UNIT I

1.**What is meant distributed system**?

1. We define a distributed system as a collection of autonomous computers linked by a network, with software designed to produce an integrated computing facility.

2. A system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing.

3. A collection of two or more independent computers which coordinate their processing through the exchange of synchronous or asynchronous message passing.

4. A collection of independent computers that appear to the users of the system as a single computers.

2. **What are the significance of distributed system?**

a. Concurrency of computers.
b. No global clock.
c. Independent failures.

3. **Why we do you need distributed system?**

a. **Functional distribution**: Computers have different functional capabilities (i.e., sharing of resources with specific functionalities).
b. **Load distribution/balancing**: Assign tasks to processors such that the overall system performance is optimized.
c. **Replication of processing power**: Independent processors working on the same task.
d. Distributed system consisting of collections of microcomputers may have processing powers that no supercomputer will ever achieve.
e. **Physical separation**: Systems that rely on the fact that computers are physically separated (e.g., to satisfy reliability requirements).
f. **Economics**: Collections of microprocessors offer a better price/performance ratio than large mainframes.mainframes:10 times faster, 1000 times as expensive.

4. **Examples of distributed system?**

a. Internet
b. Intranet
c. Mobile and ubiquitous computing.

5. **What is meant by location aware computing?**

Mobile computing is the performance of computing tasks while the users are on the move and away from their residence intranet but still provided with access to resources via the devices they carry with them. They can continue to access the intranet, they can continue to access resources in their home intranet, and there is increasing provision for users to utilize resources such as printers that are conveniently nearby as they move around. This is known as location aware computing.

6. **What are the two type of resource sharing?**

a. Hardware sharing: Printers. plotters and large disks and other peripherals are shared to reduce costs.

b. Data sharing is important in many applications:

1. Software developers working in a team need to access each other's code and share the same development tools.

2. Many commercial applications enable users to access shared data objects in a single active database.

3. The rapidly growing area of group-ware tools enables users to cooperate with in a network.

7. **List the importance of data sharing?**

- Software developers working in a team need to access each other's code and share the same development tools.
- Many commercial applications enable users to access shared data objects in a single active database.
- The rapidly growing area of group- ware tools enables users to cooperate with in a network.

8. Write the technological components of web?

- HTML
- HTTP-request-reply protocol
- URL's

9. List the distributed systems challenges?

a. Heterogeneity: standards and protocols; middleware; virtual machine;
b. Openness: publication of services; notification of interfaces;
c: Security: firewalls; encryption;
d. Scalability: replication; caching; multiple servers;
e. Failure Handling. failure tolerance; recover/roll-back; redundancy;
f. Concurrency. concurrency control to ensure data consistency.
g. Transparency. Middleware; location transparent naming; anonymity

10. What are the three components of security?

Security for information resources has three components:

- **Confidentiality:** production against disclosure to unauthorized individuals.
- **Integrity:** production against or corruption.
- **Availability:** production against interference with the means to access the resources.

11. What is the use of firewall?

A firewall can be used to form a barrier around an intranet to protect it from outside users but does not deal with ensuring the appropriate use of resources by users within the intranet.

12. What are the security challenges? List them.

a. **Denial of service attacks:** Another security problem is that the user may wish to disrupt a service for some reason. This can be achieved by bombarding the service with such a large number of pointless requests that the serious users are unable to use it. This is called a denial of service attack and there are many on well known web services.

b. **Security of mobile code:** Mobile codes needed to be handled with care. PC users sometimes send executable files as email attachments to be run by the recipient, but a recipient will not be able to run it.

13. List the challenges to be considered for designing scalable distributed system?

- Controlling the cost of physical resources
- Controlling the performance loss
- Preventing software resources running out
- Avoiding performance bottlenecks.

14. What are the types of transparencies?

a. **Access transparency:** enables local and remote resources to be accessed using identical operations.

b. **Location transparency:** enables resources to be accessed without knowledge of their location.

c. **Concurrency transparency:** enables several processes to operate concurrently using shared resources without interference between them.

d. **Replication transparency:** enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers. `

e. **Failure transparency:** enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

f. **Mobility transparency:** allows the movement of resources and clients within a system without affecting the operation of users or programs.

g. **Performance transparency:** allows the system to be reconfigures to implement performance as loads vary.

h. **Scaling transparency:** allows the system and applications to expand in scale without change to the system structure or the application algorithms.

i. Access and location transparency together provide network transparency.

15. What are the failures detected in DS?

Masking failures: Some detected failures can be hidden or made less severe. Examples of hiding failures:

1. Messages can be retransmitted when they fail to arrive

2. File data can be written to a pair of disks that if one is corrupted, the other may still be correct.

Tolerating failures: Most of the services in the Internet do exhibit failures. It would not be practical for them to detect and hide all the failures occur in such network. Their clients are designed to tolerate failures, which generally involve the users in that.

Recovery from failures: involves the design of software so that the state permanent data can be rolled back after a server has crashed.

16. List the key design goals of DS?

a. Performance
b. Reliability
c. Scalability
d. Consistency
e. Security

17. List the technical design goals of DS?

a. Naming
b. Communication
c. Software structure
d. Workload allocation
e. Maintenance of consistency.

18 .What is the use of multicast?

a. **Locating an object:** A process multicasts a message containing a name of a resource to a group of server processes. Only the process that holds the resource responds to the message.

b. **Fault tolerance:** A process multicasts its request to a group of identical server processes. The group of servers can continue to provide their service even if one of its members fails.

c. **Multiple update:** Used for example in video conferencing with multiple participants.

19. Write the models used in workload allocation?

The following four models for workload allocation are presented here.

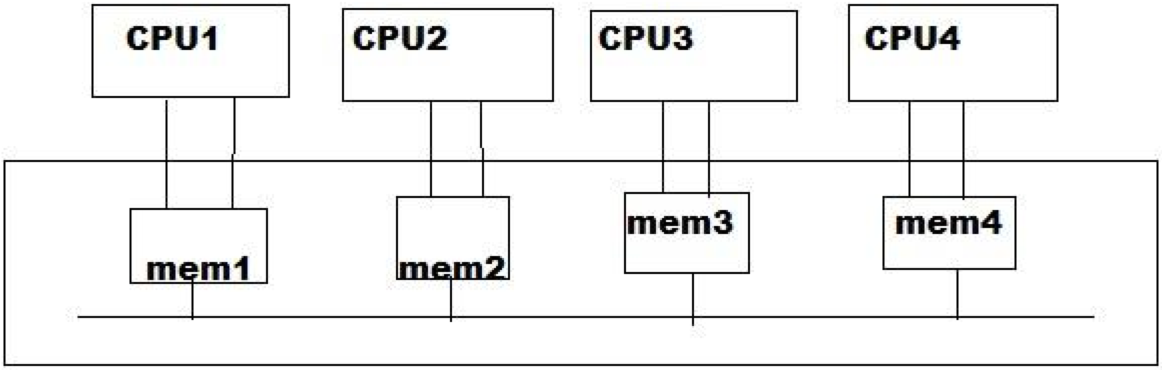
- Processor pool model
- Shared memory multiprocessors
- Parallel virtual machines
- Distributed shared memory.

20.What is meant by PVM?

PVM is an integrated set of software tools and libraries that emulates a general-purpose, flexible heterogeneous concurrent computing framework on interconnected computers of varied architecture. The overall objective of the PVM system is to enable a collection of computers to be used cooperatively for concurrent or parallel computation.

21.What is meant by DSM?

Distributed shared memory (DSM): DSM provides a global shared address space across the different machines on a cluster. The shared address space distinguishes it from packages such as PVM that provide a message passing interface between machines. There is a growing consensus in the parallel computing community that a shared memory interface is more desirable from the application programmer's viewpoint, allowing him to focus on algorithmic development rather than on managing communication.



22.List the types of consistencies in DS?

- Update consistency
- Replication consistency
- Cache consistency
- Failure consistency
- Clock consistency

23.List the user requirements used in design of DS?

- **Functionality:** what should the system do for the users.
- **Quality of service:** issue containing performance, reliability and security.
- **Reconfigurability:** the need to accommodate changes without causing disruption of the actual service.

24.List the main types of architectural model?

a. Software architecture.

b. System architecture.

- Client server model
- Services provided by multiple servers.
- Proxy servers and cache.
- Peer processes.

25. Enumerate the factors to be considered for variations in client server model?

The factors considered for several variations on the client server model.

- The use of multiple servers and caches to increase performance and flexibility.
- The use of mobile code and mobile agents.
- User’s need for low cost components with limited hardware resources that are simple to manage.
- The requirements to add remove mobile devices in a convenient manner.

26.What is meant by thin clients?

