Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 6; April-June, 2015 pp. 500-502

http://www.krishisanskriti.org/acsit.html

Centralized Approach to obtain Mutual Exclusion in Multi Process Environment

Ashish Chauhan[1], Purvi Bansal[2], Swati Tyagi[3] and Shobhit Saxena[4]

[1]M.Tech Scholar GRD Institute of Management & Technology Dehradun
[2,3,4]B.Tech Scholar, Meerut Institute of Technology Meerut
E-mail: [1]chauhan.mrt@gmail.com, [2]purvibansal87@gmail.com,
[3]soni.palak0@gmail.com, [4]shobhitsaxena2012@gmail.com

Abstract—*Distributed System is a system in which hardware or software components are located at network , computer communicate and coordinate their action only by passing messages.Mutual exclusion is a mechanism in which concurrent access to shared resources by several user request is serialized to secure the integrity of the shared resources.As the number of message exchanged is very crucial parameter for analysing the performance of any mutual exclusion algorithm the proposed algorithm reduces the number of messages exchangedless than 2*(N-1),where N is the number of sites in the system. The number of messages required to allow any site to enter critical section is reduced by using the centralized body.The centralized body has the centralised control over the entire system. Execution of sites in critical section will depend upon the value of their increasing order of their timestamp. All the sites directly communicate with the centralized and the entire algorithm executed as per directions of centralised body.*

Keywords: *Distributed System, Critical Section, Token, Non-Token, Mutual Exclusion*

1. INTRODUCTION

A Distributed system is a system in which hardware and software are located at network computers which communicate and co-ordinate their actions only by passing messages. Distributed system has a limitation of Shared Memory and absences of Global Clock. Distributed System has an interesting problem of Mutual Exclusion.

Mutual Exclusion is a mechanism in which concurrent access to Shared resources by several user, requests is serialised to secure the integrity of the Shared Resources. Mutual Exclusion requires that actions performed by the user on shared resources must be atomic. The problem of Mutual Exclusion frequently arises in Distributed System whenever concurrent access to shared resources by several sites is involved. To maintain the consistency of a system, it is necessary that the shared resources will be accessed by single user at a time.

Distributed Mutual Exclusion algorithm can be classified into two types -

1. **Token based algorithm-** In Token based algorithm a unique token is shared among all the sites and a site is allowed to enter its Critical Section only if it posses the valid token. Token based algorithm uses Sequences no, every time a site makes a request for a token, it increments its sequences no counter and merge it with request messages. Example of Token based are Raymond‘s algorithm (log (N) messages) and Suzuki Kasami Algorithm (N messages).
2. **Non-Token based algorithm-**In Non-Token based algorithm a site communicate with set of others site to decide who should execute critical section next. This algorithmuses the Timestamp to order the request for Critical section in order to avoid the conflict between sites. Example of Non-Token based are Lamport`s algorithm (3(N-1) messages) and Ricart-Agrawala`s algorithm (2(N-1) messages).

2. PERFORMANCE METRICS OF MUTUAL EXCLUSION ALGORITHM

2.1. Response Time

It is the time interval a request waits for its Critical Section Execution to be over after its Requested messages have been sendingout. Smaller the Response Time betters the performance.

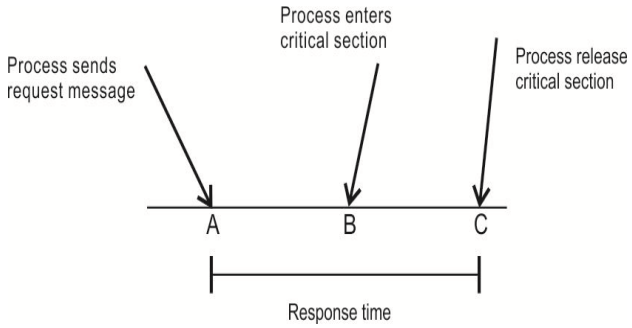

Fig. 1

2.2. Synchronization Delay

The Time interval after the site exits the Critical section and before the next site enters the Criticalsection is known as Synchronization Delay. Smaller the Synchronization delaysbetter the Performances.

