

Internet:- It is used to connect different network of computers simultaneously. It is a public network therefore anyone can access the internet. In internet, there are multiple users and it provides unlimited number of information to the users.

Intranet:- It is the type of internet which is used by privately. It is a private network therefore anyone can't access intranet. In intranet, there are limited number of users and it provides limited number of information to its users.

(b) Load Balancing:- It is the way of distributing load units across a set of processors which are connected to a network which may be distributed across the global.

A load ~~bal~~ Balancing operation consists of 3 rules:-  
Location Rule, Distribution Rule & Selection Rule

The selection rule works either in preemptive or in non-preemptive fashion. The newly generated process is always picked up by non-preemptive rule while running process may be picked up by preemptive Rule.

Optimizing Approaches:- To satisfy the demanding optimization & Quality of service (QoS) requirements exhibited by complex DS.

(2)

- (i) Server transparency
- (ii) Decentralized balancing
- (iii) Stateful replicas
- (iv) Diverse load monitoring
- (v) fault tolerant load Balancing
- (vi) Extensible load Balancing
- (vii) on-demand Replica Activation

(c) Synchronization  $\Rightarrow$  The correct behavior of concurrent program demands on synchronisation & comm<sup>n</sup> b/w its processes.

$\rightarrow$  The satisfaction of constraints on the interleaving of actions of processes.

$\rightarrow$  It is achieved via clocks.

The clock synchronization can be achieved by 2 way

(i) External clock Synchronization  $\Rightarrow$  In which an external reference clock is present. It is used as reference & nodes in system can set & adjust their time.

(ii) Internal clock Synchronization  $\Rightarrow$  In which each node shares its time with other nodes & all nodes set & adjust their time.

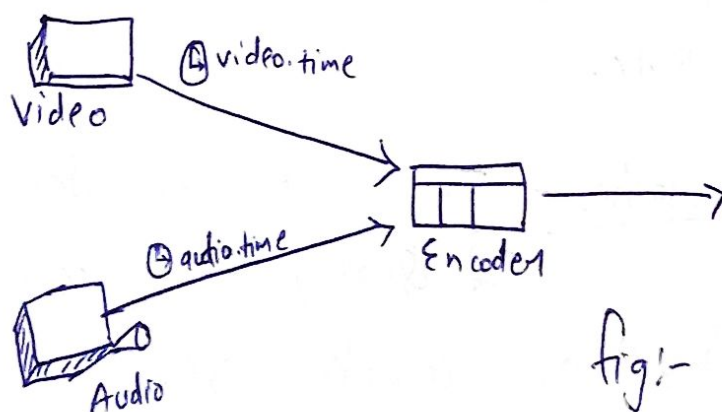
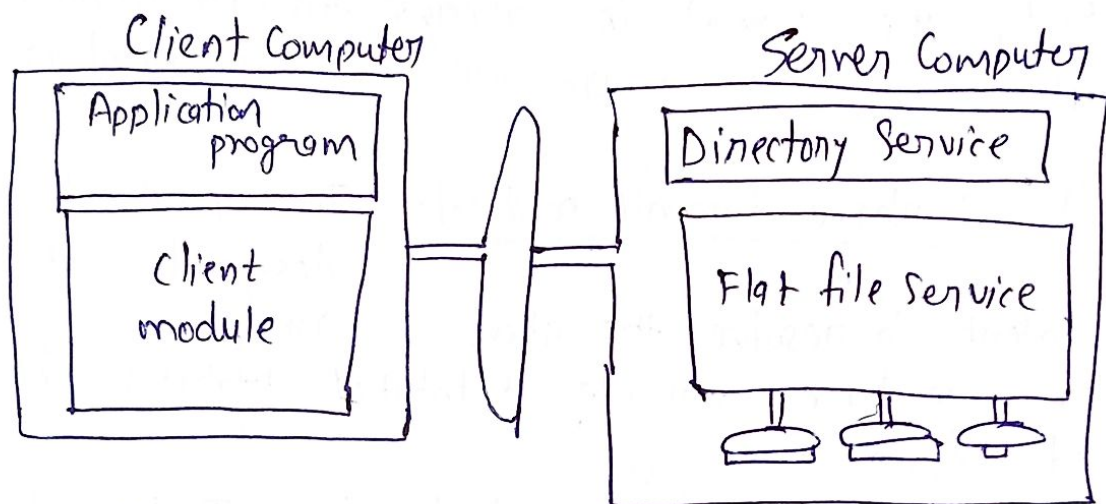