- This refers to software layer that supports a window based user interface on a computer that is local to the user while executing application programs on a remote computer.
- It has the low management and hardware costs, but it runs the application code of user’s computer in the computer server, which is the powerful computer that has the capacity to run large numbers of applications simultaneously. It can be multiprocessor or cluster computer running a multiprocessor version of OS such as UNIX or windows NT.

27.What is meant by x-11 window system?

a) The X-11 window system is a process that manages the display and interactive input devices (keyboard, mouse) of the computer on which it runs. It provides an extensive library of procedures (the X-11 protocol) for displaying and modifying graphical objects in windows as well as the creation and manipulation of windows.

b) The X-11 system is referred to as a window server process. The clients of the X-11 server are the application programs that the user is currently interacting with.

c) The client programs communicate with the server by invoking operation in the X-

11 protocol, these include operations to draw text and graphical objects in windows.

28.Enumerate the key features of spontaneous networking?

a. **Easy connection to a local network:** Wireless links avoid the need for pre-installed cabling and avoid the inconvenience and reliability issues surrounding plugs and sockets.
b. **Easy integration with local services:** Devices are able to find themselves inserted into existing networks of devices discover automatically what services are provided there, with no special configuration actions by the user.

29.List the design issues to be considered for spontaneous networking?

a. **Limited connectivity:** Users are not always connected as they move around. They are irregularly disconnected from wireless network as they travel through tunnels by train. They may also be totally disconnected for longer periods of time in regions where wireless connectivity ceases or it is too expensive to remain connected.
b. **Security and privacy:** Many vulnerable security issues arises due to the attempt of wireless connections in unsupervised way. Some systems track the physical locations of users as they move around and this may threaten the user’s privacy. This facility enables users to access their home intranet while on the move may expose data that is supposed to remain behind the intranet firewall or it may open up the intranet to attacks from outside.
c. **Discovery services:** Spontaneous networking requires client processes running on portable devices and other appliances to access services on the networks to which they are connected. Here the clients discover what services are available in the network to which they are connected and to investigate their properties. The purpose of a discovery service is to accept and store details of services that are available on the network and to respond to queries from clients about them.

30.What is the purpose solved by fundamental model?

- In general, such a fundamental model should contain only the essential ingredients that we need to consider understanding and reasoning about some aspects of a system’s behaviour. The purpose of such a model is:
- To make explicit all the relevant assumptions about the system we are modelling.
- To make generalization concerning what is possible or impossible, given those assumptions. The guarantees are our assumptions clear and explicit, we can hope to prove system properties using mathematical techniques. These properties will then hold for any system meeting our assumptions.

31.How the fundamental models are categorized?

a. Interaction
b. Failure
c. Security

32.List out the characteristics of performance of DS?

The following performance characteristics relating to latency, bandwidth and jitter.

a. **Latency:** The delay between the start of a message’s transmission from one process and the beginning of its receipt by another is referred to as latency.
b. **Bandwidth:** The bandwidth of a computer network is the total amount of information that can be transmitted over it in a given time. When large number of communication channels are using the same network, they have to share the

available bandwidth.

c. **Jitter.** Jitter is the variation in the time taken to deliver a serious of messages. Jitter is relevant to multimedia data. For example, if consecutive samples of audio data are played with differing time intervals, the sound will be badly distorted.

33.What is synchronous DS?

1) The time to execute each step of a process has known lower and upper bounds.
2) Each message transmitted over a channel is received within a known bounded time.
3) Each process has a local clock whose drift rate from real time has a known bound.
4) It is possible to suggest likely upper and lower bounds for process execution time, message delay and clock drift rates in a distributed system, but it is difficult to arrive at realistic values and to provide guarantees of the chosen values.
5) In a synchronous system it is possible to use timeouts, for example to detect the failure of a process.

34.What is asynchronous DS?

1. Many distributed systems, such as the Intranet, qualify as asynchronous system.
2. An asynchronous distributed system is one in which there are no bounds on:

1. **Process execution speeds-**for example, one process step may take only a picoseconds and another a century; all that can be said is that each step may take an arbitrarily long time.

2. **Message transmission delays-**for example, one message from process A to process B may be delivered in negligible time and another may take several years. In other words, a message may be received after an arbitrarily long time.

3. **Clock drift rates-** again, the drift rate of a clock is arbitrary.

35. What is omission failure?

The faults classified as omission failures refer to cases when a process or communication channel fails to perform actions that it is supposed to do.

36.What is meant by arbitrary failure?

1) The term arbitrary or Byzantine failure is used to describe the worst possible failure semantics, in which any type of error may occur. For example, a process may set wrong values in its data items, or it may return a wrong value in response to an invocation.
2) An arbitrary failure of a process is one in which in arbitrarily omits intended processing steps to takes unintended processing steps. Arbitrary failures in processes cannot be detected by seeing whether the process responds to invocations, because it might arbitrarily omit to reply.

37.List out the characteristics of networks hidden by stream abstraction?

a) **Message sizes:** The application can choose how much data it writes to a stream or reads from it. It may deal in very small or very large sets of data. The underlying implementation of a TCP stream decides how much data to collect before transmitting it as one or more IP packets.
b) **Lost messages:** The TCP protocol uses an acknowledgement scheme. As an example of a simple scheme (which is not used in TCP), the sending end keeps a record of each IP packet sent and receiving end acknowledges all the arrivals. If the sender does not receive an acknowledgement within a timeout, it retransmits the message.
c) **Flow control:** The TCP protocol attempts to match the speeds of the processes that read from and write to a stream. If the writer is too fast the reader, then it is

blocked until the reader has consumed sufficient data.

d) **Message duplication and ordering:** Message identifiers are associated with each IP packet, which enables the recipient to detect and reject duplicates, or to reorder message that do not arrive in sender order.

e) **Message destinations:** A pair of communication processes establishes a connection before they can communicate over a stream. Once a connection is established, the processes simply read from and write to the stream without needing to use Internet addresses and ports. Establishing a connection involves a connect request from client to server followed by an accept request from server to client before any communication can take place.

38.List the issues related to stream communication?

a. **Matching of data items:** Two communicating processes need to agree as to the contents of the data transmitted over a stream. For example, if one process writes an int followed by a double to a stream, then the reader at the other end must read an int followed by a double. When a pair of processes does not cooperate correctly in their use of a stream, the reading process may experience errors when interpreting the data or may block due to insufficient data in the stream.

b. **Blocking:** The data written to a stream is kept in a queue at the destination socket. When a process attempts to read data from an input channel, it will get data from the queue or it will block until data becomes available. The process that writes data to a stream may be blocked by the TCP flow-control mechanism if the socket at the other end is queuing as much data as the protocol allows.

c. **Threads:** When a server accepts a connection, it generally creates a new thread in which to communicate with the new client. The advantage of using a separate thread for each client is that the sever can block when waiting for input without delaying other clients.

d. **Failure model:** To satisfy the integrity to property of reliable communication, TCP streams use checksums to detect and reject corrupt packets and sequence numbers to detect and reject duplicate packets. For the deal with lost packets. Therefore, messages are guaranteed to be delivered ever when some of the underlying packets are lost.

39. What is marshalling and unmarshalling?

- An agreed standard for the representation of data structures and primitive values is called an external and data representation
- **Marshalling** is the process of taking a collection of data items and assembling them into a form suitable for transmission in a message.
- **Unmarshalling** is the process of disassembling them on arrival to produce an equivalent collection of data items at the destination. Thus marshalling consists of the translation of structured data items and primitives values into an external data representation. Similarly, unmarshalling consists of the generation of primitive values from their external data representation and the rebuilding of the data structures.

40.What are the approaches used in data representation?

a. **CORBA's common data representation,** which is concerned with an external representation for the structured and primitive types that can be passed as the arguments and results of remote method invocations in CORBA. It can be used by a variety of programming languages.

b. **Java's object serialization,** which is concerned with the flattering and external data representation of any single object or tree of objects that may need to be transmitted in a message or stored on a disk. It is for use only by java.

c. **XML (Extensible Markup Language),** which defines a textual format for representing structured data. It was originally intended for documents containing textual self-describing structured data-for example documents accessible on the Web- but it is now also used to represent the data sent in message exchanged by clients and servers in web services.

16 MARK QUESTIONS

1. Explain the need of Distributed systems its characteristics with example
2. Explain how resource sharing is done in the web
3. Explain the challenges to be considered in the design of DS
4. Explain the design goals to be considered for DS
5. Explain the system model of DS in details
6. Explain the system architecture of DS
7. Explain the variations to be considered in client server model
8. Explain the fundamental models considered in design of DS
9. Describe interprocess communication in details
10. Discuss in details about group or multicast communication

UNIT II

1. Draw the Middleware Architecture.

Application
RMI,RPC and events
Request reply protocol External data representation
Operating system

2. What are the benefits of programming with interface in DS?

I. As with any form of modular programming, programmers are concerned only with the abstraction offered by the service interface and need not be aware of implementation details.

