Date:

Q1. What is distributed deadlock detection? What are the issues in deadlock detection?

Ans 1. Distributed Deadlock Detection → Distributed deadlocks can occur when distributed transactions or concurrency control are utilized in distributed systems. It may be identified via a distributed technique like edge chasing or by creating a global wait-for-graph (WFG) from local wait-for-graphs at a deadlock detector. Phantom deadlocks are identified in a distributed system but do not exist due to internal system delays.

In a distributed system, deadlock can't be prevented nor avoided because the system is too vast. As a result, only deadlock detection is possible.

Requirements for deadlock detection -

1.) Progress
2.) Safety

1.) Progress → The method may detect all the deadlocks in the system.

2.) Safety → The approach must be capable of detecting all system deadlocks.

Approaches to detect deadlock in the distributed Sy →

a.) Centralized Approach → Only one resource is responsible for detecting deadlock in the centralized method, and it is simple and easy to use. Still, the disadvantages include excessive workload on a single node and single-point failure, making the system less reliable.

b.) Hierarchical Approach - It is the integration of both centralized and distributed approaches to deadlock detection. In this strategy, a single node handles a set of selected nodes or clusters of nodes that are in charge of deadlock detection.

c.) Distributed Approach - In the distributed technique, various nodes work to detect deadlocks. There is no single point of failure as the workload is equally spread among

all nodes.

Issues of Deadlock detection →

1.) Deadlock detection - based deadlock handling requires addressing two fundamental issues: first, detecting existing deadlocks, and second, resolving detected deadlocks.

2.) Detecting deadlocks entails tackling two issues: WFGs maintenance and searching the WFGs for the presence of cycles.

3.) In a distributed system, a cycle may include multiple sites. The search for cycles is highly dependent on the system's WFGs as represented across the system.

Q.2. Explain the deadlock prevention techniques with example.

Ans. Deadlock prevention Techniques -

A Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for a resource that is held by some other process.

There are four necessary conditions for a deadlock to happen which are-

- Mutual Exclusion
- Hold & Wait
- No preemption
- Circular Wait

So, the above four conditions are necessary for a deadlock to occur, if any one of the above four conditions is prevented, we can prevent a Deadlock to occur. There are 2 ways to prevent deadlock in a distributed system.

→ Ordered Request
→ Collective Request

a.) Ordered Request → In this Deadlock Prevention method, each resource type is assigned a certain level to maintain a resource request policy for a process. This is known as the Resource Allocation policy. For each Resource, a global level number is assigned to impose ordering of all resource types. While requesting for a resource, a Process has to make sure that it does not request for a resource whose level order is lower than the

highest - level order resource it currently holds. It can only request resources higher than the highest level resources held by the process.

Ex → There are 10 resources from level 1 to 10, and 10 is the highest level order resource. If a Process currently has resources 5 and 8, it can't request a resource below 8, it can only request resources 9 and 10.

This Method makes sure that the circular wait condition is not reached and if one of the deadlock conditions is denied, the deadlock will be prevented.

b.) Collective Request → This method prevents the Hold and Wait for condition by using any of the following Resource Allocation Policies-

- This Resource Allocation Policy ensures that a Process requests for all the required resources before the execution of the process. If any of the required resources are not available, the request is not granted.

- Here, while requesting any resource, the process should not hold any resources.

Q.3 Write short notes on

a.) Path pushing Algorithm → Path-pushing algorithms detect distributed deadlocks by keeping an explicit global WFGr. The main concept is to create a global WFGr for each distributed system site. When a site in this class of algorithms performs a deadlock computation, it sends its local WFGr to all neighboring sites. The term path-pushing algorithm was led to feature the sending around the paths of global WFGr.

b.) Edge-chasing Algorithm → An edge-chasing method verifies a cycle in a distributed graph structure by sending special messages called probes along the graph's edges. These probing messages are distinct from request and response messages. If a site receives the matching probe that it previously transmitted, it can cancel the formation of the cycle.

Q4. What do you understand by agreement protocol? Also discuss its application.

Ans4. When the system is free from failures, an agreement can easily be reached among the processors. However, when the system is prone to failure, this method does not work. This is because faulty processors can send conflicting values to other processors preventing them from reaching an agreement. In the presence of faults, processors must exchange their values with other processors and relay the values received from other processors several times to isolate the effects of faulty processors. A processor refines its value as it learns of the values of other processors. This entire process of reaching an agreement is called an agreement protocol.

Applications of Agreement Protocol →

1.) Fault-Tolerant clock Synchronization

a.) Distributed Systems require physical clocks to synchronized.
b.) Physical clocks have drift problem.
c.) Agreement Protocols may help to reach a common clock value.

b.) Atomic Commit in DDBS

a.) DDBS sites must agree whether to commit or abort the transactions.

b.) Agreement protocols may help to reach a consensus.

Qs. How can we classified agreement protocol? Explain in detail.

Ans. There are three types of classification in agreement protocol -

a.) The Byzantine Agreement Problem
b.) The consensus problem
c.) The interactive consistency problem

a.) The Byzantine Agreement Problem → In the Byzantine agreement problem, an arbitarily chosen processor, called the source processor, broadcasts its initial value to all other processors. A solution to the Byzantine agreement problem should meet the following objectives -

i.) Agreement - All non faulty processors agree on the same value.

2.) Validity → If the source processor is nonfaulty, then the common agreed upon value by all nonfaulty processors should be the initial value of the source.

b.) The Consensus Problem → In the consensus problem, every processor broadcasts its initial value to all other processors. Initial values of the processors may be different. A protocol for reaching consensus should meet the following conditions.

1.) Agreement → All nonfaulty processor agree on the same single value.

2.) Validity → If the initial value of every nonfaulty processor is v, then the agreed upon common value by all non faulty processor must be v.

C.) The Interactive Consistency Problem → In the interactive consistency problem, every processor broadcasts its initial value to all other processors. The initial value of the processors may be different. following conditions must satisfy —

Page:
Date:

1.) Agreement → All nonfaulty processors agree on the same vector.

2.) Validity → If the i^{th} processor is nonfaulty and its initial value is v_i, then the i^{th} value to be agreed on by all nonfaulty processors must be v_i.