ISSN 2277-3061

An Executive Approach to Achieve Mutual Exclusion in Distributed Data using Topology and Association Rule

Anurag Singh
Research Scholar BBDU Lucknow
anuragphd001@gmail.com
Amod Tiwari
Director Bhabha Institute of Technology aunahan, Kanpur
amodtiwari@gmail.com

ABSTRACT

In this paper, a new approach is being proposed to achieve mutual exclusion in distributed system using computer network and topology of n^{th} nodes. In this executive approach nodes communicate among themselves using message passing technique. In this executive approach, distributed system with n nodes is logically partitioned into number of sub distributed system having only $m^{½}$ nodes, where m is obtained by adding a minimum number in n to make it next perfect square number only if n is not a perfect square. Proposed algorithm is a Token based approach and achieves token optimally in 2 messages only for the best case and in worst case a node achieves token in n messages only.

Keywords: Critical section; Data Association; Distributed algorithm; Message passing; Message complexity.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATION JOURNAL OF COMPUTERS AND TECHNOLOGY

Vol. 13, No. 6

editorijctonline@gmail.com

www.cirworld.org/journals

ISSN 2277-3061

Introduction

Solution to the distributed mutual exclusion problem consists of a protocol to be executed among the processes of the distributed system solely by passing messages in order to allow one or some processes to execute private operations with one or several shared resources. In a centrally controlled system, it is not too difficult to implement the mutual exclusion on the shared object. Semaphores and monitors are commonly used. However, in a distributed environment, the solution to this problem becomes far more complex due to the absence of a global or centralised controller. In a distributed system, nodes communicate only by passing messages.

A distributed mutual exclusion algorithm requires an approach such that if a node wishes to enter in mutual exclusion then all other nodes must be aware of this that a process has already entered in critical section, and hence they themselves cannot enter into their critical section. There are number of techniques available for this. In centralised system there is the problem of congestion because only one administrator is responsible to manage the complete network but it's simple to implement. In distributed system message complexity is very high because no node has the information about the availability of the token, so the node that is wishing to enter into critical section has to send the request messages to all the nodes that are the part of the system in search of the token. Proposed algorithm behaves very balanced even if the system is heavily loaded. The rest of the paper is organized as follows. Section 2 presents a previous work. In Section 3 we examine the basis of the algorithm followed by a more formal description of the algorithm. Finally, in Section 4 we conclude the paper.

Literature Survey

Distributed mutual exclusion algorithms is token based algorithms [1], [2] where a unique token (also known as the PRIVILEGE message [2]) is shared among the sites such that possession of the token gives a site the authority to execute its CS. Singular existence of the token implies the enforcement of mutual exclusion in distributed systems.

Suzuki-Kasami (1985)

Suzuki Kasami's Algorithm requires 0 or at most N number of messages to enter into critical section [2]. A node having the token is allowed to enter into the critical section. A single node has the privilege and a node requesting critical section, broadcasts a message to all the other nodes. A site sends the privilege if the token is idle with the site. The site having token can continuously enter critical section until it sends the token to some other site. The request message has the format REQUEST(j,n), which means site j is requesting its nth critical section. Each node maintains an array RN of size N for recording latest sequence number received from each of the other nodes. The PRIVILEGE message has the format PRIVILEGE (Q, LN), where Q is queue of nodes requesting critical section and LN is an array of size N where LN[j] is the latest critical section executed by a node j. If RN[j] = LN[j]+1 means a node j has sent a request for its new sequence of critical section, and the node having the privilege adds this to the queue and if token is idle sends the node sends the PRIVILEGE(LN,Q) to the node requesting critical section. Number of message per critical section entry is (N-1) REQUEST messages plus 1 PRIVILEGE message so N messages in all or 0 if the node having the token wants to enter critical section.