II. In potentially heterogeneous distributed systems , programmers also do not need to know the programming language or underlying platform used to implementation service.

III. This approach provides natural support for software evolution in this implementation can change as long as the interface remains the same.

3. Define IDL.

Interface Definition Languages (IDLs) are designed to allow procedures implemented in different languages to invoke one another. An IDL provides a notation for defining interfaces in which each of the parameters of an operation may be described as for input or output in addition to having its type specified.

4. List the used of IDL in web services.

The concept of an IDL was initially developed for RPC systems but applies equally to RMI and also web service. Some of them are:

I. Sun XDR as an example of an IDL for RPC

II. CORBA IDL as an example of an IDL for RMI

III. The web service Description Language (WSDL),which is designed for an Internet wide RPC supporting web service.

5. What is meant by action in object model?

Action is an object oriented program is initiated by an object invoking a method in another object. An invocation can include additional information needed to carry out the method. The receiver executes the appropriate method and then returns control to the invoking objects, sometimes supplying a result .An invocation of a method can have three effects:

I. The state of the receiver may be changed.

II. A new object may be instantiated, for example, by using a constructor in java or C++.

III. Further invocation on methods in other objects may take place.

6. Define object reference.

Objects can be accessed via object reference. For example in java a variable that appears to hold an object actually holds a reference to that object. To invoke a method in an object the object reference and method name are given together with any necessary arguments. The object whose method name is invoked is sometimes called the target and sometimes the receiver. Object reference are first class values, meaning that they may be assigned to variables, passed as arguments and returned as results of methods.

7.What is meant by garbage collection?

It is necessary to provide a means of freeing the space occupied by objects when they are no longer needed. A language such as java, that can detect automatically when an object is no longer accessible recovers the space and makes it available for allocation to other objects. This process is called garbage collection; the programmer has to cope with the freeing of space allocated to objects. This can be a major source of errors.

8.List the heart of distributed object model.

Remote object references: Other objects can invoke the methods of a remote object if they have access to its remote object reference. For example a remote object reference for B must be available to A.

The notation of object reference is extended to allow any object that can receive an

RMI to have a remote object reference. A remote object reference is an identifier that can be used throughout a distributed system to refer to a particular unique remote object. Its representation which is generally different from that of a local object reference. Remote object references are analogous to local one in that:

- The remote object to receive a remote method invocation is specified by the invoker as a remote object reference.
- Remote object references may be passed as arguments and results of remote methods invocations.

Remote interfaces: Every object has a remote interface that specifies which of its methods can be invoked remotely..

The class of a remote object implements the methods of its remote interface, for example as public instance methods in java.Object in other processes can invoke only the methods that belong to its remote interface.

9.Define RMI .

Each process contains a collection of objects, some of which can receive both local and remote invocations whereas the other objects can receive only local invocations as shown in figure.

Method invocation between objects in different processes, whether in the same computer or not, are known as remote method invocations. Method invocation between objects in the same process is local method invocation. We refer to objects that can receive remote invocation as remote objects.

10. What are the main choices to be considered in design of RMI?

RMI invocation semantics

a. Retry-reply protocols, where we showed that doOperation can be implemented in different ways to provide different guarantees.

b. The main choices are:

i. Retry request message: Controls whether to retransmit the request message until either a reply is received or the server is assumed to have failed.

ii. Duplicate filtering: Controls when retransmissions are used and whether to filter out duplicate requests at the server.

Iii.Retrasmission of results: Controls whether to keep a history of result message to enable lost results to be retransmitted without re-executing the operations at the server.

11. List the choices of RPC invocation semantics.

The choices of RPC invocation semantics are defined as follows:

a **Maybe semantics:** With maybe semantics, the remote procedure call may be executed once or not at all. Maybe semantics is useful only for applications in which occasional failed calls are acceptable. Maybe semantics arises when no fault-tolerance measures are applied and suffer from the following types of failure;

i. Omission failures if the request message is lost;

ii. Crash failures when the server containing the remote operation fails.

b**. At-least-once semantics :**With at-least-once semantics ,the invoker receives either a result ,in which case the invoker knows that the procedure was executed at least once ,or an exception informing it that no result was received .At-least-once semantics can be achieved by the retransmission of request message . Which masks the omission failures of the request message. At-least-once semantics can suffer from the following types of failure:

i. crash failure when the sever containing the remote procedure fails;

ii. arbitrary failures – incase when the request message is retransmitted, the remote server may receive it and execute the procedure more than once, possible causing wrong values to be returned.

c. At-most-once semantics : With at-most-once semantics , the caller receives either a Result , in which case the caller knows that the procedure was executed exactly once, or an Exception informing if that no r using result was received, in which case the procedure will have been executed either once or not at all. At-most-once semantics can be achieved by all, At-most-once semantics can be achieved by using all of the fault-tolerance.

12. Sketch the RMI reply-request message structure.

Message Type
Request Id
Object Reference
Method Id
arguments

13. List the action of remote reference module.

The action of the remote module is as follows:

- When a remote object is to be passed as an argument or a result for the first time, the remote reference module is asked to create a remote object reference which it adds to its table.
- When a remote object reference arrives in a request or reply message, the remote reference module is asked for the corresponding local object reference, which may refer either to a proxy or to a remote object.
- This module is called by components of the RMI software when they are marshalling and unmarshalling remote object reference.

14. List the function of activator.

- processes that start server processes to host remote objects are called activators, for the following reasons.

  i) A remote object is described as active when it is available for invocation within a running process wherever it is called passive if it is not currently active but can be made active

  ii) A passive object consists of two parts:

  1. the implementation of its methods.
  2. its state in the marshaled form.

- An activator is responsible for:

  i. Registering passive objects that are available for activation which involves recording the names of servers against the URLs or file names of the corresponding passive objects.

  ii Starting named server processes and activating remote objects in them

  iii. keeping track of the locations of the servers for the remote objects that it has already activated.

15.What is persistent object store?

An object that is guaranteed to live between activations of processes is called a persistet object. Persistent objects are generally managed by persistent object stores, which store their state in a marshaled from on disk. In generally a persistent object store will manage very large numbers of persistent object which are stored in a disk or in a database until they are needed.

16. How we decide whether the object is persistent or not.

There are two approaches to deciding whether an object is persistent are not:

i. The persistent object store maintains some persistent roots and any object that is reachable from a persistent root is defined to be persistent. This approach is used by persistent java, java Data Objects and perDis. They make use of a garbage collector to dispose of objects that are no longer reachable from the persistent roots.

ii. The persistent objects store provides some classes on which persistent is based objects belong to their subclasses.

17. Define RPC.

The software components required to implement RPC are

The client that accesses a service includes one stub procedure for each procedure the service interface. The stub procedure behaves like a local procedure to the client but instead of executing the call it marshals the procedure identifier and the arguments into a request message which it sends via its communication module to the server. When the reply message arrives it unmarshals the results.

The server process contains a dispatcher together with one server stub procedure and one service procedure for each procedure in the service interface. The dispatcher selects one of the server stub procedures according to the procedure identifier in the request message.

18.What is event notification?

The distributed event based system extend the local event model by allowing multiple object at different location to be notified of events takes place at an object. They use the publish subscribe paradigm. A publish subscribe system is a system where publishers publish structured events to an event service and subscriber express interest in particular events through subscriptions which can be arbitrary patterns over the structure events.

19. List the example of publish subscribe system.

Publish-subscribe system is used in a wide variety of application domains particularly those related to a large scale dissemination of events.

- Financial information systems.
- Other area with live feeds of real time data(including RSS feeds)

- Supports for cooperative working where a number of participants need to be informed of events of shared interest
- Support for computing including the management of events from the infrastructure

20. List the characteristics of publish-subscribe system.

Heterogeneity: When event notifications are used as a means of communication components in a distributed system that were not designed to interoperate can be made to work together. All that is required is that event generating objects publish the types of events they offer and that other objects subscribe to patterns of events and provide an interface for receiving and dealing with the resultant notification.

21. Define callbacks.

The general idea behind callbacks is that instead of clients polling the server to find out whether some event has occurred, the server should inform its clients whenever that event occurs. The term callback is used to refer to a server's action of notifying clients about an event.

22. How callback is implemented in RMI.

Callback can be implemented in RMI as follows;

i. The client creates a remote object that implements an interface that contains method for the server to call. We refer to that as a callback object.

ii. The server provides an operation allowing interest clients to inform it of the remote object references of their callback objects. It records these in a list.

iii. Whenever an event of interest occurs, the server calls the interested clients, For example, the whiteboard server would call its clients whenever a graphical object is added.