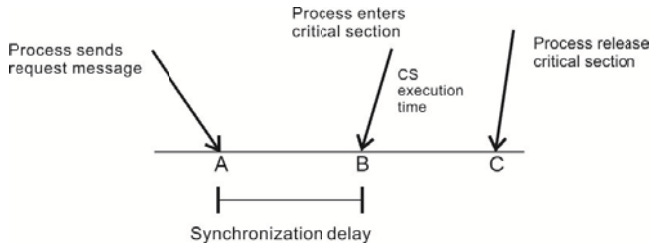

Fig. 2

2.3. No. of Messages necessary for Critical section invocation

Before entering in the Critical section, site sends a request messages to all the sites and after the execution of Critical section site sends a reply messages to all other sites. Lesser the No. of Messages necessary for Critical section invocation better the Performances.

2.4. System Throughput

The rate at which system executes the request for Critical Section is known as System Throughput. Maximum the System Throughput Better the Performances.

System Throughput= 1/ (SD + E)

Where SD is Synchronization Delay and E is Average Critical Section Execution Time.

3. PROPOSED WORK

3.1. System model

The centralized body is used which controls the overall functioning of the system.All the sites communicate with the centralized body.Centralized body accept the request of sites and allow them to enter into the critical section according to their timestamp.

There are m sites (where m<n) m denotes no of sites wants to enter in Critical Section and n denotes total no of sites.

Two queues are associated with centralized body
1. One for storing the IDof the sites.
2. Another for storing the timestamp in increasing order of the respective site.

At any time site may have several request for critical section. A site queues upthis request and serves them one at a time.
A site can be in one of the three states-
1. Requesting Critical Section:-the site is blocked and cannot make further requests for Critical section.
2. Idle:-In idle state site is executing outside its critical section (no conflict).
3. Executing:-the site is executing in its critical section.

3.2. Algorithm

3.2.1. Requesting

1. The requesting sites sends the request message in the form [TS_i,i] to centralized body where TS_i is timestamp and i is the id.
2. When centralised body receive the request messages it performs the following task-
   When the sites request, their timestamp is stored in timestamp queue in increasing order and accordingly their id's are also stored in id queue.

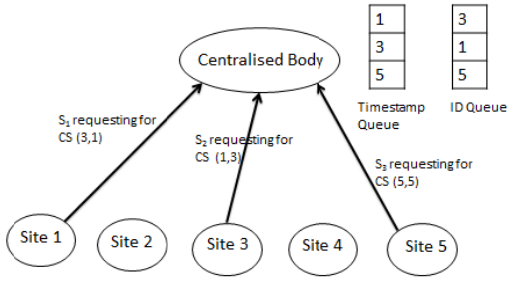

Fig. 3

3.2.2. Executing

Centralised body sends a ENTER message to id pointed by top entry of id queue and remove the top entry from id queue. Centralised body become idle for the time period pointed by the top entry of the timestamp queue after that it sends ENTER message to next site.

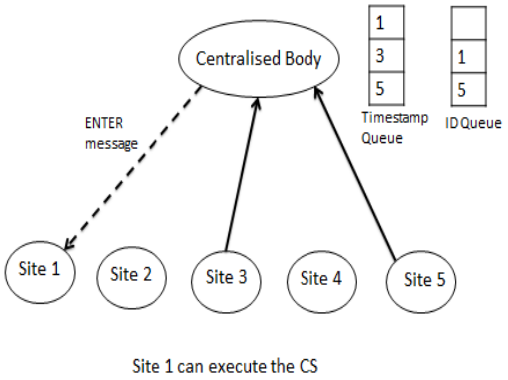

Fig. 4

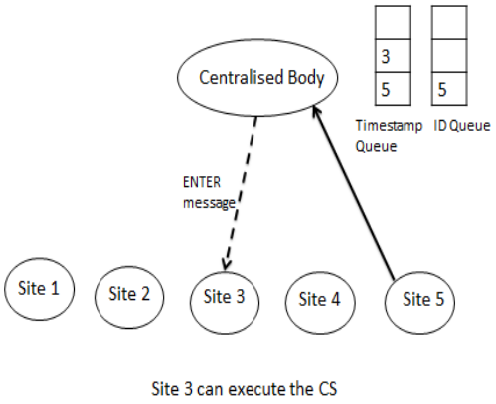

Fig. 5

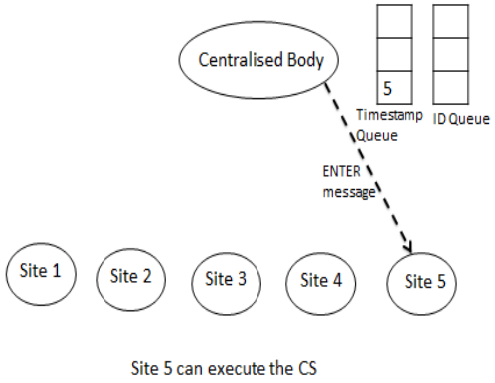

Fig. 6

3.2.3. Releasing

After executing the Critical Section for the time period pointed by the top entry of timestamp queue, site release the Critical Section and removes the top entry of the timestamp queue.

