Bh.IMPs - DC | Computers Sem 8 | Final Sem Exams 2023

1. Explain the Goals/Features of Distributed Systems.

2. What is the role of Middleware in a Distributed System and mention the services provided by it.

3. Explain the RPC model and its types.

4. Difference between Message and Stream-Oriented Communication.

5. Why is clock synchronization necessary in distributed systems?

6. Explain Rica-Agarwala’s non-token-based algorithm.

7. Write a short note on- Suzuki-Kasami’s Broadcast Algorithm.

8. What is Process Migration? Explain the methods used for this.

9. Explain the Fault Tolerance Mechanism.

10. What is consistency? Compare Eventual and Strong-Consistency in Distributed Databases.

11. What Are The Features of DFS. Explain Anyone in detail.

Module 3

Task Assignment Approach Sum (Definitely going to come)

**3. Synchronization**

Mutual Exclusion

A condition in which there is a set of processes, only one of which is able to access a given resource or perform a given function at any time

REQUIREMENTS

1. Safety of property

At most one process may execute in the critical region (CR) at a time.

2. Liveness property

A process requesting entry to the CR is eventually granted it. There should not be deadlock and starvation

3. Fairness

Each process should get a fair chance to execute the CR.

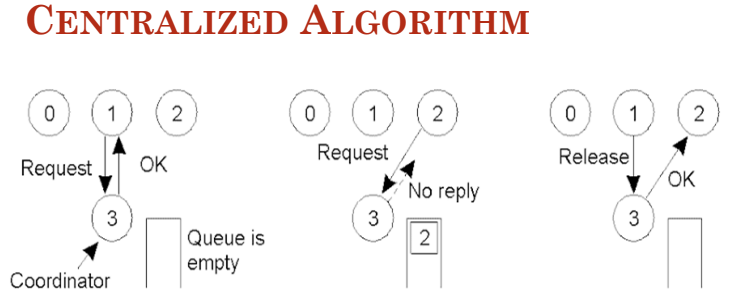
Performance Metrics

1. Message complexity: Number of messages that are required per CR execution by a process.

2. Synchronization delay: Time interval between critical region (CR) exit and new entry by any process.

3. System throughput: Rate at which requests for the CR get executed.

4. Response time: Time interval from a request send to its CR execution completed.

****

PERFORMANCE PARAMETERS

1. The algorithm is simple and fair as it handles the request in sequential order.

2. It guarantees no starvation.

3. It uses three messages REQUEST, REPLY, and RELEASE.

4. It has a single point of failure.

5. Coordinator selection could increase synchronization delay, especially at times of frequent failures.

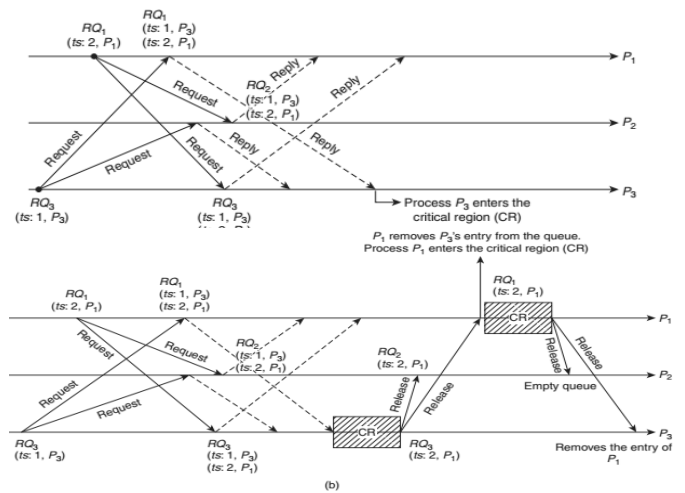
**Distributed mutual exclusion algorithms**

Non-token-based approach:

● Two or more successive rounds of messages are exchanged among the sites to determine which

the site will enter the CS next.

• Lamport’s Distributed Mutual Algorithm



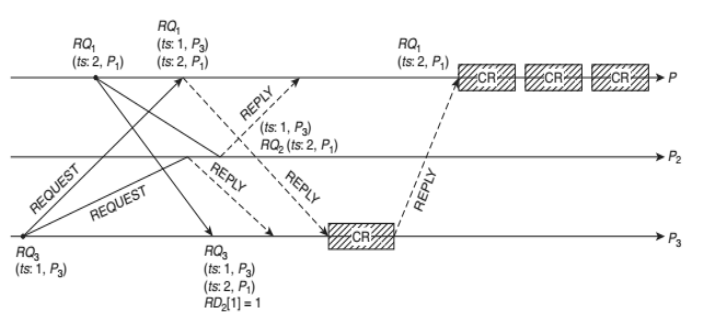
PERFORMANCE PARAMETERS

• Lamport’s algorithm has message overhead of total 3(N − 1) messages: N – 1 REQUEST messages to All process (N minus itself), N −1 REPLY messages, and N −1 RELEASE messages per CR invocation.