23. List the responsibilities of core OS.

The core OS components and their responsibilities are :

Process manager: Creation of and operations upon process. A process is a unit of resource management, including an address space and one or more threads.

Thread manager: Thread creation, synchronization and scheduling. Threads are schedulable activities attached to processes.

Communication manager: Communication between threads attached to different processes on the same computer some kernels also support communication between threads in remort processes. Other kernels have no notion of other computers built into them, and an additional service is required for external communication.

Memory manager: Management of physical and virtual memory. It describes the utilization of memory management techniques for efficient data copying and sharing.

Supervisor: Dispatching of interrupts, system call traps and other exceptions; control of memory management unit and hardware caches; processor and floating-point unit register manipulation. This is known as the Hardware Abstraction Layer in Windows.

24. Define process.

A process consists of an execution environment together with one or more threads.

25. Define thread.

A thread is the operating system abstraction of an activity (the term derives from the phrase 'thread of execution'). An execution environment is the unit of resource management: a collection of local kernel managed resources to which its threads have access.

26. Define Unix address space.

This representation of an address space as a sparse set of disjoint regions is a generalization of the UNIX address space, which has three regions: a fixed, unmodifiable text region containing program code; a heap, part of which is initialized by values stored in the program's binary file , and which is extensible towards higher virtual addresses ; and a stack, which is extensible towards lower virtual addresses.

27. List the uses of shared region.

The uses of shared regions include the following:

Libraries: Library code can be very large and would waste considerable memory if it was loaded separately into every process that used it.

Kernel: Often the kernel code and data are mapped into every address space at the same location. Data sharing and

Communication: Two processes, or a process and the kernel, might need to share data in order to cooperate on some task. It can be considerably more efficient for the data to be shared by being mapped as regions in both address spaces than by being passed in messages between them.

28. List the architecture of multi threaded server.

- Working pool Architecture
- Thread-per-request Architecture;

- Thread-per-connection Architecture
- Thread-per-object Architecturre:

29. Compare process and threads.

a. Creation a new thread within an existing process is cheaper than creating a process.

b. More importantly switching to a different thread within the same process is cheaper than switching between threads belonging to different processes.

c. Threads within a process may share data and other resources conveniently and efficiently compared with separate processes.

d. But by the same token threads within processes are not protected from one another.

30. Explain thread lifetime.

A new thread is created on the same Java Virtual machine (JVM) as its creator in the SUSPENDED state. After it is made RUNNABLE with the start() method, it execute in the run() method of an object designated in its constructor, The JVM and the threads on top of it all execute in a process on top of the underlying operating system. Threads can be assigned a priority so that a java implementation that supports priorities will run a particular threads in preference to any thread with lower priority.

16 MARKS QUESTION

1. Explain communication between distributed objects.
2. Explain the design issue of RMI.
3. Explain in detail SUN RPC.
4. Explain in detail publish-subscribe system.
5. Explain dealing room system in detail.
6. Explain the architecture of multithreaded servers.
7. Explain OS architecture used in DS.
8. Explain in detail RMI with example.
9. Explain communication and invocation between address space in DS.

8MARKS QUESTION:

1. Explain the implementation of RMI.
2. Explain RPC in detail.
3. Explain the core OS layers functionality.
4. Explain the process creation with an example.
5. Explain in detail about threads.
6. How invocation are made concurrent in DS.

UNIT III

1. Enumerate the properties of storage system?

	Sharing	**Persistent**	**Distributed cache/replicas**	**Consistency maintenance**	**Example**
Main memory	No	No	No	1	RAM
File system n	No	Yes	No	1	UNIX file system
Distributed file system	Yes	Yes	Yes	Yes	Sun NFS
web	Yes	Yes	Yes	No	Web server
Distributed shared memory	Yes	No	Yes	Yes	Ivy(DSM)
Remote objects(RMI/ORB)	Yes	No	No	1	CORBA
Persistent object store	Yes	Yes	No	1	CORBA persistent state service
Peer to peer storage system	Yes	Yes	Yes	2	Ocean Store

2. List out file system modules.

Directory module:	Relates file names to file IDs
File module:	Relates file IDs to particular files
Access control module:	Checks permission for operation requested
File access module:	Read or writes file data or attributes
Block module:	Accesses and allocates disk blocks
Device module:	Disk I/O and buffering

Sketch the file attributes and record structure.

File Length
Creation Time stamp
Read Timestamp
Write time stamp
Attribute time stamp
Reference count
Owner
File Type
Access control List

3.List out the UNIX file system Operations:

fieldes=open(name,mode)

fieldes=create(name,mode)

status=close(fieldes)

count=read(fieldes,buffer,n)

count=write(fieldes,buufer,n)

pos=Iseek(filedes,offset,whence)

status=unlink(nmae)

status=link(name1,nmae2)

status=stat(name,buffer)

4. List out the transparencies in file system.

i. Access transparency
ii. Location transparency
iii. Mobility transparency
iv. Performance transparency
v. Scaling transparency

5. What is meant by concurrency control?

Changes to a file by one client should not interfere with the operation of other clients simultaneously accessing or changing the same file. This is well-known issue of concurrency control .The need for concurrency control for accss to shared data in many applications Is widely accepted and techniques are known for its implementation ,but they are costly .Most current file services follow morden UNIX standards in providing advisery or mandatoryfile or record-level locking.

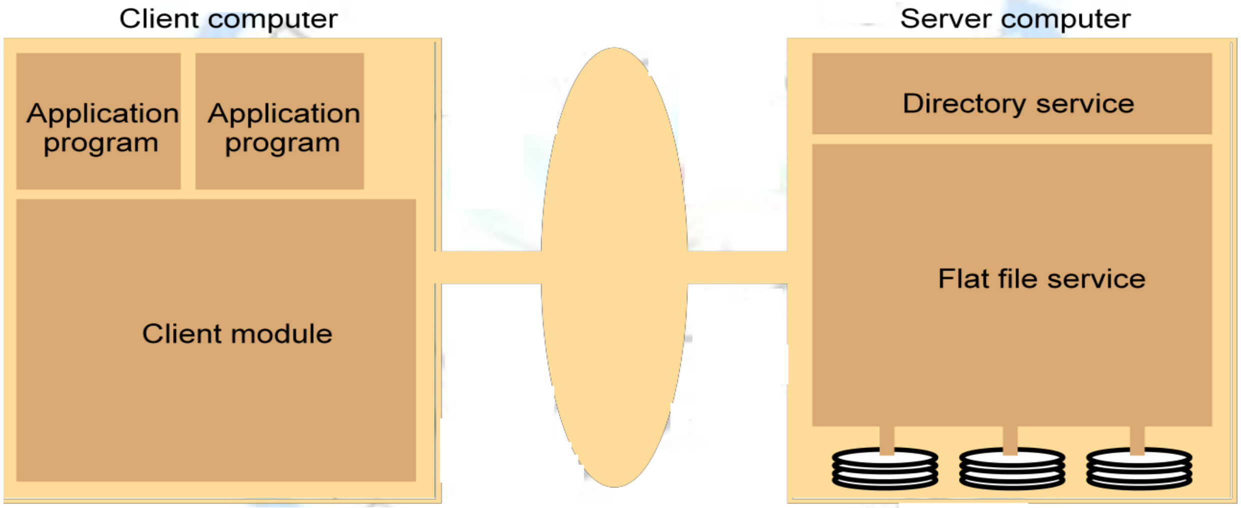
6. What is file replication?

In a file service that supports replication, a file may be represented by several copies of its contents at different locations. This has two benefits-its enables multiple servers to share the load of providing a service to clients accessing the same set of files, enhancing the scalability of the service, and it enhances fault tolerance by enabling clients to locate another server that holds a copy of the file when one has failed. Few file services support replication fully, but most support the catching of files or portions of files locally, a limited form of replication.

7. What is meant by directory services?

The directory services provide a mapping between text names for files and their UFIDs. Client may obtain the UFIDs of a file by quoting its text name to the directory services. The directory services provides the function needed to generate directories, to add new file name to directories and to obtain UFIDs from directories. It is client of the flat file services; its directory is stored in files of the flat services. When a hierarchic file-naming scheme is adopted as in UNIX, directories hold references to other directories.

8.Sketch the file service architecture?



9. List the flat file service operation.

Read (file/d,I,N)>data-throws bad position - if 1 ≤ 1 ≤ length(file):reads a sequence of up to N items From a file starting at item/and returns it in data

Write(File/D,I,Ddata)-throws bad position - if 1 ≤ 1 ≤ length(file)+1: writes a sequence of data to a File, starting at item 1,extending the file if necessary

Create()->FileID -creates a new file of length 0 and delivers a UFID for it

Delete(FileID) -removes the file from the file store

GetAttributes(FileID)->> -returns the file attributes for the file

SetAttributes(FileID) -sets the file attributes(only those attributes that not Shaded in)

10.List the directory service operation.