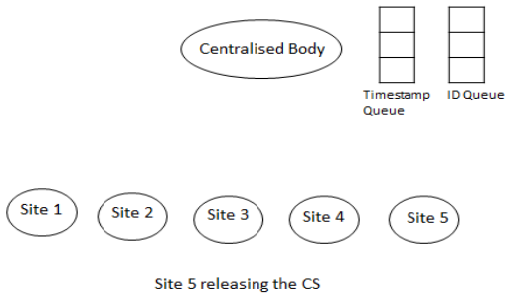

Fig. 7

3.3. Implementation

The proposed algorithm is demonstrated for 7 sites out of which 5 sites want to enter in the Critical Section. Here S_1, S3, S5, S6, S7 request for the Critical Section execution to the Centralized Body.

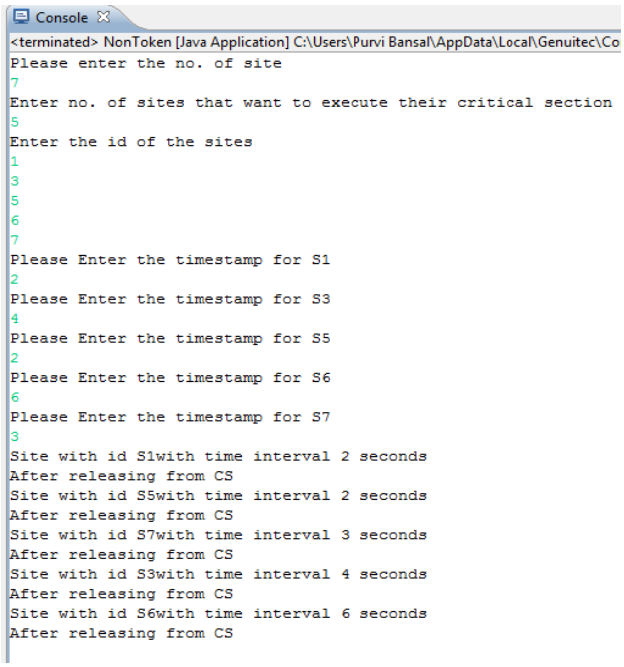

Fig. 8

4. CONCLUSION

This algorithm implements the Mutual Exclusion in a system. The modification is made by using the advantage of Centralized body. The Centralized body has the whole control over the system.Through Centralized body the number of messages for the Critical Section invocation is reduced to2m. All The sites executes in an increasing order of their timestamp. Sites only communicate by passing messages to the centralized body and we see that the number of messages are reduced as there is no node to node communication take place.No algorithm uses fewer messages,operates faster and centralized control.

REFERENCE

[1] G. Ricart and A. K. Agrawala, &ldquo, "An Optimal Algorithm for Mutual Exclusion in Computer Networks &rdquo, *Comm".ACM*, vol. 24, no. 1, pp. 9-17, Jan. 1981.

[2] J.-H. Yang and J. Anderson, &ldquo,Time Bounds for Mutual Exclusion and Related Problems &rdquo, *Proc. 26th Ann. ACM Symp. "Theory of Computing*, pp. 224-233," May 1994.

[3] D.Agrawal, A.El. Abbadi, *"An efficient and fault tolerant solution for distributed mutual exclusion"*ACM Transaction on Computer Systems 9 (1) (1991) 1 – 20.

[4] Moharram Challenger, PeymanBayat&Mohammad Reza Meybodi (2006)*"A reliable optimization on distributed mutual exclusion algorithm",IEEE Conference: Testbeds and*Research Infrastructures for the development of networks and communities - *TRIDENTCOM* pp.

[5] Ricart, G., and Agrawala, A.K.:(Jan. 1981) *"An Optimal Algorithm for Mutual Exclusion inComputer Networks", Commun.of the ACM*,vol. 24, pp. 9-17

[6] B.A. Sanders "*The information structure of distributed mutual exclusion algorithms*", ACM Transactions on ComputerSystems 5 (3) (1987) 284–299.

[7] R. Cypher "*The communication requirements of mutual exclusion"*. In Proceedings of the 17th Annual Symposium on Parallel Algorithms and Architectures, pp.147{156, 1995.