fig:- Clock Synchronization.



(d) File Service Architecture:- An architecture that offers a clear separation of the main concerns in providing access to file obtained by structuring file service as 3 components:- (3)

- A flat file service
- A Directory Service
- A client module



Flat file Service:- UFIDs are used to refer to files in all request for FFS.

Directory Service:- Client may obtain UFID of file by quoting its text name to directory Service

Client modules:- It runs on each computer & provides integrated service as a single API to application program.

(\*) Q - (6) The route for transmitting a packet in a wide area Network is determined dynamically for each packet. So, there is no guarantee that all the packets addressed to the same destination will follow the same route or at least will need the same

amount of time to reach destination. TCP readdresses this by adding sequence numbers to the packets and re-ordering them at the receiving host. (4)

It can't happen in LANs because there is only a single comm<sup>n</sup> medium, the messages are transmitted in strict sequence.

The data transmission in ATM networks is connection oriented, the comm<sup>n</sup> is always through virtual circuits & these assure the order preservation.

Q-2) 2 phase Commit protocol  $\Rightarrow$  This protocol is designed with the core intent to resolve the above programs. Consider we have multiple distributed databases which are operated from servers.

Let's say  $S_1, S_2 \dots S_n$  and transaction  $T$  has also been divided into subtransactions  $T_1, T_2 \dots T_n$  and each  $T_i$  are assigned to  $S_i$ . This all maintains by separate transaction managers at each  $S_i$ .

We assigned anyone site as a Coordinator.

$\rightarrow$  Some points to be considered regarding this protocol: —

- (a) Each site logs actions at that site but not global.
- (b)  $C_i$  plays role in doing confirmation.
- (c) messages are made to send b/w coordinator ( $C_i$ ) and other sites.

Phase - 1 :-

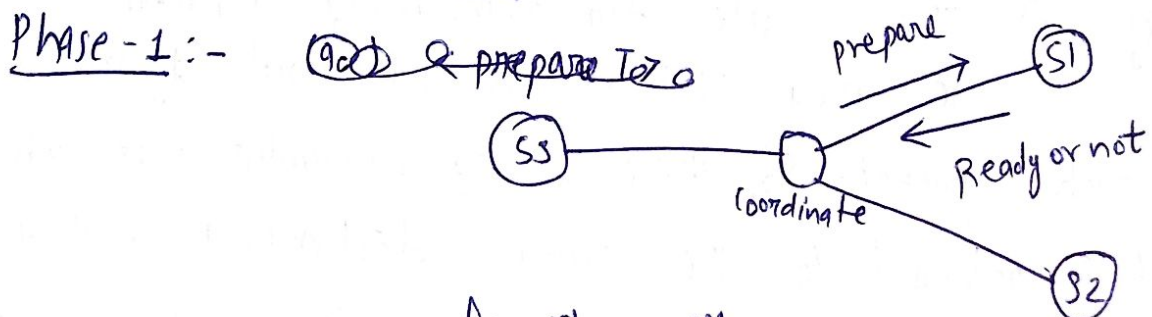
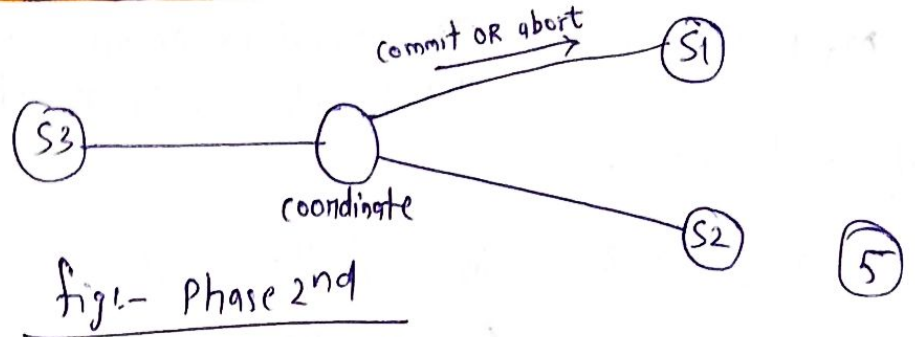