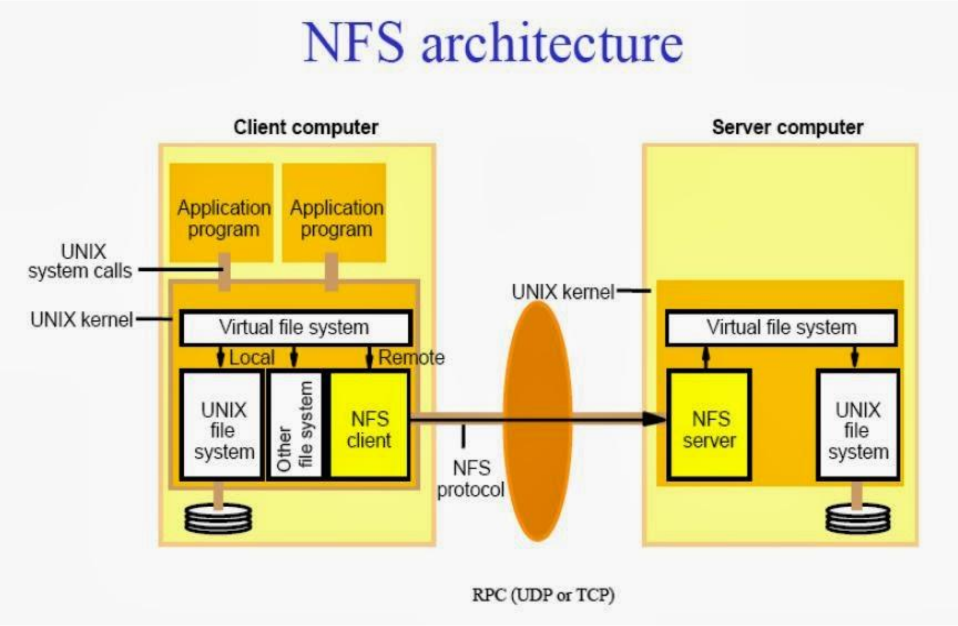
Lookup(Dir,Name)→FileID –throws not found Locate the text name in the directory and returns the relevant UFID.If name is not in the directory, throws an exception

AddName(Dir,Name,File) If the nameis not in the directory, adds(Nmae,File) to the directory and updates the file's Attribute record.If name is already in the Directory;throws an exception.

UnName (Dir,Name)—throws not found If name is in the directory: The entry containing name is removed from the directory.
If name is not in the directory;throws an exception.

GetName(Dir,Pattern)→Names Returns all the text names in the directory that match the regular expression pattern.

11. Sketch NFS architecture.



12. List the NFS file server operation.

Lookup(DirFH,Name)→FH,Attr

Returns File handles and attributes for the File Name in the directory DirFH.

Create(DirFH,Name,Attr)→NewFH,Attr

Creates a new file in directory DirFH with Attributes Attr and Returns the new File handle and Attributes.

Remove(DirFH,Name) Status

Removes file name from directory DirFH.

GetAttr(FH)→Attr

Returns the file attributes of file FH.(similar to UNIX stat system call)

Read(FH,Offset,count)→Attr,Dir

Returns up to count bytes of data from the file starting at the offset ,also return the attributes of the file.

Write(FH,Offset,count,Data)→Attr

Writes count bytes of to a file starting at offset. Returns the attributes of the file after write has taken place.

13. what are the timestamps in called caching?

i. Tc is the time when the cache entry was last validated.
ii. Tm is the when the block was last modified at the server.
iii. A cache entry is valid at time T if T-Tc is less than a freshness interval t,or if the value for Tm recorded t the client matches the value of Tm at the server (that is,the data has not been modified at the server since the cache entry was made).

14.what is condition used to validate caching?

(T-Tc<t) v (Tm client=Tm server)

15.write the measures to be considered to reduce traffic in getattr.

- Whenever a new value of Tmserver is received at a client, it is applied to all cache entries derived from the relevant file.
- The current attribute values are sent 'piggybacked' with the result of every operation on a file, and if the value of Tmserver has changed the client uses it to update the cache entries relating to the file.

- The adaptive algorithm for setting freshness interval t outlined above reduces the traffic considerably for most files.

16. When the name is resolved?

The name is resolved when it is translated into data about the named resource or object, often in order to invoke an action upon it. The association between a name and an object is called a binding. In general, names are bound to attributes of the named objects, rather than the implementation of the objects themselves. An attribute is the value of a property associated with an object.

17. What is meant by URI?

URI-Uniform Resource Identifiers came about from the need to identify resources on the web, and other internet resources such as electronic mailboxes. An important goal was to identify resources in a coherent way, so that they could all be processed by common software such as browser. URIs is 'uniform' in that their syntax incorporates that of indefinitely many individual types of resource identifier(i.e URI schemas),and there are procedures for managing the global namespace of schemas. The advantage of uniformity is that eases the process of introducing new types of identifier, as well as using existing types of identifier in new contexts without disrupting existing usage.

18. What is mean by URN?

Uniform Resource Names are URIs that are used as pure resource names rather than locators. For example, the URI:

Mid:0E4FC272-5C02-11D9-B115-000A95B55BC8@hpl.hp.com

Is a URN that identifies the email message containing it in its ‘message-id’ field. The URI distinguishes that message from any other email message. But it does not provide the message’s address in any store, so a lookup operation is needed to find it.

19.What is global name services?

The Global name Service developed at the Digital Equipment corporation systems, Research Center,is a descentdant of Grapevine with ambitious goals, including:

- com- Commercial organization.
- edu- universities and other educational institutions.
- gov- US governmental agencies
- mil- US military organization
- net- major network support centers
- org- organizations not mentioned above
- int – International oragnisations.

20. What is meant by navigation?

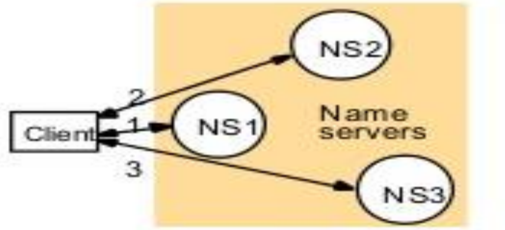
The process of locating naming data from than more than one name server in order to resolve a name is called navigation. The client name resolution software carries out navigation on behalf of the client. It communicates with name servers as necessary to resolve a name.

21. What is multicast navigation?

In multicast navigation, a client multicast the name to be resolved and required object type to the group of name servers. Only the server that holds the named attributes responds to the request. Unfortunately, however, if the name proves to be unbound, the request is greeted with silence.

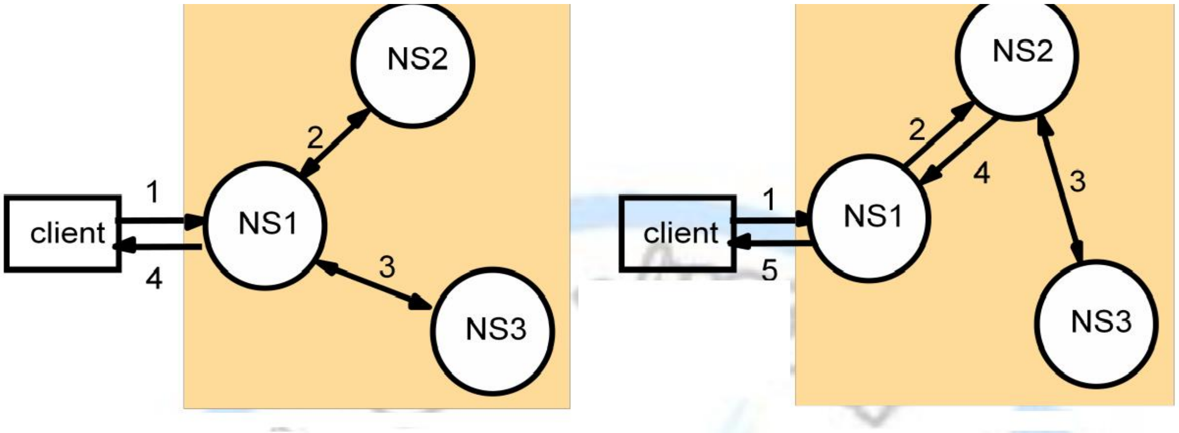
22. What is iterative navigation?

