ports used for connecting the rack to the cluster-level switch. The bandwidth available from local disks is 200 MB/s, whereas the bandwidth from offrack disks is 25 MB/s via shared rack uplinks. The total disk storage in the cluster is almost 10 million times larger than local DRAM.

4. Cooling System of a Data-Center Room

The data-center room has raised floors for hiding cables, power lines, and cooling supplies. The cooling system is somewhat simpler than the power system. The raised floor has a steel grid resting on stanchions about 2–4 ft above the concrete floor.

The under-floor area is often used to route power cables to racks, but its primary use is to distribute cool air to the server rack. The CRAC (computer room air conditioning) unit pressurizes the raised floor plenum by blowing cold air into the plenum.

The cold air escapes from the plenum through perforated tiles that are placed in front of server racks. Racks are arranged in long aisles that alternate between cold aisles and hot aisles to avoid mixing hot and cold air. The hot air produced by the servers circulates back to the intakes of the CRAC units that cool it and then exhaust the cool air into the raised floor plenum again.

5. Data-Center Interconnection Networks

A critical core design of a data center is the interconnection network among all servers in the datacenter cluster. This network design must meet five special requirements: low latency, high bandwidth, low cost, message-passing interface (MPI) communication support, and fault tolerance.

6. Application Traffic Support

The network topology should support all MPI communication patterns. Both point-to-point and collective MPI communications must be supported. The network should have high bisection bandwidth to meet this requirement. Network Expendability The interconnection network should be expandable. With thousands or even hundreds of thousands of server nodes, the cluster network interconnection should be allowed to expand once more servers are added to the data center. The network should be designed to support load balancing and data movement among the servers. The topology of the interconnection should avoid such bottlenecks.

However, the design could be very challenging when the number of servers increases sharply. The most critical issue regarding expandability is support of modular network growth for building data-center containers One single data-center container contains hundreds of servers and is considered to be the building block of large-scale data centers.

7. Switch-centric Data-Center Design

At the time of this writing, there are two approaches to building data-center-scale networks: One is switch centric and the other is server-centric. In a switch-centric network, the switches are used to connect the server nodes. The switch-centric design does not affect the server side. No modifications to the servers are needed. The server-centric design does modify the operating system running on the servers. Special drivers are designed for relaying the traffic. Switches still have to be organized to achieve the connections. Modular Data Center in Shipping Containers A modern data center is structured as a shipyard of server clusters housed in truck-towed containers.

8. Container Data-Center Construction

The data-center module is housed in a truck-towable container. The modular container design includes the network, computer, storage, and cooling gear. One needs to increase cooling efficiency by varying the water and airflow with better airflow management. Another concern is to meet seasonal load requirements. The construction of a container-based data center may start with one system (server), then move to a rack system design, and finally to a container system. The container must be designed to be weatherproof and easy to transport. Modular datacenter construction and testing may take a few days to complete if all components are available and power and water supplies are handy. The modular data-center approach supports many cloud service applications

5.2 Architectural Design of Compute and Storage Clouds

Q2. Explain about architecture of storage clouds.

Ans :

1. Cloud Computing and Service Models

In recent days, the IT industry has moved from manufacturing to offering more services (service-oriented). As of now, 80% of the industry is ‘service-industry’. It should be realized that services are not manufactured/invented from time-to-time; they are only rented and improved as per the requirements.

Clouds aim to utilize the resources of data centers virtually over automated hardware, databases, user interfaces and apps [1].

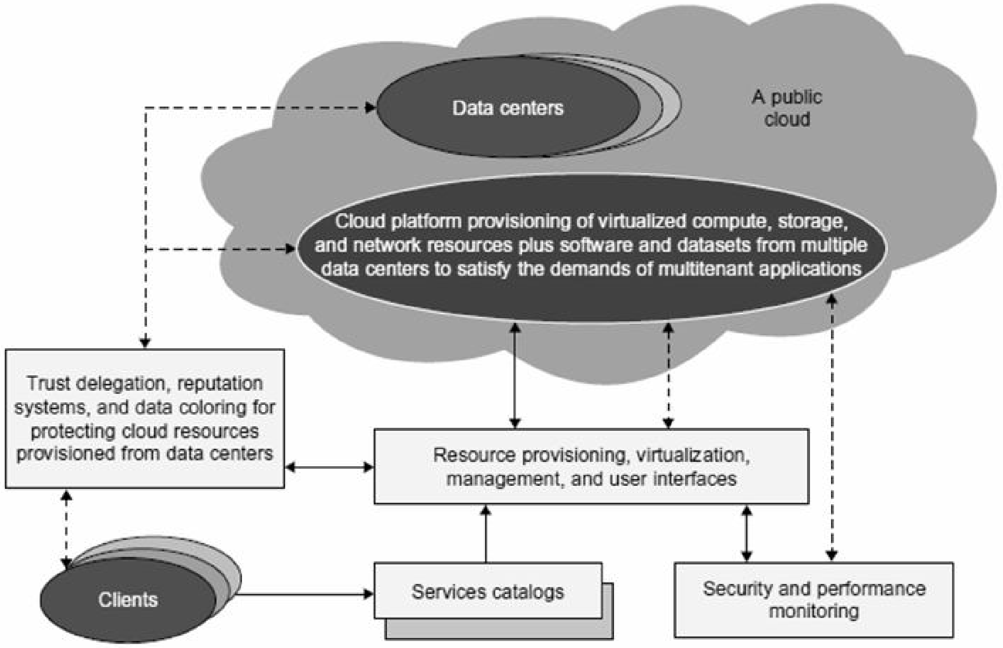
2. Public, Private and Hybrid Clouds

Cloud computing has evolved from the concepts of clusters, grids and distributed computing. Different resources (hardware, finance, time) are leveraged (use to maximum advantage) to bring out the maximum HTC. A CC model enables the users to share resources from anywhere at any time through their connected devices.

Advantages of CC: Recall that in CC, the programming is sent to data rather than the reverse, to avoid large data movement, and maximize the bandwidth utilization. CC also reduces the costs incurred by the data centers, and increases the app flexibility.

CC consists of a virtual platform with elastic resources [2] and puts together the hardware, data and software as per demand. Furthermore, the apps utilized and offered are heterogeneous.

Cloud Architecture: A generic cloud architecture can be seen Figure. The Internet Cloud is imagined as a massive cluster of servers. The different resources (space, data, and speed) of the concerned servers are allocated as per demand dynamically.