fig:-phase - 1st



Phase 2nd :-



Ans - (5) An individual system can provide resources and use them from another system at same time. This means that a device could be both a server and a client at the same time.

→ A server is a system that provides resources, data, services or programs to other computers, known as client, over a network.

→ To function as a server, device must be configured to listen to requests from clients on network connection. This functionality can exist as part of OS as an installed application, role or a combination of two.

Types of Servers:-

File Servers:- It stores & distribute files. Multiple clients may share file stored on a server.

Print Servers:- It allows for management and distribution of printing functionality.

Application Servers:- It runs application in lieu of client computers running application locally. Application Servers often run resource intensive applications.

DNS Servers:- It provides name resolution to client computers by converting names easily understood by human into IP addresses.

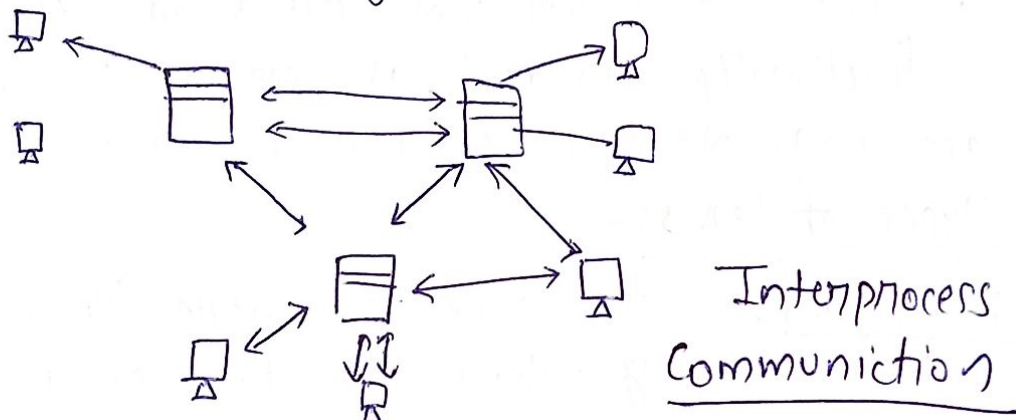
Mail Servers:- It receives emails sent to user & store them til requested by client on behalf of said user. (6)

Web Servers:- One of the most abundant types of servers. A web server is a special kind of application server that hosts programs & data requested by users across the internet or an intranet.

Ans - (4) IPC Communication is a process of exchanging the data b/w 2 or more independent process in distributed enviro. is called as Interprocess Communication.

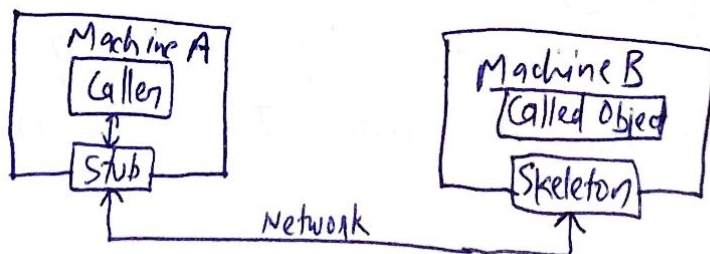
It has 2 functions :-

- (i) Synchronization
- (ii) Message Passing



Invocation:- Distributed Object Commn realizes commn b/w distributed objects.