One navigation model that DNS supports is known as iterative navigation. To resolve a name, a client present the name to the local name server, which attempts to resolve it. If the local name server has the name, it returns the result immediately. If it does not it will suggest another server that will ba able to help. Resolution proceeds at the new server, with further navigation as necessary until the name is located or is discovered to be unbounded.



23. What is meant by recursive and non recursive navigation?

In the non recursive and non recursive server controlled navigation. Under non recursive server controlled navigation ,any name server may be chosen by the client. This server communicates by multicast or iteratively with its peer in the style described above, as through it were a client. Under recursive server-controlled navigation ,the client once more contacts a single server. If this server doesnot store the name,the server contains a peer to storing a prefix of the name,which in turn attempts to resolve it. This procedure continues recursively until the name resolved.



Non-recursive server-controlled

Recursive server-controlled

A name server NS1 communicates with other name servers on behalf of a client

24. Write the disadvantage of existing name service?

This original scheme was soon seen to suffer from three major shortcomings:

- It did not scale to large numbers of computers.
- Local organization wished to administer their own naming system.
- A general name service was needed- not one that serves only for looking up computers address.

25. What is meant by DNS?

The domain name system is a name service design whose main naming database is used across the internet. It was devised principally by mockapetris and specified in RFC 1034 and RFC 1035. DNS replaced the original internet naming scheme in which all host names and address were held in a single central master file and downloaded by FTP to all computer that required them.

Domain Names: The DNS is designed for use in multiple implementations, each of which may have its own name space. In practice, however, only one is in widespread use, and that is one used for naming across the internet. The internet DNS name space is partitioned both organizationally and according to geography. The names are written with the highest-level domain on the right. The original top-level organizational domains in use across the internet were:

26. What is meant by zone?

The DNS naming data are divided into zones. A zone contains the following data:

- Attributes data for names in a domain, less any subdomains administrates by lower level authorities.
- The names and address of at least two name servers that provide authoritative data for the zone. These are versions of zone data that can be relied upon as being reasonably up to date.
- The names of name servers that hold authoritative data for delegated sub domains; and ‘glue’ data giving the IP address of these servers.
- Zone-management parameters, such as those governing the catching and replication of zone data.

27. What is lookup operation?

28. List out options of NFS write operation?

Data in write operation received from client is stored in the memory cache at the server and written to disk before a reply is sent to the client. This is called writethrough caching. The client can be sure that is data stored persistently as soon as reply has been received.

Data in write operation is stored only in the memory cache. It will be written to disk when a commit operation is received for the relevant file. The client can be sure that the data is persistent stored only when a reply to a commit operation for the relevant file has been received. Standard NFS clients use this mode of operation, issuing a commit whenever a file that was open for writing is closed.

29. What is BIND implementation of DNS?

The Berkeley internet name domain is an implementation of the DNS for computers running UNIX. Client programs link in library software as the resolver. DNS name server computers run the named deamon. BIND allows for three categories of name server: primary server, secondary yserver, caching only servers. The named program implements just one of these types, according to the content of a configuration file. Caching only servers read in form a configuration file sufficient names and address of authoritative servers to resolve any name. Thereafter, they only store this data and data they learn by resolving names for clients.

30. Write the motivation openness?

Name management is separated from other service largely because of the openness of distributed system, which brings the following motivation:

- Unification
- Integration

31. Define name space.

A namespace is the collection of all valid names recognized by a particular service. The service will attempt to look up a valid name, even though that name may prove not to correspond to any object. Name space requires a syntactic definition to separate.

PART B

1. Discuss in details file service architecture
2. Explain in SUN Network file system
3. Explain the name service in details

4. Explain Domain Name System
5. Describe the character tics of file system
6. List the NFS server operation
7. Explain Global Name Services in details
8. Explain in details name server navigation and its types

UNIT IV

1. **What is meant by hardware and software clock?**

Clock devices can be programmed to generate interrupts at regular intervals in orders that, for example, time slicing can be implemented.The operating system reads the node’s hardware clock value, H(t) , scales it and adds an offset so as to produce software clock $C\ \ (t)=\alpha H_i(t)+\beta$ that approximately measures real ,physical time t for process pi.

2. **What is clock resolution?**

Note that successive events will correspond to different timestamps only if the clock resolution-the period between updates of the clock-value-is smaller than the time interval betw4een successive events. The rate at which events occur depends on such factors as the length of the processor instruction cycle.

3. **What is clock drift?**

Clock drift, which means that they count time at different rates and so diverge. The underling oscillators are subject to physical variations, with the consequence that their frequencies of oscillation differ. Moreover ,even the same clock’s frequency varies with temperature. Designs exist that attempt to compensate for this variation, but they cannot eliminate it. A clock’s drift rate is the change in the offset (difference in reading) between the clock and a nominal perfect reference clock per unit of time measured by the reference clock.

4. **What is IAT?**

Computer clocks can be synchronized to external sources of highly accurate time. The most accurate physical clocks use atomic oscillators, whose drift rate is about one part in 1013.the most accurate physical clocks use atomic oscillators, whose drift rate is about one part in 1013. The output of these atomic clocks is used as the standard for elapsed real a time, known as International Atomic Time.

5. **What is CUT?**

Coordinated Universal Time-abbreviated as UTC (From the French equivalent)-is as international standard for timekeeping. It is based on atomic time, but a so-called ‘leap second’ is inserted-or, more rarely, deletedoccasionally to keep it in step with astronomical time.

6. **What is meant by external synchronization?**

In order to know at what time of day events occur at the processes in our distributed system –it is necessary to synchronize the processes' clocks, C , with an authoritative, external source of time. This is external synchronization.

We define these two modes of synchronization more closely as follows, over an interval of real time I:

- For a synchronization bound D>0,$|C_i(t)-C_j(t)|<D$ for i=0,1,2,… N and for a source S of UTC time, $|S(t)-C_i(t)|<D$,for i=1,2,.. and for all real times t in I.
- The clocks C_i are accurate to within the bound D.

7. **What is internal synchronization?**

And if the clocks C are synchronized with one another to known degree of accuracy, then we can measure the interval between two events occurring at different computers by appealing to their local clocks, even though they are not necessarily synchronized to an external source of time. This is internal synchronization.

- For a synchronization bound D > 0 and for a source S of UTC times, $|S(t)-C_i(t)| < D$, for all real times t in I.
- Clocks C agree with in the bound D.

8. **Define NTP and its design aims**.

Cristian's method and the Berkeley algorithm are intended primarily for use within intranets. The Network Time Protocol (NTP) [Mills1995] defines an architecture for a time service and a protocol to distribute time information over the Internet.

NTP's chief design aims and features are as follows:

- To provide a service enabling clients across the Internet to be synchronized accurately to UTC:
- To provide a reliable service that can survive lengthy losses of connectivity:
- To enable clients to resynchronize sufficiently frequently to offset the rates of drift found in most computers: To provide protection against interference with the time service, whatever malicious or accidental.

9. What is strata?

The NTP service is provided by a network of servers located across the Internet. Primary servers are connected directly to a time source such as a radio clock receiving UTC; secondary servers are synchronized, ultimately, with primary servers. The servers are connected in a logical hierarchy called a synchronization subnet whose levels are called strata.

10. Enumerate the mode of synchronization in NTP servers.

- NTP servers synchronize with one another in one of three: multicast, procedure-call and symmetric mode
- ***Multicast mode*** is intended for use on a high-speed LAN. One or more servers periodically multicasts the time to the servers running in other computers connected by the LAN, Which set their clocks assuming a small delay. This mode can achieve only relatively low accuracies, but ones that nonetheless are considered sufficient for many purposes.
- ***Procedure-call mode*** is similar to the operation of Cristian's algorithm. In this mode, one server accepts requests from other computers, which it processes by replying with its timestamp (current clock reading). This mode is suitable when higher accuracies are required than can be achieved with multicast, or where multicast is not supported in hardware.
- ***In, symmetric mode*** is intended for use by the servers that supply time information in LANs and by the higher levels of the synchronization subnet, where the highest accuracies are to be achieved.

11. What is filter dispersion?

NTP servers apply a data filtering algorithm to successive pairs which estimates the offset o and calculates the quality of this estimates as a statistical quantity called the filter dispersion.

12. What is synchronization dispersion?

Peers with lower stratum numbers are more favoured than those in higher strata because they are 'closer' to the primary time sources. Also, those with the lowest synchronization dispersion are relatively favoured. This is the sum of the filter dispersions measured between the server and the root of the synchronization subnet.