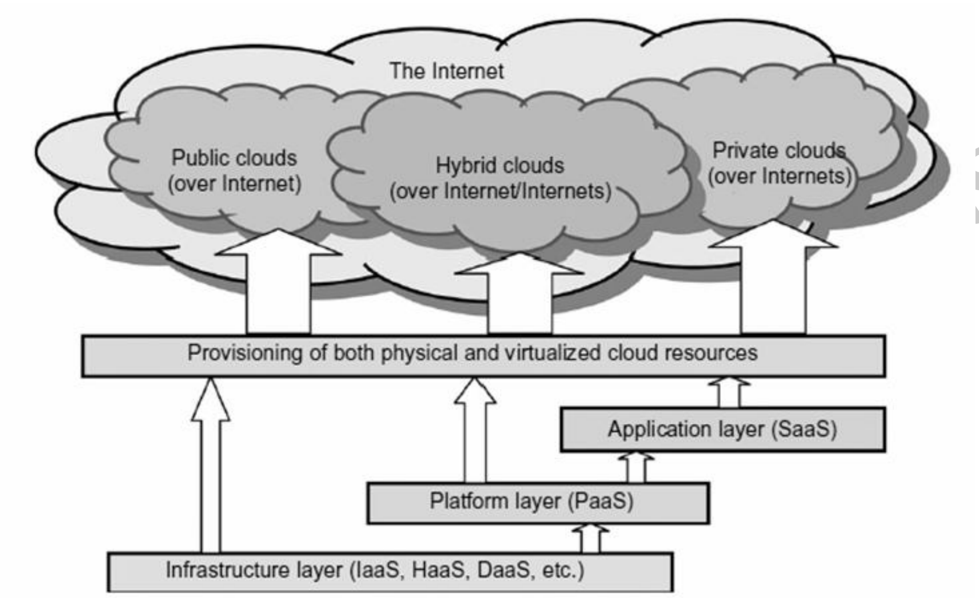

Data centers
A public cloud
Cloud platform provisioning of virtualized compute, storage, and network resources plus software and datasets from multiple data centers to satisfy the demands of multitenant applications
Trust delegation, reputation systems, and data coloring for protecting cloud resources provisioned from data centers
Resource provisioning, virtualization, management, and user interfaces
Clients
Services catalogs
Security and performance monitoring

The cloud platform demands distributed storage and different services (PaaS, IaaS and SaaS). Though the resources and services do exist and work in parallel, the user need not know about the real-work behind the screen. Any software in the cloud is a service and any service demands high amount of trust on the data retrieved from the data. Other cloud resources include storage area networks (SANs), firewalls, and security devices.

The usage and performance of granted resources are monitored and metered by special units. The software infrastructure of a cloud platform must automatically handle all the resource grants and management and note the status of each node system/server when it joins/leaves the cluster. The physical location of the data center, type of power used (general/solar/hydroelectric) and cooling required are also important points.

Typically, private clouds are easier to manage and public clouds are easier to access. In future the clouds which utilize the best resources from both the types (hybrid) are expected to grow. Finally, security becomes a critical issue in CC to grant the success of all the services.

Cloud Architecture in Layers: Cloud architecture is developed at three layers: infrastructure, platform and app. This can be noticed in Figure.


The Internet
Public clouds (over Internet)
Hybrid clouds (over Internet/Internets)
Private clouds (over Internets)
Provisioning of both physical and virtualized cloud resources
Application layer (SaaS)
Platform layer (PaaS)
Infrastructure layer (IaaS, HaaS, DaaS, etc.)

Different VZ standards are framed and utilized in all these layers to provision the resources allocated for a cloud. The services offered to public, private and hybrid through different networking supports over the Internet and intranets.

- Infrastructure layer is deployed first to support the IaaS layer. It also serves as a foundation for the PaaS layer services.
- Platform layer itself is a foundation for the SaaS services.

The layers demand resource allocation as per demand and are granted.

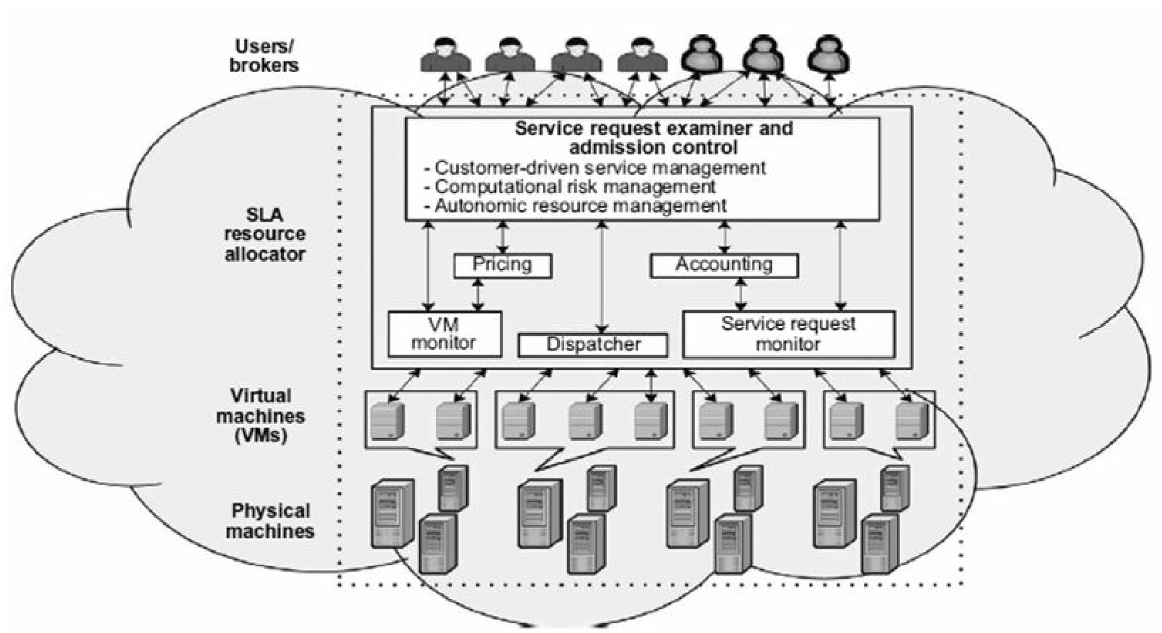
- The infrastructure layer is built with virtualized compute, storage, and network resources. Proper utilization of these resources provides the flexibility demanded by the users. Note that VZ demands automated provisioning of the resources and minimum management time.
- The platform layer is for general purpose and repeated usage of the service resources. Proper environment is provided for the development, testing, deployment and monitoring the usage of apps. Indirectly, a virtualized cloud platform acts as a 'system middleware' between the infrastructure and application layers of a cloud.

- The application layer is formed with the collection of different modules of all software that are needed for the SaaS apps. The general service apps include those of information retrieval, doc processing, and authentication services. This layer also used in large-scale by the CRMs, financial transactions, and supply chain management.

Note that all the layers are built from the scratch (bottom-up) with dependence relations in between.

Market-Oriented Cloud Architecture

This can be seen in the Figure below.