Invoking a method on the remote object is known as remote method invocation (RMI) and is the object oriented programming analog of Remote procedure call (RPC).





~~Q. 5~~ Section (I)  
Ay-(e)

(7)

Client sends request message, times out & then retransmits the request message, expecting only one reply.

The server which is operating under a heavy load, eventually receives both request messages and sends two replies. When client sends subsequent request it will receive reply from earlier call as a request. If request identifiers are copied from request to reply messages, the client can reject reply to earlier message.

Ay (5) Distributed Process Implementation:-

3 significant application Scenario

- (i) Remote Service:- The message is interpreted as a request for a known service at remote site
- (ii) Remote Execution:- The message contains a program to be executed at remote site.
- (iii) Process Migration:- The messages representing process are migrated to remote site for continuing execution.

(a) Relocation:- It is the process of assigning load addresses for position-dependent code & data of a program & adjusting the code & data to reflect the assigned addresses.

Migration:- It is specialized form of process management whereby processes are moved from one computing environment to another.

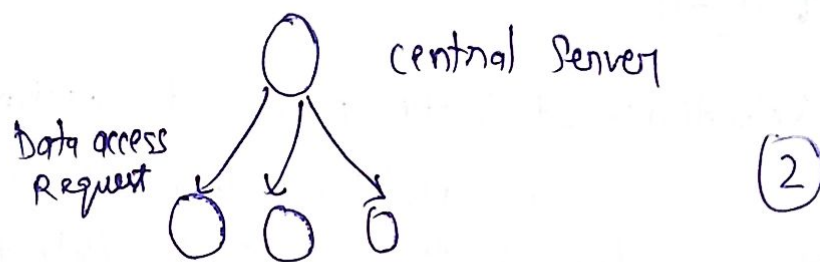
Transparency:- The concealment from user & the application programmer of separation of components in a DS.

(b) Distributed Shared memory:- It is a form of memory architecture where physically separated memories can be addressed as one logically shared address space. Shared means there is single centralized memory, but that the address space is shared.

Central Server Algorithm → All shared data is maintained by the central server. Other nodes of the DS request for reading & writing data to the server which serves



the request & updates or provides access to the data along with acknowledgment messages.



(c) Lamport timestamp algorithm is a simple logical clock algo used to determine the order of events in DS.

→ It is a numerical s/w counter value maintained in each process.

The counter is incremented before the event timestamp is assigned. If a process has 4 events a, b, c, d they would get Lamport timestamp of 1, 2, 3, 4.

(d) Randomised Distributed Agreement is one of the fundamental problems in the theory & practice of fault-tolerant DS.

It requires  $n$  processes to agree on an initial value held by one of them, despite the behavior of a subset of size  $t$ .

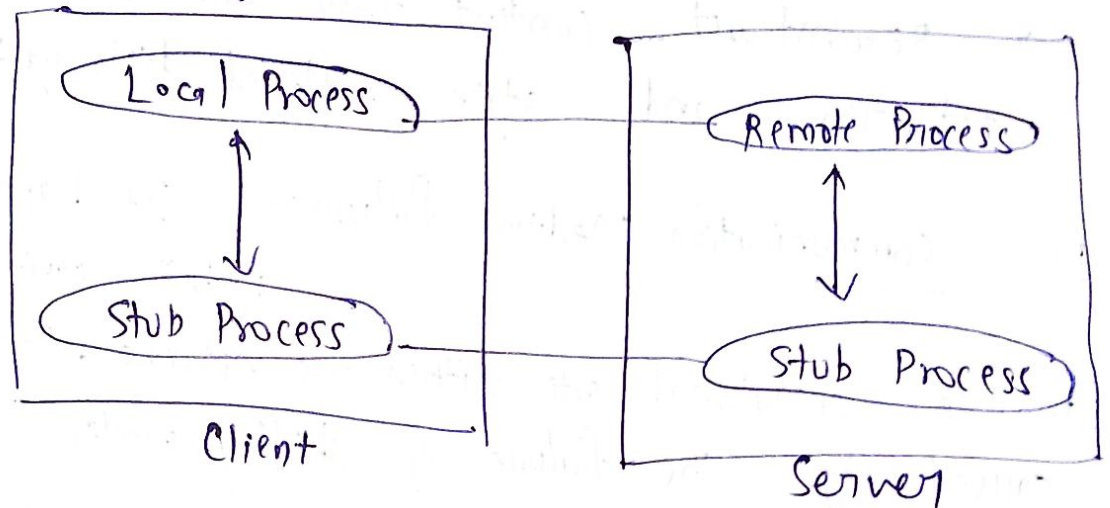
We present a succinct & efficient RDA for asynchronous networks that works for  $n > 5t$  processors.