• The synchronization delay is T. Throughput is 1/(T + E).

• The algorithm has been proven to be fair and correct. It can also be optimized by reducing the number of RELEASE messages sent.

• Ricart–Agrawala Algo



PERFORMANCE PARAMETERS

1.The algorithm does not use explicit RELEASE message. The dequeuing is done on the receipt of REPLY itself. Thus, total message overhead would be 2(N − 1) messages, that is, for entering a CR, (N − 1) requests and exiting (N − 1) replies.(Improvement over lamport’s algo)

2. The failure of any process almost halts the algorithm (recovery measures are needed) as it requires all replies.

3.Single point of failure is replaced by multiple point of failure.

• Maekawa’s Algorithm

Maekawa’s algorithm is a quorum or voting-based mutual exclusion algorithm.

It suggests that a process Pi does not require to request all processes, but only to a subset of processes

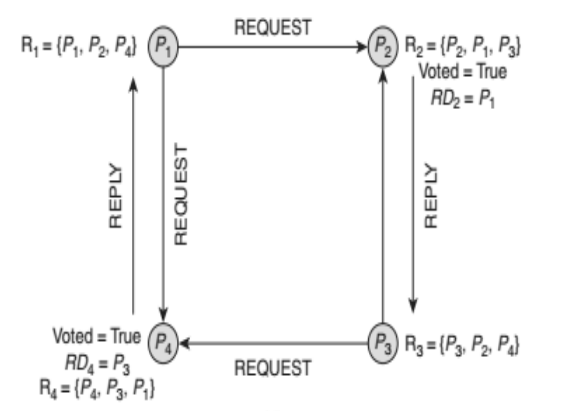
(the quorum) called Ri.

Each process Pi in the quorum set Ri gives permission to at most one process at a time.

PROBLEM OF DEADLOCKS

Though Maekawa’s algorithm has been proven to be correct and safe, but the property of liveness is not satisfied by it because it can lead to deadlock

Maekawa’s algorithm can deadlock because a site is exclusively locked by other sites and requests are not prioritized by their timestamps . Thus, a site may send a REPLY message to a site and later force a higher priority request from another site to wait.



PERFORMANCE PARAMETERS

Size of a request set is √ N.(QUORUM SIZE)

1. An execution of the CR requires √N REQUEST, √ N REPLY and √ N RELEASE messages, thus requiring total 3 √N messages per CR execution.

2. Synchronization delay is 2T.

3. M = K = √N works best.

• Token-based algorithm:

Token-based approach:

● A unique token is shared among the sites.

● A site is allowed to enter its CS if it possesses the token.

● Mutual exclusion is ensured because the token is unique.

• Suzuki–Kasami’s Broadcast Algorithm

1. It is simple and efficient.

2. The algorithm requires at most N messages to obtain the token to enter CR.

3. The synchronization delay in this algorithm is 0 or T(message delay). Zero synchronization delay, if the process already holds the token or message delay

• Singhal’s Heuristic Algorithm

The number of REQUEST messages can vary from N/2

(Average value of the identifier) to N (max value).

• Raymond’s Tree-Based Algorithm

The algorithm exchanges only O(log N) messages under light load and four messages under heavy load to execute the CR, where N is the number of nodes in the network.

Comparative Performance Analysis.

**4 Resource and Process Management**

4.1 Desirable Features of a Global Scheduling Algorithm,

**Task assignment approach,**

Load balancing approach,

load sharing approach

4.2 Introduction to process management, process migration, Threads, Clients, Servers,

Code Migration

**5 Consistency, Replication, and Fault Tolerance**

Introduction to replication and consistency,

DATA-CENTRIC CONSISTENCY MODELS

● Strict Consistency

● Linearizability and Sequential Consistency

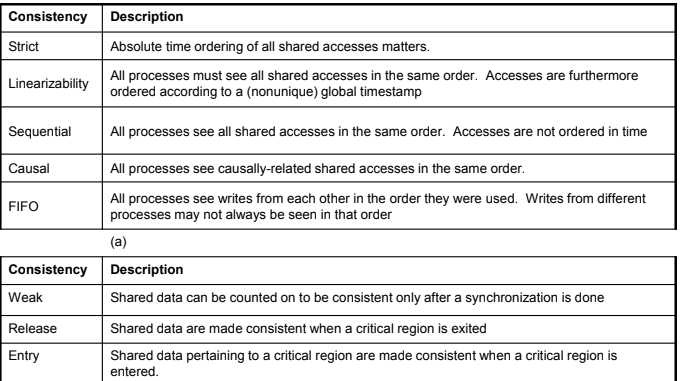
● Causal Consistency

● FIFO Consistency

● Weak Consistency

● Release Consistency

● Entry Consistency



Four models in client-centric consistency:

Monotonic Read Consistency

Monotonic Write Consistency

Read-your-writes Consistency

Writes-follows-reads Consistency