Users/ brokers
SLA resource allocator
Service request examiner and admission control
- Customer-driven service management
- Computational risk management
- Autonomic resource management
Pricing
Accounting
VM monitor
Dispatcher
Service request monitor
Virtual machines (VMs)
Physical machines

(SLA = > Service Level Agreements)

A high level architecture can be seen in the figure for supporting market oriented resource allocation in a CC environment. The entities here are users, brokers (acting on behalf of a set of similar users), and resource allocators. When a request is made, the service request examiner comes into picture and acts as an interface between the user and the data center resources.

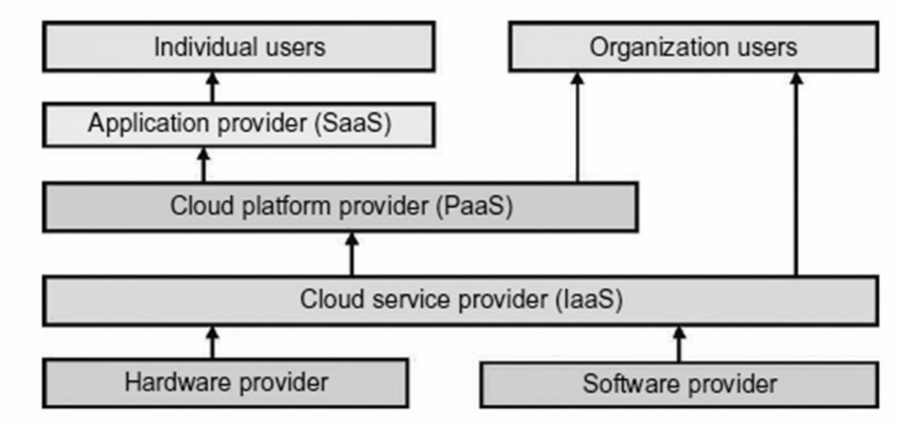
5. 3 Public Cloud Platforms

Q3. Explain about various public cloud platforms.

Ans : **(Imp.)**

Public Clouds and Service Offerings Google App Engine (GAE) Google has the world's largest search engine facilities. The company has extensive experience in massive data processing that has led to new insights into data-center design. and novel programming models that scale to incredible sizes. Google has hundreds of data centers and has installed more than 460,000 servers worldwide.

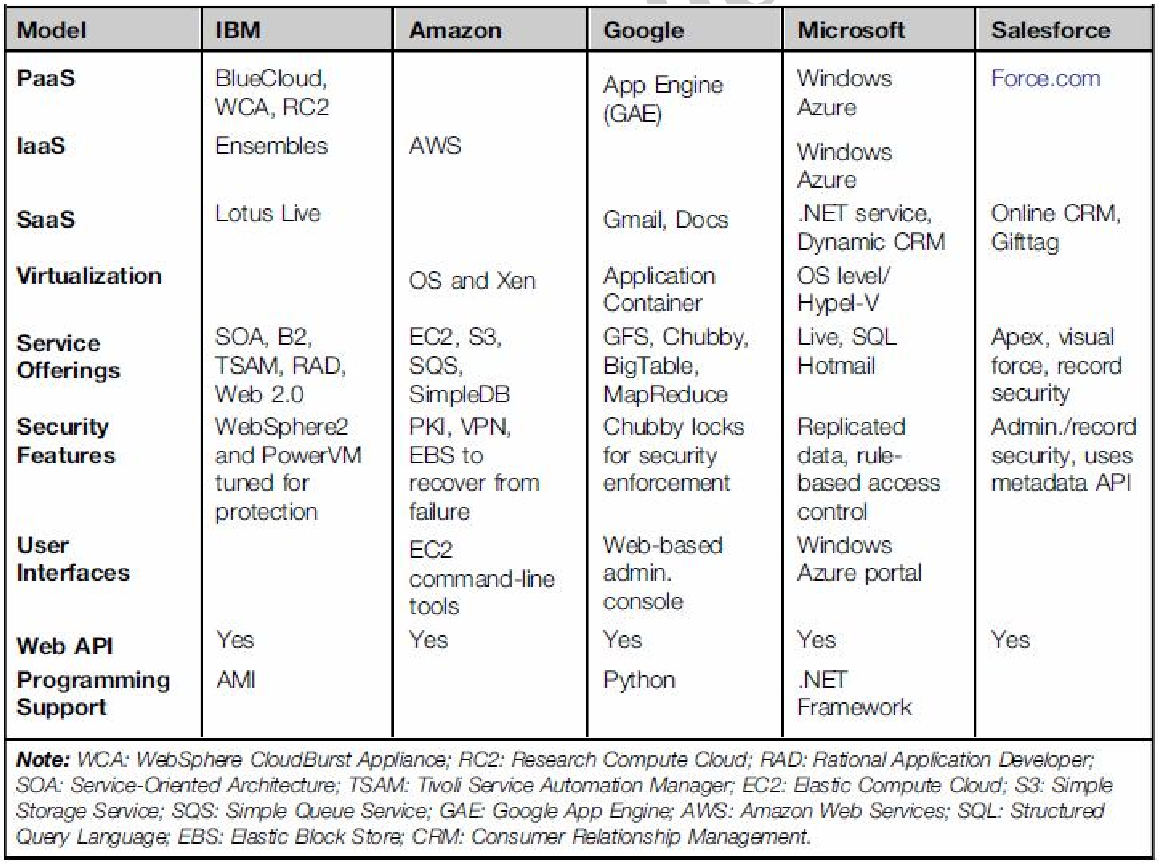
Cloud services are provided as per demand by different companies. It can be seen in Figure [1] that there are 5 levels of cloud players.


Individual users
Organization users
Application provider (SaaS)
Cloud platform provider (PaaS)
Cloud service provider (IaaS)
Hardware provider
Software provider

The app providers at the SaaS level are used mainly by the individual users. Most business organisations are serviced by IaaS and PaaS providers. IaaS provides compute, storage, and communication resources to both app providers and organisational users. The cloud environment is defined by PaaS providers. Note that PaaS provides support both IaaS services and organisational users directly.

Cloud services depend upon machine VZ, SOA, grid infrastructure management and power efficiency. The provider service charges are much lower than the cost incurred by the users when replacing damaged servers. The Table shows a summary of the profiles of the major service providers.