Kerry Raymond (1989)

In this algorithm nodes are arranged in an un-rooted tree structure [3]. All messages are sent along the undirected edges of the tree. Every node knows about the existence of its immediate neighbours. Again a PRIVILEGE message has to be received by a node to enter into critical section. At every node a variable HOLDER points to a node along the path to the PRIVILEGE. A node having the PRIVILEGE the HOLDER points to itself. When a non-privileged node wants to enter critical section it generates a request and adds it to its REQUESTQ, which is a queue maintained by each node. If it has not sent a message along the directed path towards the node pointed by the holder variable, it sends a message along the edge to the token holder. On receiving a message, the nodes forward the message to the token holder along path. However, before forwarding the nodes add the request in their REQUESTQ. When the request reaches to the node having the PRIVILEGE, then if the node is not executing the critical section, it sends the PRIVILEGE to that node from which the message is received. On receiving the PRIVILEGE if the nodes own id is on the top of the queue, it executes critical section else sends the PRIVILEGE to the node pointed by the id, and set its holder to point to that node. The number of messages required to execute critical section can be 0 or typically 2D, where D is the diameter of the tree on which the algorithm is running, however this is reduced to maximum of four messages per critical section execution under full load when the topology is proper tree and two messages when it's a chain.

Mukesh singhal (1989)

This algorithm [4] makes use of state information which is defined as the set of states of mutual exclusion processes in the system. Each site maintains information about the state of other sites and uses it to deduce a subset of sites likely to have the token. Consequently, the number of messages exchanged for a critical section invocation is a random variable between 0 and n (n is the number of sites in the system). Sites use sequence numbers to distinguish between a current token request and old delay an token request. Every site keeps a counter. When a site has to execute its CS, it increments its counter and sends the updated value (called sequence number) in token request messages. Each site keeps a record of the highest known sequence number (along with the latest known state information) of each site. By comparing the sequence number in a received message with the latest known sequence number of its sender site, the token to a requesting site with the lowest sequence number is granted.

Sebastian Cantarell, Ajoy K. Datta, Franck Petit (2001)

Sebastian Cantarell, Ajoy K. Datta, Franck Petit assume that there exist two layers in the system: the application layer (the higher layer) and the GME layer (the lower layer). The interface between the two layers is implemented [6] by using two types of messages: Request-Session and Grant-Session. When the application layer needs to access a session, says Session X, the process running the application layer sends the message Request-Session(X) to the GME layer. Eventually, the GME layer grants the application layer the access to Session X by sending the message Grant Session. The size of messages is 2 x log (m + 1) bits only. Every resource request generates O(n 2) messages in the worst case, but zero messages in the best case.

Quazi Ehsanul Kabir Mamun and Hidenori Nakazato (2006):

Quazi Ehsanul Kabir Mamun and Hidenori Nakazato have presented a new token based protocol for group mutual exclusion [5] in distributed systems. The protocol uses one single token to allow multiple processes to enter the critical section for a common session. One of the significant characteristics of the protocol is concurrency; throughput and waiting time can be regulated adjusting the time period for which a session is declared. The minimum and the maximum number of messages to enter the CS is 0 and (n + 2) respectively where n is the total number of processes in the system.

Proposed Approach

It is assumed that the network is fully connected and there are no faulty processors in the network. We partition the n nodes of the network logically into m sets of m nodes each. Each set is called a local group (LG). Nodes in a local group can communicate directly with each other for the purposes of entering the critical section. That is, all nodes in a local group are fully connected. One node from the local group is selected as the local coordinator (LC). And the local coordinators from the all groups from another group of local coordinators. This group of local coordinators is called the global group *(GG)*. One member from the global group is selected as the Global coordinator (GC). Each node of the global group can communicate with all other nodes of the global group and also with all nodes of its local group to which they belong.

Basis of the Algorithm

The proposed algorithm is token based and only one token exists in the network. Permission to enter the critical section is granted by the acceptance of the token.

