Suzuki Kasami Broadcast Algorithm

Need and Importace of Algorithm
In a distributive database, data is stored across multiple nodes. Only one process at a time should access or update a critical resource such as: A shared ~~ba~~ database suilable for system who read/write operate on shared database must be synchronized without using a centralized Coordinator.

In Dirstributive System

- Multiple node may attempt simultaneous updates
- Concurrency can cause inconsistancy
- mutual exclusion ensures seriability

Suzuki hazami iprove over permission based algoritm by.

- Reducing message capacity
- concurrency can cause inconsistany
- mutual exclusion ensures seriability
- Eleminating deadlock
- Abolting Starvation.

Data structure used => Request Number (RN) Array
RN[i] = largest request number from P_i to this Process.

Token (unique, only one exist)

- LN[i]: last request number from process P_i that has been satisfied.
- Q.: FIFO queue of repeting process.

Algorithm Opeation

Step ①: Requesting the critical section, when Process P_i wants to enter the CS - Invents it's request number

$RN[i]++;$

- Broadcast REQUEST (i, $RN[i]$ to all other)

Step ②: When Process P_j recives a request (i, n)

$RN[i] = MAX(RN[i], n)$

Then P_i is eligible → token may be sent.

Step ③: Token Transfer:

If the token header is not in CS
Find the pending request

then,

① Add requesting Process to token queue.

② Send token to first Process in Q.

Step ④ Entering the critical section if P_i recive token it Enters

Step ⑤: Releasing the critical section

After existing CS:

① Update: $LN[i] = RN[i]$

② Checking for other pending request

if $RN[j] = LN[j]+1$ → add j to Q

③ if queue not empty → Pass token.

Mutual Exclusion: Only token holder can enter CS

Deadlock freedom: Token circulation ensures progress

Starvation freedom: FIFO queue ensures fairness

Message complexity

Request: N-1 message (broadcast)

Token transfer: 1 message

Total per CS entry: O(n)

Application in Dirstibutive system

- Dirstibuted transaction managment
- Replicated database updates.
- Dirstibuted file system
- Shared resouce coordination.

Advantages

No deadlock

No starvation

Fair access (FIFO)

lover of message overhead than permission based algorithm.

Efficient CS access is frequent

Disadvantage

- token loss cause failure.
- Not fault tolerant

- Broadcast cost is high for large system
- Assume reliable communication

Suzuki- kazami broadcast Algorithm provides an efficient, fair and deadlock free sol^n for mutual exclusion in distributed ~~but~~ database. It's token based approach reduces communication overhead and ensures strict access control making it ideal for medium scale distributed system where reliability is high.