Table: Five Major cloud platforms and their service offering



Model	IBM	Amazon	Google	Microsoft	Salesforce
PaaS	BlueCloud, WCA, RC2		App Engine (GAE)	Windows Azure	Force.com
IaaS	Ensembles	AWS		Windows Azure	
SaaS	Lotus Live		Gmail, Docs	.NET service, Dynamic CRM	Online CRM, Gifttag
Virtualization		OS and Xen	Application Container	OS level/ Hypel-V	
Service Offerings	SOA, B2, TSAM, RAD, Web 2.0	EC2, S3, SQS, SimpleDB	GFS, Chubby, BigTable, MapReduce	Live, SQL Hotmail	Apex, visual force, record security
Security Features	WebSphere2 and PowerVM tuned for protection	PKI, VPN, EBS to recover from failure	Chubby locks for security enforcement	Replicated data, rule-based access control	Admin./record security, uses metadata API
User Interfaces		EC2 command-line tools	Web-based admin. console	Windows Azure portal	
Web API	Yes	Yes	Yes	Yes	Yes
Programming Support	AMI		Python	.NET Framework	

Note: *WCA: WebSphere CloudBurst Appliance; RC2: Research Compute Cloud; RAD: Rational Application Developer; SOA: Service-Oriented Architecture; TSAM: Tivoli Service Automation Manager; EC2: Elastic Compute Cloud; S3: Simple Storage Service; SQS: Simple Queue Service; GAE: Google App Engine; AWS: Amazon Web Services; SQL: Structured Query Language; EBS: Elastic Block Store; CRM: Consumer Relationship Management.*

PKI = > Public Key Infrastructure; VPN = > Virtual Private Network

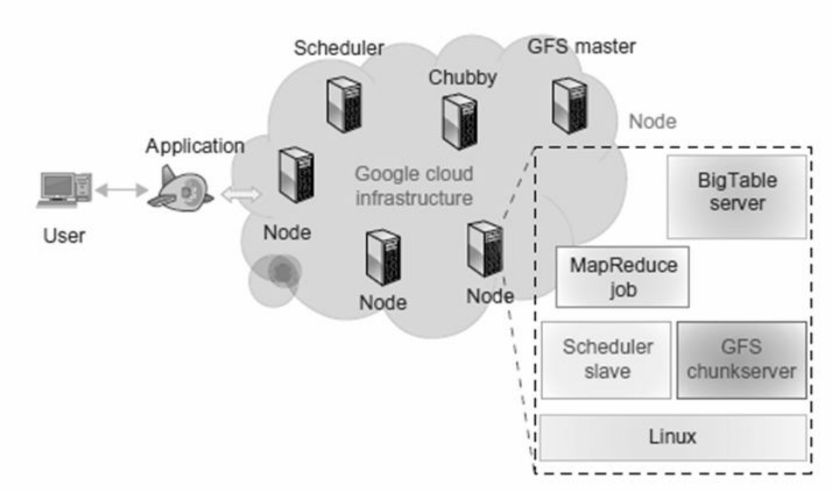
5.4 GAE

Q4. Define Google App Engine. Explain the function modules of GAE?

Ans :

Meaning

The Google platform is based on its search engine expertise and is applicable to many other areas (Ex: MapReduce). The Google Cloud Infrastructure consists of several apps like Gmail, Google Docs, and Google Earth and can support multiple no. of users simultaneously to raise the bar for HA (high availability). Other technology achievements of Google include Google File System (GFS) [like HDFS], MapReduce, BigTable, and Chubby (A Distributed Lock Service). GAE enables users to run their apps on a large number of data centers associated with Google's search engine operations. The GAE architecture can be seen in Figure below.



The building blocks of Google's CC app include GFS for storing large amounts of data, the MapReduce programming framework for developers, Chubby for distributed lock services and BigTable as a storage service for accessing structural data.

GAE runs the user program on Google's infrastructure where the user need not worry about storage or maintenance of data in the servers. It is a combination of several software components but the front end is same as ASP (Active Server Pages), J2EE and JSP.

Functional Modules of GAE

(a) Datastore offers OO, distributed and structured data storage services based on BigTable techniques. This secures data management operations.

(b) Application Runtime Environment: It is a platform for scalable web programming and execution. (Supports the languages of Java and Python)

(c) Software Development Kit: It is used for local app development and test runs of the new apps.

(d) Administration Console: Used for easy management of user app development cycles instead of physical resource management.

(e) Web Service Infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources.

The well-known GAE apps are the search engine, docs, earth and Gmail. Users linked with one app can interact and interface with other apps through the resources of GAE (synchronise and one login for all services).

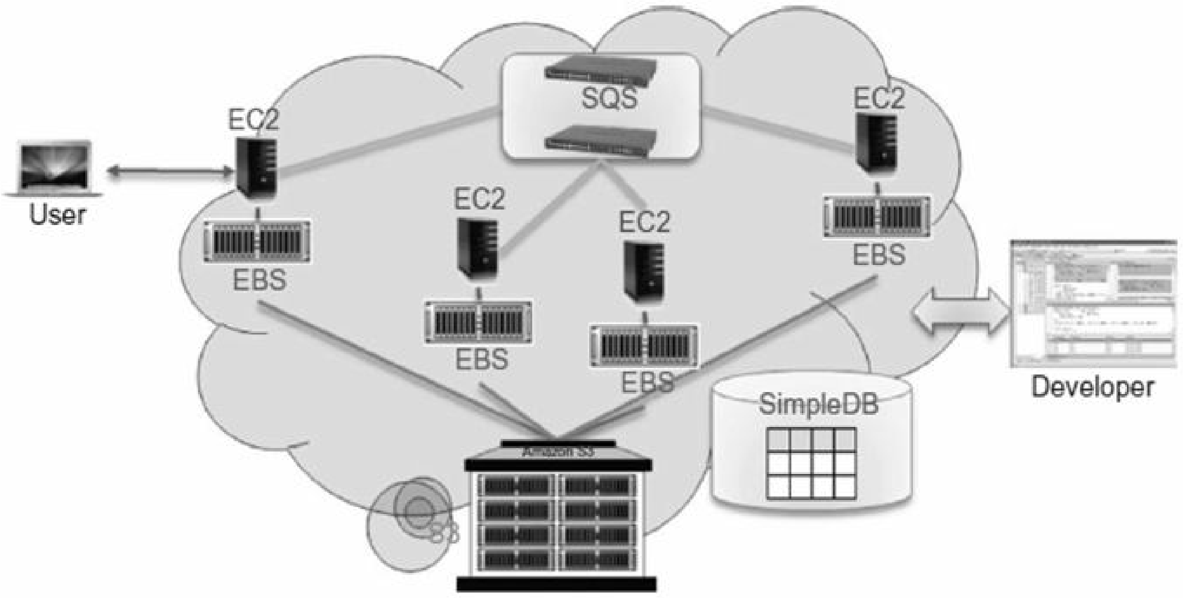
5.5 AWS

Q5. Write about Amazon Web services (AWS).

Ans : **(Imp.)**

Meaning

Amazon applies the IaaS model in providing its services. The Figure below shows the architecture of AWS:



- **EC2** provides the virtualized platforms to host the VMs where the cloud app can run.
- **S3** (Simple Storage Service) provides the OO storage service for the users.
- **EBS** (Elastic Block Service) provides the block storage interface which can be used to support traditional apps.
- **SQS** (Simple Queue Service) ensures a reliable message service between two processes.

Amazon offers a **RDS** (relational database service) with a messaging interface. The AWS offerings are given below in Table.