## (P) Application of Distributed process implementation

⇒

- Remote Service
- Remote Execution
- Process Migration

(3)



### Section - (3)

#### (a) Various failures in DS ⇒

(I) Method Failure :- The DS is generally halted & unable to perform the execution. Sometimes it leads to ending up the execution resulting in an associate incorrect outcome.

→ Method can be prevented by aborting the method or restarting it from its prior state.

(II) System failure :- The processor associated with DS fails to perform the execution.

This is caused by computer code errors & h/w issues.

→ This can be cured by rebooting the system as soon as possible & configuring the failure point & wrong state.



③ Secondary Storage device failure:— It is claimed to have occurred once the keep information can't be accessed. This failure is sometimes caused by parity error, head crash, or dirt particles.

→ Reconstruct content from the archive & log of activities and style reflected disk system.

④ Communication Medium failure:— It happens once a website can't comm<sup>n</sup> with another operational site within the network. Its typically caused by the failure of shift nodes.

→ Reroute, error-resistant comm<sup>n</sup> protocols.

(b) Caching OR Distributed Cache:—

A Distributed cache is a system that pools together the RAM of multiple networked computers into a single in memory data store used as a data cache to provide fast access to data. While most caches are traditionally in one physical server or h/w component, a distributed cache can grow beyond the memory limits of a single computer by linking together multiple computers - referred to as a distributed architecture or a distributed cluster - for larger capacity and increased processing power.