1. In every local group g, there exists a local coordinator (LC) that is known to all nodes in that group G. When a node of a local group wants to enter the critical section it sends a request message to the LC.
2. LC after receiving the request for the token, it waits till the availability of the token from the local group, if the token is locally available in the group. LC forwards the token to the requesting node after receiving it from the previous process holding the token from local group.
3. Upon completion of the critical section operation it forwards the token to LC of the local group.
4. If the token is not locally available in the group then the LC will forward the request among the nodes in GG.
5. All the local coordinators from the respective groups verify the availability of the token in the group, by forwarding the request for the token and the node having the token reply the LC about it.
6. The LC that is having the idle token forwards the token to the requesting LC. And now the LC forwards the token to the requesting node.
7. Upon completion of the critical section operation it forwards the token to LC of the local group.
8. LC having the idle token verifies the status of the request queue to check any pending request.
9. If any LC finds local request then it forwards the token to it, if the request is from the one LC to another LC, then also the token is forwarded to requesting LC.
10. If the node is required by any node then it will request for it only to its local coordinator(LC)
11. Prove the equality above by showing that for any homomorphism token f and elements a, b, c, f(abc)=f(a)f(b)f(c) and $f(a^{-1})=f(a)^{-1}$.
12. Find all homeomorphisms from Z to Z and from F_2 to $Z_3 \times Z_3$.

The approach of picking where generators of a group go and then "extending" the homomorphism to the rest of the group very often comes in handy. However, this can only be done when the elements the generators are sent to satisfy all the relations between the generators themselves.

Recall that Z_3 is generated by R_{120}. Suppose we try to define a homomorphism $f:Z_3 \rightarrow Z$ by letting $f(R_{120})=1$, sending a generator to a generator. Does this extend to a homomorphism? What relation does R_{120} satisfy that $1 \in Z$ does not? There are many homeomorphisms from F_2 to $Z \times Z$. Take for instance f(a)=(1,0) and f(b)=(0,2).

If you're itching for a challenge, try to find all the homomorphisms from $Z \times Z$ to F_2. What do they have in common?

```
While (true)
        Do
{
   Select a group g € G;
    Request (g);
```

- Entry protocol.

Critical Section

Release;

Exit Protocol.

}

Suppose f:G→H is a homomorphism between two groups, with the identity of G denoted e_G and the identity of H denoted e_H. Show that $f(e_G)=e_H$, that is, identity is sent to identity by any homomorphism. It is clear that use the fact that e=ee and the defining property of homomorphism's.

Consider the map $f:Z_9\rightarrow Z_3$ given by $f(R_m)=R_{3m}$ (recall that R_m is a counterclockwise rotation by m degrees). Is this a homomorphism? Find a homomorphism from Z_6 to Z_3.

Is the map $f:Z_6\rightarrow Z_5$ given by $f(R_m)=R_0$ (the identity) a homomorphism? Find a homomorphism from F_2 to Z×Z

Scheduler of Two Types of Data

How to schedule the process between the two huge amounts of data for smallest number of periods? Two data examination taken by the same student cannot be scheduled at the same time using association rule [13]. The huge amount of data either process through horizontal arrangement algorithm or vertical arrangement algorithm.

So by the algorithm [12] make a graph G whose vertices are the exams, two vertices joined by an edge if some student is taking both exams? Then an exam schedule in k periods exists if and only if the graph can be coloured with k colours, that is, there is a homomorphism from G to the complete graph K. Let H be the core of G and choose homeomorphisms φ: H → G and ψ: G → H. Then φ is an embedding of H as induced subgraph of G, and φ and ψ are isomorphism between H and G| (V H) φ, so that φψ is an automorphism of H. Now, for any automorphism α of G, φαψ is an automorphism of H. We can choose α to map any vertex in (V H) φ to any other; so the automorphism group of H is vertex-transitive.

Conclusion

Paper presents a token based mutual exclusion algorithm for distributed system and the proposed approach gives the better results if compared with the existing token based mutual exclusion techniques. Proposed approach reduces the message complexity up to $n^{½}$ in the worst case and it is just 2 in the best case.