13. What is meant by HB relation?

- Lamport called the partial ordering obtained by generalizing these two relationships the happened-before relation. It is also sometimes known as the relation of causal ordering or potential causal ordering.
- We can define the happened-before relation, denoted -> by as follow:

  HB1: If process p_i:e->$_i$e’ , then e->e’

  HB2: For any message m, send (m) receive (m)

  HB3: IF e, e’ and e” are events such that e->e’ then e->e”

14. What is logical clock?

- Lamport [1978] invented a simple mechanism by which the happened before ordering can be captured numerically, called a logical clock.
- A Lamport logical clock is a monotonically increasing software counter, whose value need bear no particular relationship to any physical clock.
- Each process p keeps its own logical clock, L , which it uses to apply so-called Lamport timestamps to a events.
- We denote the timestamp of event e at pi by L (e) , and by L(e) we denote the timestamp of event e at whatever process it occurred at.

15. Define Vector clock

Vector clocks for a system of N processes is an array of N integers

• Shortcoming of Lamport clocks:

$$L(e) < L(e') \text{ doesn't imply } e \rightarrow e'$$

• Vector clock: an array of N integers for a system of N processes

– Each process keeps its own vector clock Vi to timestamp local events

– Piggyback vector timestamps on messages

• Rules for updating vector clocks:

– $Vi[i]]$ is the number of events that pi has timestamped

– $Viji]$ ($j \neq i$) is the number of events at pj that pi has been affected by

VC1: Initially, $Vi[\ j\ ] := 0$ for pi, j=1.. N (N processes)

VC2: before pi timestamps an event, $Vi[\ i\ ] := Vi[\ i\ ]+1$

VC3: pi piggybacks $t = Vi$ on every message it sends

VC4: when *pi* receives a timestamp *t*, it sets *Vi*[*j*] := max(*Vi*[*j*] , *t*[*j*]) for *j*=1..*N* (merge operation)

16. What do you meant by distributed garbage

An object is considered to be garbage if there are no longer any reference to it anywhere in the distributed system. The memory taken up by that object can be reclaimed once it is known as to be garbage.

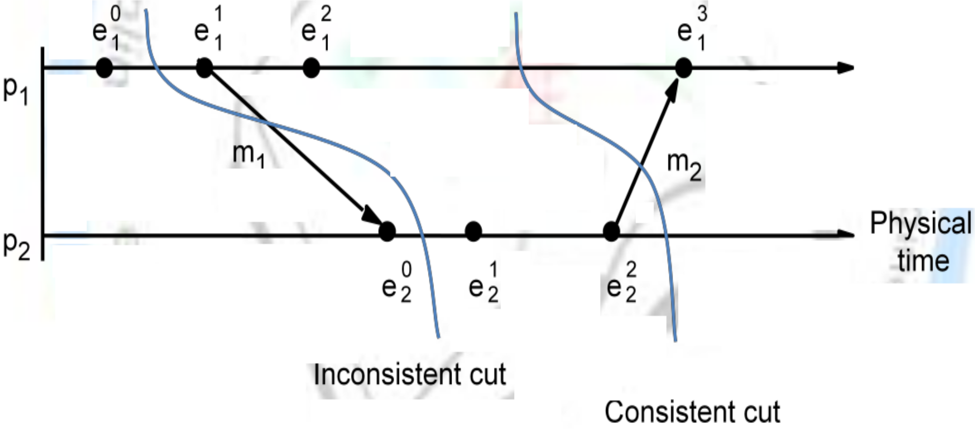
17. Define Global History

Let us return to our general system p of N processes p_i(i=1,2,3,…..N)

Here a series of events occurs at each process, and that we may characterize the execution of each process by its history

18. What is meant by cut?

Consider the events occurring at processes p_1 and p_2 shown in figure



19. Define Global state predicate

- A Global State Predicate is a function that maps from the set of global process states to True or False.
- Detecting a condition like deadlock or termination requires evaluating a Global State Predicate.
- A Global State Predicate is stable: once a system enters a state where it is true, such as deadlock or termination, it remains true in all future states reachable from that state.
- However, when we monitor or debug an application, we are interested in non stable predicates.

20. List the assumption considered in snapshot algorithm

- – Neither channels nor processes fail
- – Reliable communications ensure every message sent is received exactly once
- – Channels are unidirectional
- – Messages are received in FIFO order
- – There is a path between any two processes
- – Any process may initiate a global snapshot at any time
- – Processes may continue to function normally during a snapshot

21. Define Failure detector.

A failure detector is a service that processes queries about whether a particular process has failed .It is often implemented by an object local to each process that runs failure detection algorithms in conjunction with its counterparts at the other processes.

22. List the properties of failure detector

A failure detector is not necessarily accurate. Most fails into the category of ***unreliable failure detectors.***

- A result of unsuspected
- A result of Suspected

23. Define critical section problem

The application – level protocol for executing a critical section is as follows

- enter() - enter critical section – block if necessary
- resurceAccesses() - access shared resources in critical section
- exit() - leave critical section other processes may now enter.

24. What is meant by election

Election: choosing a unique process for a particular role is called an election

– All the processes agree on the ***unique*** choice

– For example, server in dist. mutex

25. List the famous mutual exclusion algorithms

- Center server algorithm
- Ring- Based algorithms
- Mutual Exclusion using multicast and Logical Clocks

- Maekawa’s Voting algorithms
- Mutual Exclusion algorithms comparison

26. What do you meant by bully algorithms and types of messages

Assumption: Each process knows which processes have higher identifiers, and that it can communicate with all such processes

•Compare with ring-based election

– Processes can crash and be detected by timeouts

• synchronous

• timeout *T* = 2*Ttransmitting* (max transmission delay) + *Tprocessing* (max processing delay)

Three types of messages

a. Election: announce an election

b. Answer: in response to Election

c. Coordinator: announce the identity of the elected process

27. What are the types of ordering in multicast

Three types of message ordering

- **FIFO (First-in, first-out) ordering**: if a correct process delivers a message before another, every correct process will deliver the first message before the other
- **Casual ordering**: any correct process that delivers the second message will deliver the previous message first
- **Total ordering**: if a correct process delivers a message before another, any other correct process that delivers the second message will deliver the first message first

28. Define Consensus

Consensus more precisely and relates it to three related Problems of agreement.For processes to agree on a value (consensus) after one or more of the processes has proposed what that value should be Covered topics: ***byzantine generals, interactive consistency, totally ordered multicast***

• The byzantine generals problem: a decision whether multiple armies should attack or retreat, assuming that united action will be more successful than some attacking and some retreating

• Another example might be space ship controllers deciding whether to proceed or abort. Failure handling during consensus is a key concern

29. What are the requirements of interactive consistency?

Three requirements of a consensus algorithm

- ***Termination***: Eventually every correct process sets its decision variable
- ***Agreement***: The decision value of all correct processes is the same: if pi and pj are correct and have entered the *decided* state, then di=dj for all (i,j=1,2, …, N)
- ***Integrity***: If the correct processes all proposed the same value, then any correct process in the *decided* state has chosen that value

30. What are the requirements of consensus algorithms

The requirements of a consensus algorithms are that are following conditions should hold for every execution of it:

- Termination
- Agreement
- Integrity

PART B (16 MARK QUESTIONS)

1. Explain clocks, events and process state in details
2. Discuss in details any two algorithms used for synchronizing clocks in intranet
3. Explain logical clocks in details
4. Discuss the snapshot algorithms
5. Explain distributed mutual exclusion algorithms in details
6. Explain group and multicast communication in details
7. Explain ordering multicast communication and its types
8. Explain consensus and its related problems in details
9. Explain ring based election algorithms
10. Explain bully’s algorithms
11. Explain global states and consistent cuts in details

UNIT V

1. **What do you meant by DSM**

   Distributed shared memory (DSM) is an abstraction used for sharing data between computers that do not share physical memory. Processes access DSM by reads and updates to what appears to be ordinary memory within their address space. However, an underlying runtime system ensures transparently that processes executing at different computers observe the updates made by one another.

2. **List the three approaches of DSM structure**

   - Hardware
   - Paged virtual memory
   - Middleware

3. **Define Sequential consistency**

   A DSM system is said to be sequentially consistent if *for any execution* there is some interleaving of the series of operations issued by all the processes that satisfies the following two criteria:

   SC1: The interleaved sequence of operations is such that if R(x) a occurs in the sequence, then either the last write operation that occurs before it in the interleaved sequence is W(x) a, or no write operation occurs before it and *a* is the initial value of *x*.

   SC2: The order of operations in the interleaving is consistent with the program order in which each individual client executed them.

4. **What is coherence**

   Coherence is an example of a weaker form of consistency. Under coherence, every process agrees on the order of write operations to the same location, but they do not necessarily agree on the ordering of write operations to different locations. We can think of coherence as sequential consistency on a locationby- location basis. Coherent DSM can be implemented by taking a protocol for implementing sequential consistency and applying it separately to each unit of replicated data – for example, each page.