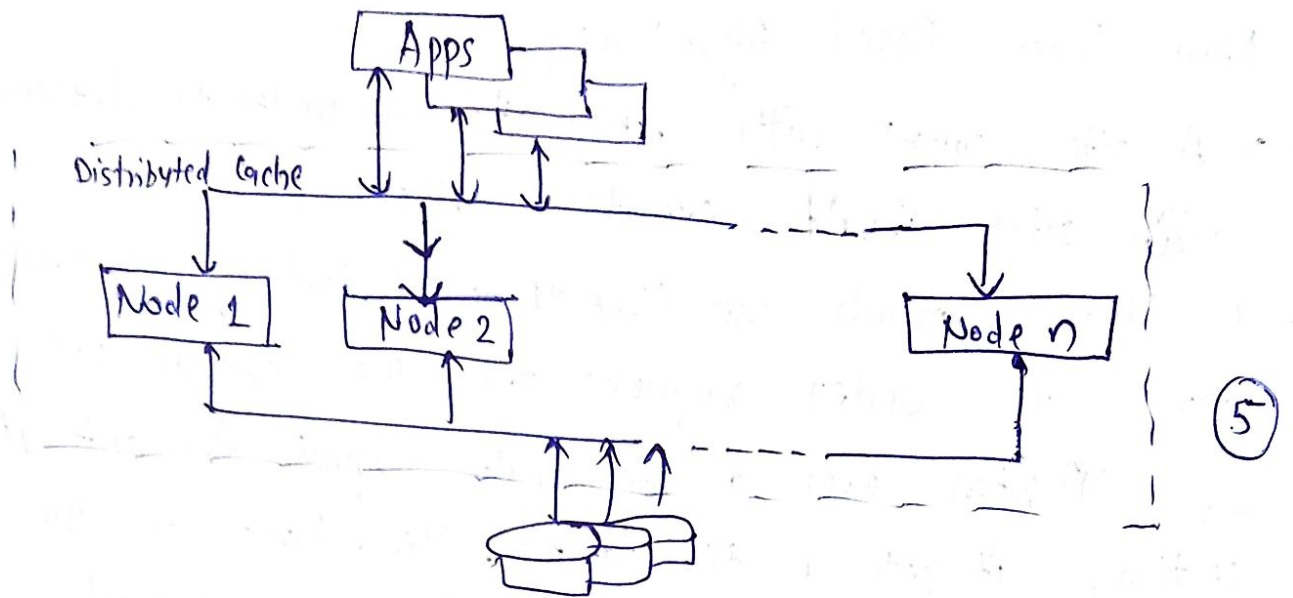


fig: Caching

- By storing the most frequently accessed data in a distributed cache, you can dramatically reduce the I/O bottleneck of disk-based systems.

### Section - (B)

#### (B) Token-Based Algorithm :-

- A unique token is shared among all the sites.
- If a site possesses the unique token, it is allowed to enter its critical section.
- This approach uses sequence no. to order requests for the critical section.
- Each request for critical section contains a sequence no. This sequence no. is used current requests.

Eg - Suzuki-Kasumi's Broadcast Algorithm



## Non-Token Based Algorithm:-

- A site comm<sup>n</sup> with other sites in order to determine which sites should execute critical section next. (6)
- This approach use timestamps instead of sequence no. to order requests for the critical section.
- When ever a site make request for critical section, it gets a timestamp. Timestamp is also used to resolve any conflict b/w critical section requests.

Eg. - ~~CSMA~~ Lamport Algorithm.

## (f) Distributed Deadlock Handling:-

Transaction location & transaction control are the 2 deadlock handling concerns in DS.

- ① Transaction location:- The data items from multiple sites are used & transactions are processed in multiple sites in a Distributed database systems.  
All the sites does not have a uniform distribution of the data processing amount with varying time period.

- ② Transaction Control:- The sites that are essential for transaction processing in DS are designated and controlled by transaction control.  
→ The choice of transaction processing and designing the centre of control has one server selected as the centre of control which travels from one server to another.

Difference:- In a centralised system, there is one deadlock detection & in DS, there can be more than one deadlock detections. A deadlock detection can find deadlock for the sites under its control.

(7)

(e) Agreement protocol is used to ensure the DS is able to achieve the common goal even after occurrence of various failures in DS. There are some standard agreement in DC.

Byzantine Problem in DS : It is a primitive to the other 2 problems. So, the other two agreement problems are:-

- Consensus Problem
- Interactive Consistency Problem.

An arbitrarily chosen processor, called the source processor, broadcasts its initial value to all other processors.

Agreement:- All non-faulty processors should agree on same value.

Validity:- If the source processor is non-faulty, then common agreed upon value by non-faulty processors must be same as the initial value of the source.



(d) Network file system (NFS) is a distributed file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to access files over a computer network much like local storage is accessed. So This ~~is~~ is a Sun NFS. (8)

CODA file system:- It is developed as a research project at Carnegie Mellon

university since 1987 under the direction of Mahadev. CODA has several features like

- freely available
- high performance
- server replication
- security
- Network Bandwidth Adaptation
- Good Scalability

CORBA :- Common Object Request Broker architecture is a standard architecture for distributed object systems. It allows a distributed, heterogeneous collection of objects to interoperate.

It describes a messaging mechanism by which objects distributed over a network can comm<sup>n</sup> with each other irrespective of a platform & language used to develop those objects. The object that requires the services of other objects is called the client.