Table: AWS Offerings in 2011

Service Area	Service Modules and Abbreviated Names
Compute	Elastic Compute Cloud (EC2), Elastic MapReduce, Auto Scaling
Messaging	Simple Queue Service (SQS), Simple Notification Service (SNS)
Storage	Simple Storage Service (S3). Elastic Block Storage (EBS), AWS Import/ Export
Content Delivery	Amazon CtoudFront
Monitoring	Amazon CbudWatch

Support	AWS Premium Support
Database	Amazon SimpleOB, Relational Database Service (RDS)
Networking	Virtual Private Cloud (VPC) Elastic Load Balancing
Web Traffic	Alexa Web Information Service. Alexa Web Sites
E-Commerce	Fulfillment Web Service (FWS)
Payments and Billing	Flexble Payments Service (FPS), Amazon DevPay
Workforce	Amazon Mechanical Turk

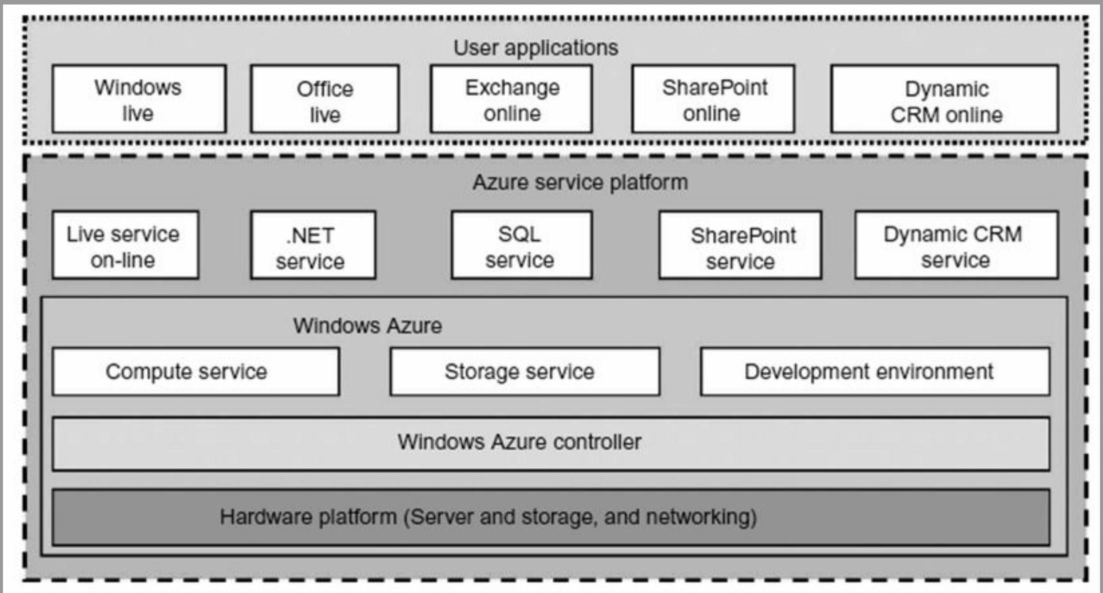
5.6 AZURE

Q6. Explain about MS AZURE.

Ans :

Meaning

The overall architecture of MS cloud platform, built on its own data centers, is shown in Figure. It is divided into 3 major component platforms as it can be seen. Apps are installed on VMs and Azure platform itself is built on Windows OS.



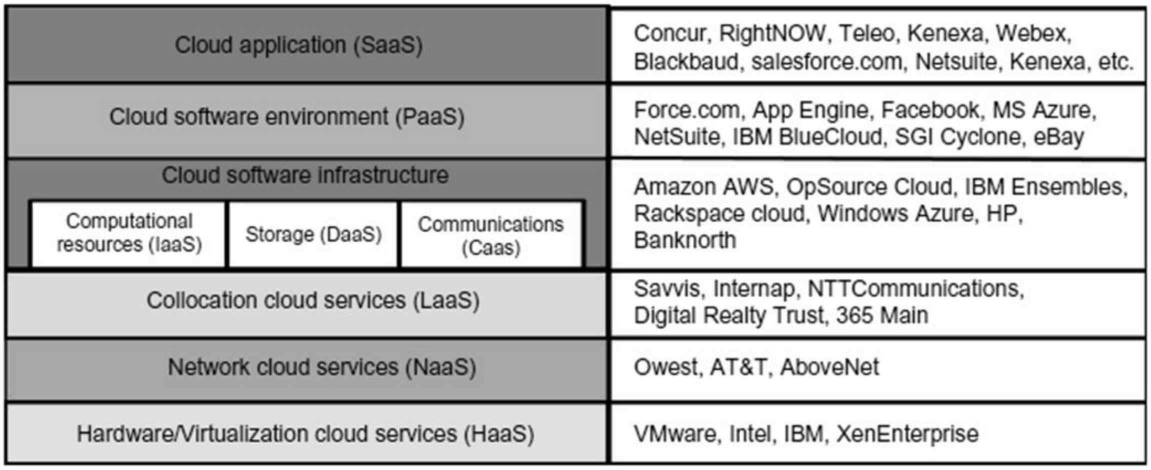
- **Live Service**: Through this, the users can apply MS live apps and data across multiple machines concurrently.
- **.NET Service**: This package supports app development on local hosts and execution on cloud machines.
- **SQL Azure**: Users can visit and utilized the relational database associated with a SQL server in the cloud.
- **SharePoint Service**: A scalable platform to develop special business apps.
- **Dynamic CRM Service**: This provides a business platform for the developers to manage the CRM apps in financing, marketing, sales and promotions.

5.7 Inter Cloud Resource Management

Q7. Explain about inter cloud resource management.

Ans :

Extended CC Services



Cloud application (SaaS)	Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.
Cloud software environment (PaaS)	Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay
Cloud software infrastructure Computational resources (IaaS) | Storage (DaaS) | Communications (Caas)	Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth
Collocation cloud services (LaaS)	Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main
Network cloud services (NaaS)	Owest, AT&T, AboveNet
Hardware/Virtualization cloud services (HaaS)	VMware, Intel, IBM, XenEnterprise

The top three service layers are SaaS, PaaS and IaaS. The bottom three layers are related to physical requirements and are as Hardware as a Service (HaaS), Network as a Service (NaaS), Location as a Service (LaaS), and Security as a Service (SaaS). Table shows that cloud players are into three classes.

Table: Cloud Differences in perspectives of providers vendors and users

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

Software Stack for CC

A software stack [7] is a group of programs that work in tandem (in order) to produce a common goal. It may also refer to any set of apps that works in a specific order toward a common goal.

Example

Like a set in maths or a cluster in DM. The system has to be designed to meet goals like HT, HA, and fault tolerance. Physical or virtual servers can be used making the platform more flexible and be able to store and utilize large amount of data.