5. **What is meant by weaker consistency**

   This model exploits knowledge of synchronization operations in order to relax memory consistency, while appearing to the programmer to implement sequential consistency (at least, under certain conditions that are beyond the scope of this book). For example, if the programmer uses a lock to implement a critical section, then a DSM system can assume that no other process may access the data items accessed under mutual exclusion within it. It is therefore redundant for the DSM system to propagate updates to these items until the process leaves the critical section. While items are left with 'inconsistent' values some of the time, they are not accessed at those points; the execution appears to be sequentially consistent.

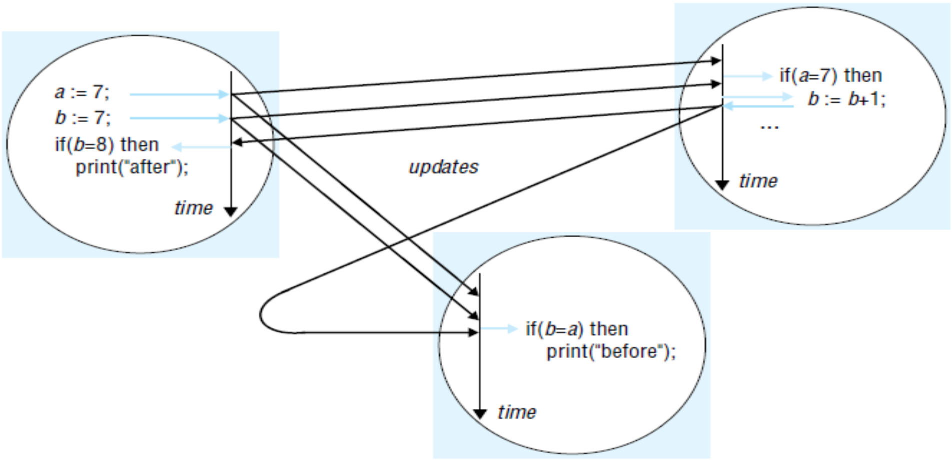
6. What is granularity

An issue that is related to the structure of DSM is the granularity of sharing. Conceptually, all processes share the entire contents of a DSM. As programs sharing DSM execute, however, only certain parts of the data are actually shared and then only for certain times during the execution. It would clearly be very wasteful for the DSM implementation always to transmit the entire contents of DSM as processes access and update it.

7. What is meant by multiple reader/writer sharing

Write-update: The updates made by a process are made locally and multicast to all other replica managers possessing a copy of the data item, which immediately modify the data read by local processes. Processes read the local copies of data items, without the need for communication. In addition to allowing multiple readers, several processes may write the same data item at the same time; this is known as multiple-reader/multiple-writer sharing.

DSM using write-update



8. List the types of manager strategy

There are three alternatives:

a) Fixed distributed page management

b) Multicast based management

c) Dynamic distributed management

9. What is release consistency

Release consistency was introduced with the Dash multiprocessor, which implements DSM in hardware, primarily using a write-invalidation protocol [Lenoski *etal*. 1992]. Munin and Treadmarks [Keleher *et al*. 1992] have adopted a software implementation of it. Release consistency is weaker than sequential consistency and cheaper to implement, but it has reasonable semantics that are tractable to programmers.

The idea of release consistency is to reduce DSM overheads by exploiting the fact that programmers use synchronization objects such as semaphores, locks and barriers.

10. List the types of memory access

The main distinction is between *competing* accesses and *noncompeting* (*ordinary*) accesses. Two accesses are competing if:

- they may occur concurrently (there is no enforced ordering between them) and
- at least one is a *write*.

11. What are the requirement of release consistency

The requirements that we wish to meet are:

- to preserve the synchronization semantics of objects such as locks and barriers;
- to gain performance, we allow a degree of asynchronicity for memory operations;
- to constrain the overlap between memory accesses in order to guarantee executions that provide the equivalent of sequential consistency.

12. List out the points to be considered for Munin's implementation of release consistency

The following points apply to Munin's implementation of release consistency:

- Munin sends update or invalidation information as soon as a lock is released.
- The programmer can make annotations that associate a lock with particular data items. In this case, the DSM runtime can propagate relevant updates in the same

message that transfers the lock to a waiting process – ensuring that the lock's recipient has copies of the data it needs before it accesses them.

13. What is meant by casual consistency

Causal consistency: Reads and writes may be related by the happened-before relationship . This is defined to hold between memory operations when either (a) they are made by the same process; (b) a process reads a value written by another process; or (c) there exists a sequence of such operations linking the two operations. The model's constraint is that the value returned by a read must be consistent with the happened-before relationship.

14. Define processor consistency

Processor consistency: The memory is both coherent and adheres to the pipelined RAM model (see below). The simplest way to think of processor consistency is that the memory is coherent and that all processes agree on the ordering of any two write accesses made by the same process – that is, they agree with its program order.

15. Define CORBA

CORBA is a middleware design that allows application programs to communicate with one another irrespective of their programming languages, their hardware and software platforms, the networks they communicate over and their implementers.

Applications are built from CORBA objects, which implement interfaces defined in CORBA's interface definition language, IDL. Clients access the methods in the IDL interfaces of CORBA objects by means of RMI. The middleware component that supports RMI is called the Object Request Broker or ORB.

16. What are the steps to be taken for semantics parameter passing in CORBA IDL

Passing CORBA objects:

Any parameter whose type is specified by the name of an IDL interface, such as the return value *Shape* in line 7, is a reference to a CORBA object and the value of a remote object reference is passed.

Passing CORBA primitive and constructed types:

Arguments of primitive and constructed types are copied and passed by value. On arrival, a new value is created in the recipient's process. For example, the *struct GraphicalObject* passed as argument (in line 7) produces a new copy of this *struct* at the server.

Type *Object:*

Object is the name of a type whose values are remote object references. It is effectively a common super type of all of IDL interface types such as *Shape* and *ShapeList*.

17. What is meant by CORBA naming service

It is a binder that provides operations including *rebind* for servers to register the remote object references of CORBA objects by name and *resolve* for clients to look them up by name. The names are structured in a hierarchic fashion, and each name in a path is inside a structure called a *Name Component*. This makes access in a simple example seem rather complex.

18. What is meant by CORBA security service?

CORBA Security Service

The CORBA Security Service [Blakley 1999, Baker 1997, OMG 2002b] includes the following:

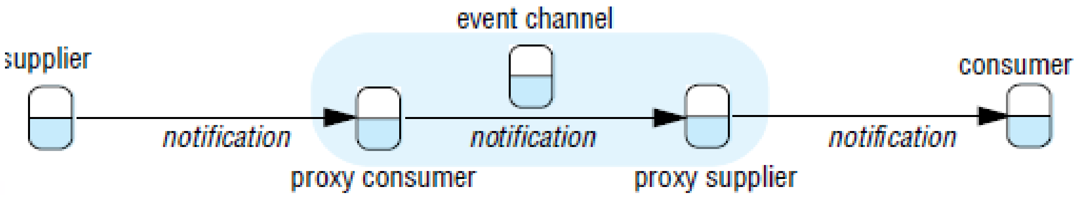
- Authentication of principals (users and servers); generating credentials for principals (that is, certificates stating their rights); delegation of credentials is supported
- Access control can be applied to CORBA objects when they receive remote method invocations. Access rights may for example be specified in access control lists (ACLs).
- Security of communication between clients and objects, protecting messages for integrity and confidentiality.
- Auditing by servers of remote method invocations.
- Facilities for non-repudiation. When an object carries out a remote invocation on behalf of a principal, the server creates and stores credentials that prove that the invocation was done by that server on behalf of the requesting principal.

19. What is meant by CORBA notification services

The CORBA Notification Service extends the CORBA Event Service, retaining all of its features including event channels, event consumers and event suppliers. The event service provides no support for filtering events or for specifying delivery requirements. Without the use of filters, all the consumers attached to a channel have to receive the same notifications as one another. And without the ability to specify delivery requirements, all of the notifications sent via a channel are given the delivery guarantees built into the implementation.

20. What is meant by CORBA event services?

The CORBA Event Service specification defines interfaces allowing objects of interest, called *suppliers*, to communicate notifications to subscribers, called *consumers*. The notifications are communicated as arguments or results of ordinary synchronous CORBA remote method invocations. Notifications may be propagated either by being *pushed* by the supplier to the consumer or *pulled* by the consumer from the supplier.



PART B (16 MARK QUESTIONS)

1. Explain DSM and its implementation in details
2. Explain the consistency models in DSM
3. Discuss Ivy system model in details
4. Explain CORBA RMI services and its features
5. Explain CORBA services with example
6. Explain CORBA Naming services and IDL with an example
7. Explain CORBA notification services elaborately
8. Explain sequential consistency with example