Ricart–Agrawala algorithm

From Wikipedia, the free encyclopedia
(Redirected from Ricart-Agrawala Algorithm)

> This article **does not** **cite** any **references or sources**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(December 2009)*

The **Ricart-Agrawala Algorithm** is an algorithm for mutual exclusion on a distributed system. This algorithm is an extension and optimization of Lamport's Distributed Mutual Exclusion Algorithm, by removing the need for $release$ messages. It was developed by Glenn Ricart and Ashok Agrawala.

Algorithm [edit]

Terminology [edit]

- A *site* is any computing device which is running the Ricart-Agrawala Algorithm
- The *requesting site* is the site which is requesting entry into the critical section.
- The *receiving site* is every other site which is receiving the request from the requesting site.

Algorithm [edit]

Requesting Site

- Sends a message to all sites. This message includes the site's name, and the current timestamp of the system according to its logical clock (*which is assumed to be synchronized with the other sites*)

Receiving Site

- Upon reception of a request message, immediately send a timestamped *reply* message if and only if:
  - the receiving process is not currently interested in the critical section OR
  - the receiving process has a lower priority (*usually this means having a later timestamp*)
- Otherwise, the receiving process will defer the reply message. This means that a reply will be sent only after the receiving process has finished using the critical section itself.

Critical Section:

- Requesting site enters its critical section only after receiving all reply messages.
- Upon exiting the critical section, the site sends all deferred reply messages.

Performance [edit]

- Number of network messages; 2*(N-1)
- Synchronization Delays: One message propagation delay

Common Optimizations [edit]

Once site P_i has received a $reply$ message from site P_j, site P_i may enter the critical section multiple times without receiving permission from P_j on subsequent attempts up to the moment when P_i has sent a $reply$ message to P_j. This is called Roucairol-Carvalho optimization or Roucairol-Carvalho algorithm.

Problems [edit]

One of the problems in this algorithm is failure of a node. In such a situation a process may starve forever. This problem can be solved by detecting failure of nodes after some timeout.

See also [edit]

- Lamport's Bakery Algorithm
- Lamport's Distributed Mutual Exclusion Algorithm
- Maekawa's Algorithm
- Suzuki-Kasami algorithm
- Raymond's Algorithm
- Naimi-Trehel's Algorithm

References [edit]

- Maekawa, M.,Oldehoeft, A.,Oldehoeft, R.(1987). Operating Systems: Advanced Concept.Benjamin/Cummings Publishing Company, Inc.

Categories: Distributed algorithms