Resource Provisioning and Platform Deployment

1. **Provisioning of Compute Resources (VMs)**

   The provisioning of resources like CPU, memory, and bandwidth are distributed among the users as per the service level agreements (SLAs) signed before the start of the work. The problem here is the ever-changing levels of requests from the user, power management and conflicts in the SLAs.

Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures. Resource provisioning also demands fast discovery of services and data in the provided infrastructure. Ex: Efficient installation of VMs, live VM migration, and fast recovery from failures. Providers like Amazon, IBM and MS-Azure use VM templates, automation of provisioning and power-efficient schemes.

2. **Resource Provisioning Methods**:

(a) Demand-Driven Resource Provisioning

This method adds or removes computing instances based on the current utilization level for the allocated resources. This method automatically allocates two processors for the user app, if the user utilizes more than 60% of time for an extended period. That is, if the resource utilization has crossed a threshold of the concerned resource, extra resources will be allocated. This methodology is implemented by Amazon in EC2.

(b) Event-Driven Resource Provisioning

This scheme adds or removes machine instances based on an event like festival season. At this time, the no. of users peaks and so does the traffic. This anticipation results in good QoS and customer satisfaction.

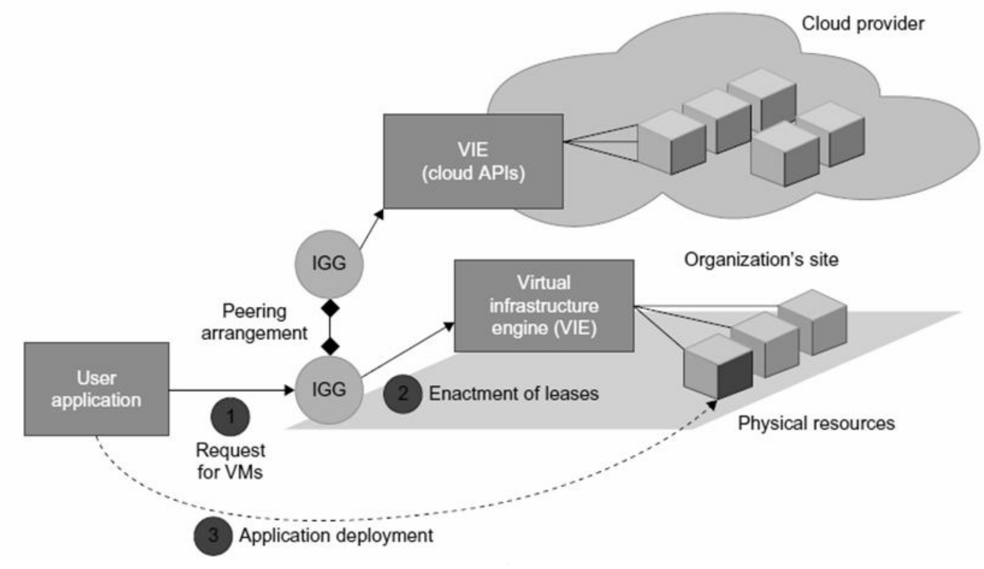
(c) Popularity-Driven Resource Provisioning

In this method, The Internet searches for popularity of certain apps and creates extra instances if the popularity has risen.

(d) Dynamic Resource Deployment

This can be implemented to achieve scalability in performance through efficient allocation of resources at every place in the grid as the situation demands. To achieve this, we need an inter-grid gateway (IGG) between different grids that allocates the resources from a local cluster to deploy apps by requesting the VMs, enacting (endorse) the leases, and deploying the VMs as per requests.

The Inter-Grid provides and allocates a distributed virtual environment (DVE). It is a virtual cluster of VMs that runs in isolation from other virtual clusters. This process is carried out by a component called DVE manager. Received massages are handled in parallel in a thread pool. All these methodologies are depicted in Figure.



3. Provisioning of Storage Resources

The data in CC is stored in the clusters of the cloud provider and can be accessed anywhere in the world. Ex: email. For data storage, distributed file system, tree structure file system, and others can be used. Ex: GFS, HDFS, MS-Cosmos. This method provides a convenient coding platform for the developers. The storage methodologies and their features can be found in Table.

Table: Storage services in Three cloud computing systems

S.No.	Storage System	Features
1.	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.
2.	HDFS: Hadoop Distributed File System	The open sou ee clone of GFS. Written in Java. The programming interfaces are similar to POSIX but not identical.
3.	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.

POSIX = > Portable OS Interface

EBS = > Elastic Block Storage

EC2 = > Elastic Compute Cloud

S3 = > Amazon Simple Storage Service

Short Question and Answers

1. What is data cetre?

Ans :

A Data Center is an operational facility that organizations use to store, process, and distribute data and applications. The design is based on a network of computing and storage resources that enable the delivery of shared applications and data.

A data center is often built with a large number of servers through a huge interconnection network. The design of large-scale data centers and small modular data centers that can be housed in a 40-ft truck container.

2. Types of data centres

Ans :

i) Warehouse-Scale Data-Center Design

A data center that is as large as a shopping mall (11 times the size of a football field) under one roof. Such a data center can house 400,000 to 1 million servers. A small data center could have 1,000 servers. The approximate monthly cost to operate a huge 400-server data center is estimated by network cost $13/Mbps; storage cost $0.4/GB; and administration costs.

A data center is often built with a large number of servers through a huge interconnection network. the design of large-scale data centers and small modular data centers that can be housed in a 40-ft truck container.

ii) Warehouse-Scale Data-Center Design

A data center that is as large as a shopping mall (11 times the size of a football field) under one roof. Such a data center can house 400,000 to 1 million servers. A small data center could have 1,000 servers. The approximate monthly cost to operate a huge 400-server data center is estimated by network cost $13/Mbps; storage cost $0.4/GB; and administration costs.

iii) Data-Center Construction Requirements

Most data centers are built with commercially available components. An off-the-shelf server consists of a number of processor sockets, each with a multicore CPU and its internal cache hierarchy, local shared and coherent DRAM, and a number of directly attached disk drives. The DRAM and disk resources within the rack are accessible through first-level rack switches and all resources in all racks are accessible via a cluster-level switch. Consider a data center built with 2,000 servers, each with 8 GB of DRAM and four 1 TB disk drives. Each group of 40 servers is connected through a 1 Gbps link to a rack-level switch that has an additional eight 1 Gbps ports used for connecting the rack to the cluster-level switch. the bandwidth available from local disks is 200 MB/s, whereas the bandwidth from offrack disks is 25 MB/s via shared rack uplinks. The total disk storage in the cluster is almost 10 million times larger than local DRAM.

3. Public cloud platforms.

Ans : **(Imp.)**

Public Clouds and Service Offerings Google App Engine (GAE) Google has the world's largest search engine facilities. The company has extensive experience in massive data processing that has led to new insights into data-center design. and novel programming models that scale to incredible sizes. Google has hundreds of data centers and has installed more than 460,000 servers worldwide.