Here the red and blue graph represents two cases (case I, case II). Case I, read line/graph represents the uniform loads with fixed interval. Case II, blue line/graph represents the non uniform loads with fixed interval.

Table 1.1 Data / Message complexity of $n^{1/2}$ Non-Uniform load

Request Generations	10	10	10	10	20	30.5	40.5	50
Total No of Message	0	2.50E.05	5.00E.05	1.00E.04	1.00E.03	2.50E.03	5.00E.03	1.00E.02

Table 1.2 Data / Message complexity of $n^{1/2}$ Uniform load

Request Generations	20	20	20	20	40	50	60	50
Total No of Message	0	2.50E.05	5.00E.05	1.00E.04	1.00E.03	2.50E.03	5.00E.03	1.00E.02

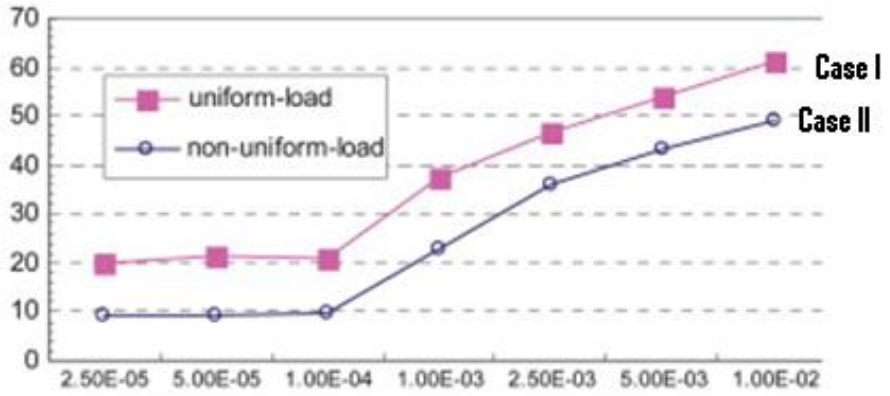

Figure 1.0 Message complexity of $n^{1/2}$ between two cases

References

[1] Mohamed Naimi, Michel Trehel, and Andre Arnold. A log(n) distributed mutual exclusion algorithm based on path reversal. J. Parallel Distrib. Comput. 34(1):1-13, 1996.

[2] Ichiro Suzuki and Today Kasami. A distributed mutual exclusion algorithm. ACM Trans. Comput. Syst., 3(4):344-349, 1985.

[3] Kerry Raymond. A tree-based algorithm for distributed mutual exclusion. ACM Trans. Comput. Syst., 7(1):61-77, 1989.

[4] Mukesh Singhal. A heuristically-aided algorithm for mutual exclusion in distributed systems. IEEE Trans. Comput., 38(5):651-662, 1989.

[5] Quazi Ehsanul Kabir Mamun and Hidenori Nakazato. A new token based protocol for group mutual exclusion in distributed systems. In ISPDC, pages 34-41, 2006.

[6] Sebastien Cantarell, Ajoy Kumar Datta, Franck Petit, and Vincent Villain. Token based group mutual exclusion for asynchronous rings. In ICDCS, pages 691-694, 2001.

[7] Kerry Raymond, A Tree-Based Algorithm for Distributed Mutual Exclusion, ACM, (1989) ISBN 0734-2071/89/0200-0061.

[8] Supriya Madhuram, Anup Kumar, A Hybrid Approach for Mutual Exclusion in Distributed Computing Systems, IEEE, (1994) ISBN 0-8186-6427-4/94.

[9] Quazi Ehsanul Kabir Mamun, Hidenori Nakazato, A New Token Based Protocol for Group Mutual Exclusion in Distributed Systems, IEEE, 2006, ISBN 0-7695-2638-1/06.

[10] Cormen, Thomas H. