Distributed System. Unit-1.

Q1)- What is distributed System. Explain the characteristics of distributed System.

A distributed system is a software system in which components located on networked computers ~~componen~~ communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal.

A distributed system consists of a collection of autonomous computers, connected through a network and distribution middleware, which enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility.

Characteristics :-

1). Resource Sharing → Ability to use any hardware software or data anywhere in the system.

2). Openness → Openness is concerned with extensions and improvements of distributed systems.

3). Concurrency → Components in distributed systems are executed in concurrent processes. Components access and update shared resources.

4). Scalability → (i) Accomodate more user
(ii) Respond faster.

5). Fault Tolerance → DS must maintain availability even at low levels of hardware software, network reliability.

6). Transparency → DS should be perceived by users and application programmers as a whole rather than as a collection of cooperating components.

Q2). What is distributed ~~system~~ mutual exclusion? How it can be classified?

Mutual exclusion is a concurrency control property which is introduced to prevent race conditions. It is the requirement that a process can not enter its critical section while another concurrent process is currently present or executing in its critical section i.e. only one process is allowed to execute the critical section at any given instance of time.

Classification:

1). Mutual exclusion in single computer system:-

In single computer system, memory and other resources are shared between different processes. The status of shared resources and the status of users is easily available in the shared memory so with the help of shared variable mutual exclusion

problem can be solved.

2). Mutual exclusion in distributed system:

In distributed systems, we ~~require~~ neither have shared memory nor a common physical clock and therefore we can not solve mutual exclusion problem using shared variables. To eliminate the mutual exclusion problem in distributed system approach based on message passing is used.

Q3). Describe the system model in detail.

Distributed system models are as follows:-

1). Architectural model:-

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

a). Client - Server model:-

The system is structured as a set of processes, called servers, that offer services to the users called clients. This model is usually based on a simple request/reply protocol, implemented with send/receive primitives or using remote procedure calls or remote method invocation.

b). Peer - to - Peer

All processes play similar role. Processes interact without particular distinction between clients and servers.

2). Interaction Model:

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

a) These are <u>synchronous distributed systems.</u>

→ Lower and upper bounds on execution time of processes can be set.

→ Transmitted messages are received within a known bounded time.

→ Drift rates between local clocks have a known bound.

b). Other type is <u>asynchronous distributed systems.</u>

→ No bound on process execution time.

→ No bound on message transmission delay.

→ No bound on drift rates between local clocks.

3). Fault Models:

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

→ Fault models are needed in order to build systems with predictable behavior in case of faults.

→ Such a system will function according to the predictions, only as long as the real faults behave as defined by the "fault model".

a). Logical clock:-

Logical clocks refers to implementing a protocol on all machines within your distributed system, so that the machines are able to maintain consistent ordering of events within some virtual timespan. A logical clock is a mechanism for capturing chronological and causal relationships in a distributed system. Distributed systems may have no physically synchronous global clock, so a logical clock allows global ordering on events from different processes in such systems.

b). Causal ordering of messages:

Causal ordering of messages is one of the four semantics of multicast communication namely unordered, totally ordered, causal, and sync-ordered communication. Multicast communication methods vary according to the message's reliability guarantee and ordering guarentee. The causal ordering of messages describes the causal relationship between a message send event and a message receive event.

c). Termination Detection:

Huang's algorithm is an algorithm for detecting termination in a distributed system. In a distributed

system, a process is either in an active state or in an idle state at any given point of time. Termination occurs when all of the processes becomes idle and there are no any in transit computational message.

d). Token and non-token based algorithm:

Token based :-

→ In the token based algorithm, a unique token is shared among all the sites in distributed computing systems.

→ A site is allowed to enter the critical section if it possesses the token.

Non-token based :-

→ In non-token based algorithm, there is no token even not any concept of sharing token for access.

→ Here, two or more successive rounds of messages are exchanged between sites to determine which site is to enter the critical section next.

Q.6). what are the requirements of distributed mutual exclusion theorem? Also discuss the performance metrics of it.

→ Requirements of Mutual exclusion theorem :-

i). No Deadlock → Two or more site should not endlessly wait for any message that

will never arrive.

2). No Starvation → Every site who wants to execute critical section should get an opportunity to execute it in finite time. Any site should not wait indefinitely to execute critical section while other site are repeatedly executing critical section.

3). Fairness → Each site should get a fair chance to execute critical section. Any request to execute critical section must be executed in the order they are made i.e. critical section execution requests should be executed in the order of their arrival in the system.

4). Fault Tolerance → In case of failure, it should be able to recognize it by itself in order to continue functioning without any disruption.

→ Performance metrics for mutual exclusion:

① . Response time → The interval of time when a request waits for the end of its critical section execution after its ~~critical section~~ solicitation message have been conveyed.

② . Synchronization Delay → The time required for the next process to enter the critical section after a process leaves the critical section.

3). Message complexity → The number of messages required to execute each critical section by the process.

4). Throughput → Throughput is the amount at which the system executes requests for the critical section.

5). Low and High load performance → The amount of request that arrives for critical section execution denotes the load.