4. Define Google App Engine.

Ans :

The Google platform is based on its search engine expertise and is applicable to many other areas (Ex: MapReduce). The Google Cloud Infrastructure consists of several apps like Gmail, Google Docs, and Google Earth and can support multiple no. of users simultaneously to raise the bar for HA (high availability). Other technology achievements of Google include Google File System (GFS) [like HDFS], MapReduce, BigTable, and Chubby (A Distributed Lock Service). GAE enables users to run their apps on a large number of data centers associated with Google's search engine operations.

5. Functional Modules of GAE

Ans :

(a) Datastore offers OO, distributed and structured data storage services based on BigTable techniques. This secures data management operations.

(b) Application Runtime Environment: It is a platform for scalable web programming and execution. (Supports the languages of Java and Python)

(c) Software Development Kit: It is used for local app development and test runs of the new apps.

(d) Administration Console: Used for easy management of user app development cycles instead of physical resource management.

(e) Web Service Infrastructure provides special interfaces to guarantee flexible use and management of storage and network resources.

6. Amazon Web services.

Ans :

Amazon applies the IaaS model in providing its services.

- **EC2** provides the virtualized platforms to host the VMs where the cloud app can run.
- **S3** (Simple Storage Service) provides the OO storage service for the users.
- **EBS** (Elastic Block Service) provides the block storage interface which can be used to support traditional apps.
- **SQS** (Simple Queue Service) ensures a reliable message service between two processes.

7. AZURE.

Ans :

- **Live Service**: Through this, the users can apply MS live apps and data across multiple machines concurrently.
- **.NET Service**: This package supports app development on local hosts and execution on cloud machines.
- **SQL Azure**: Users can visit and utilized the relational database associated with a SQL server in the cloud.
- **SharePoint Service**: A scalable platform to develop special business apps.
- **Dynamic CRM Service**: This provides a business platform for the developers to manage the CRM apps in financing, marketing, sales and promotions.

8. Resource Provisioning Methods

Ans :

(a) Demand-Driven Resource Provisioning

This method adds or removes computing instances based on the current utilization level for the allocated resources. This method automatically allocates two processors for the user app, if the user utilizes more than 60% of time for an extended period. That is, if the resource utilization has crossed a threshold of the concerned resource, extra resources will be allocated. This methodology is implemented by Amazon in EC2.

(b) Event-Driven Resource Provisioning

This scheme adds or removes machine instances based on an event like festival season. At this time, the no. of users peaks and so does the traffic. This anticipation results in good QoS and customer satisfaction.

(c) Popularity-Driven Resource Provisioning

In this method, The Internet searches for popularity of certain apps and creates extra instances if the popularity has risen.

(d) Dynamic Resource Deployment

This can be implemented to achieve scalability in performance through efficient allocation of resources at every place in the grid as the situation demands. To achieve this, we need an inter-grid gateway (IGG) between different grids that allocates the resources from a local cluster to deploy apps by requesting the VMs, enacting (endorse) the leases, and deploying the VMs as per requests.

Choose the Correct Answers

1. What is the purpose of a cloud data center? [a]
   (a) To provide on-demand access to computing resources
   (b) To store data backups in the cloud
   (c) To provide internet connectivity to cloud users
   (d) To provide hardware support for cloud applications

2. Which of the following is a benefit of using a cloud data center? [a]
   (a) Reduced hardware costs (b) Improved physical security
   (c) Increased latency (d) Reduced network bandwidth

3. What is the difference between a public cloud data center and a private cloud data center? [b]
   (a) A public cloud data center is owned by a single organization, while a private cloud data center is shared by multiple organizations.
   (b) A public cloud data center is accessible to the public, while a private cloud data center is accessible only to authorized users.
   (c) A public cloud data center is located on-premises, while a private cloud data center is located off-premises.
   (d) There is no difference between a public cloud data center and a private cloud data center.

4. Which of the following is a challenge associated with cloud data centers? [a]
   (a) Maintaining physical security (b) Limited scalability
   (c) Low bandwidth (d) High latency

5. Which of the following is an example of a public cloud platform? [a]
   (a) Microsoft Azure (b) VMware vSphere
   (c) Oracle Cloud Infrastructure (d) OpenStack

6. Which of the following is a benefit of using GAE? [a]
   (a) Reduced hardware costs (b) Improved physical security
   (c) Increased control over infrastructure (d) Reduced scalability

7. Which of the following is not a part of AWS? [c]
   (a) EC2 (b) S3
   (c) Azure (d) Lambda

8. Which service is used to store and retrieve files in AWS? [b]
   (a) EC2 (b) S3
   (c) RDS (d) Lambda

9. What is Azure? [b]
   (a) An operating system (b) A cloud computing platform
   (c) A database management system (d) A programming language

10. Which of the following is a cloud resource management tool provided by Google Cloud Platform? [c]
   (a) AWS Auto Scaling (b) Azure Monitor
   (c) Google Kubernetes Engine (d) None of the above

Fill in the Blanks

1. A physical facility that houses servers and other computing infrastructure is called ________
2. Cloud computing has evolved from the concepts ________, ________, and ________
3. The ________ layer is for general purpose and repeated usage of the service resources
4. Public Clouds and Service Offerings ________ Google has the world's largest search engine facilities
5. The ________ Google platform is designed based on its search engine expertise and is applicable to many other areas
6. The ________ method adds or removes computing instances based on the current utilization level for the allocated resources
7. The ________ scheme adds or removes machine instances based on an event like festival season
8. The ________ can be implemented to achieve scalability in performance through efficient allocation of resources at every place in the grid as the situation demands.
9. The ________ service is used to create and manage databases in AWS
10. The process of managing resources in a cloud environment is called ________

Answers

1. Cloud data center
2. Of clusters, grids and distributed computing
3. Platform
4. Google App Engine (GAE)
5. GAE.
6. Demand-Driven Resource Provisioning
7. Event-Driven Resource Provisioning
8. Dynamic Resource Deployment
9. RDS
10. Cloud resource management

FACULTY OF INFORMATICS

BCA II Year IV Semester (CBCS) Examination

Model Paper - I

DISTRIBUTED AND CLOUD COMPUTING

Time : 3 Hours] **[Max. Marks : 70**

Note : Answer all questions from Part - A, & any five questions from Part - B Choosing one questions from each unit.

PART - A (10 × 2 = 20 Marks)

ANSWERS

1. (a) Trends in distributed systems. **(Unit-I, SQA-2)**

(b) WWW **(Unit-I, SQA-6)**

(c) MPI **(Unit-II, SQA-3)**

(d) What is an overlay network. **(Unit-II, SQA-1)**

(e) Platform as a Service (PaaS) **(Unit-III, SQA-10)**

(f) Define Cloud Computing **(Unit-III, SQA-1)**

(g) Memory Virtualization **(Unit-IV, SQA-5)**

(h) Features of VMware? **(Unit-IV, SQA-11)**

(i) What is data cetre? Explain different types. **(Unit-V, SQA-1)**

(j) Resource Provisioning Methods **(Unit-V, SQA-8)**

PART - B (5 × 10 = 50 Marks)

UNIT - I

2. (a) What is distributed system? Write various examples of distributed systems. **(Unit-I, Q.No. 1)**

(b) Explain about Resource Sharing in Distributed System **(Unit-I, Q.No. 4)**

OR

3. (a) What is the process of synchronization in IPC? Explain. **(Unit-I, Q.No. 13)**

(b) Explain the Methods in Interprocess Communication. **(Unit-I, Q.No. 11)**

UNIT - II

4. What is an overlay network? Explain its applications. **(Unit-II, Q.No. 1)**

OR

5. Explain briefly about RMI state in architecture and working mechanism. **(Unit-II, Q.No. 9)**

UNIT - III

6. Explain the basic principles of cloud compuitng. **(Unit-III, Q.No. 8)**

OR

7. Explain about system models for distributed and cloud computing. **(Unit-III, Q.No. 4)**

UNIT - IV

8. (a) Explain about Memory Virtualization. **(Unit-IV, Q.No. 5)**

   (b) Explain about various virtual clusters. **(Unit-IV, Q.No. 7)**

OR

9. (a) Explain XEN API with architecture diagram. **(Unit-IV, Q.No. 11)**

   (b) Write about some other types of virtualization techniques. **(Unit-IV, Q.No. 9)**

UNIT - V

10. What is data centre? Explain different types of data centres. **(Unit-V, Q.No. 1)**

OR

11. Explain about intes cloud resource management. **(Unit-V, Q.No. 7)**

FACULTY OF INFORMATICS

BCA II Year IV Semester (CBCS) Examination

Model Paper - II

DISTRIBUTED AND CLOUD COMPUTING

Time : 3 Hours] **[Max. Marks : 70**

Note : Answer all questions from Part - A, & any five questions from Part - B Choosing one questions from each unit.

PART - A (10 × 2 = 20 Marks)

ANSWERS

1. (a) Inter process communication **(Unit-I, SQA-7)**

(b) Challenges in distributed computing. **(Unit-I, SQA-4)**

(c) Components of RMI Architecture **(Unit-II, SQA-8)**

(d) Publish-Subscribe **(Unit-II, SQA-10)**

(e) Roots of cloud computing **(Unit-III, SQA-3)**

(f) Features of a Cloud. **(Unit-III, SQA-5)**

(g) Desktop Virtualization **(Unit-IV, SQA-8)**

(h) Explain different types of virtualization. **(Unit-IV, SQA-4)**

(i) Public cloud platforms. **(Unit-V, SQA-3)**

(j) Amazon Web services. **(Unit-V, SQA-6)**

PART - B (5 × 10 = 50 Marks)

UNIT - I

2. (a) Explain the web challenges in distributed computing. **(Unit-I, Q.No. 6)**

(b) Explain about various types of distributed systems. **(Unit-I, Q.No. 2)**

OR

3. (a) Explain about External data Representation in Distributed Systems. **(Unit-I, Q.No. 15)**

(b) What are the emerging trends in distributed systems? Explain. **(Unit-I, Q.No. 3)**

UNIT - II

4. Explain the concept of Publish-Subscribe? **(Unit-II, Q.No. 15)**

OR

5. State the advantages and disadvantages of RPC? **(Unit-I, Q.No. 8)**

UNIT - III

6. Write about the challenges of cloud computing. **(Unit-III, Q.No. 3)**

OR

7. Explain different types of clouds. **(Unit-III, Q.No. 7)**

UNIT - IV

8. (a) Explain about the levels of virtualization. **(Unit-IV, Q.No. 1)**

   (b) Explain about resource management in cloud computing **(Unit-IV, Q.No. 8)**

OR

9. (a) Write about I/O virtualization. **(Unit-IV, Q.No. 6)**

   (b) Explain the features of VMware? **(Unit-IV, Q.No. 15)**

UNIT - V

10. Explain about various public cloud platforms. **(Unit-V, Q.No. 3)**

OR

11. Write about Amazon Web services (AWS). **(Unit-V, Q.No. 5)**

FACULTY OF INFORMATICS

BCA II Year IV Semester (CBCS) Examination

Model Paper - III

DISTRIBUTED AND CLOUD COMPUTING

Time : 3 Hours] **[Max. Marks : 70**

Note : Answer all questions from Part - A, & any five questions from Part - B Choosing one questions from each unit.

PART - A (10 × 2 = 20 Marks)

ANSWERS

1. (a) Characteristics of Inter-process Communication **(Unit-I, SQA-8)**

(b) Multicast Communication **(Unit-I, SQA-10)**

(c) Batch-mode RPC **(Unit-II, SQA-6)**

(d) Callback RPC **(Unit-II, SQA-5)**

(e) Principle of Cloud Computing **(Unit-III, SQA-7)**

(f) Cloud Computing Layers **(Unit-III, SQA-5)**

(g) Desktop Virtualization **(Unit-IV, SQA-8)**

(h) Network Virtualization **(Unit-IV, SQA-7)**

(i) AZURE **(Unit-V, SQA-7)**

(j) Functional Modules of GAE **(Unit-V, SQA-5)**

PART - B (5 × 10 = 50 Marks)

UNIT - I

2. (a) State the Characteristics of Inter-process Communication **(Unit-I, Q.No. 10)**

(b) Explain about distributed system model. **(Unit-I, Q.No. 8)**

OR

3. (a) Explain about the API for the Internet protocols. **(Unit-I, Q.No. 14)**

(b) Explain about inter process communication is distributed systems. **(Unit-I, Q.No. 9)**

UNIT - II

4. Explain about remote method invocation. **(Unit-II, Q.No. 3)**

OR

5. Explain, the evolution of components from objects. **(Unit-I, Q.No. 21)**

UNIT - III

6. Explain the basic principles of cloud compuitng. **(Unit-III, Q.No. 9)**

OR

7. Explain about different cloud service models. **(Unit-III, Q.No. 11)**

UNIT - IV

8. (a) Explain about virtualization structures tools and mechanisms. **(Unit-IV, Q.No. 2)**

   (b) Why CPU Virtualization is Important? **(Unit-IV, Q.No. 4)**

OR

9. (a) Explain the concept of data centre automation. **(Unit-IV, Q.No. 10)**

   (b) What are different types of VMware products? Explain. **(Unit-IV, Q.No. 14)**

UNIT - V

10. Define Google App Engine. Explain the function modules of GAE? **(Unit-V, Q.No. 4)**

OR

11. Explain about architecture of storage clouds. **(Unit-V, Q.No. 2)**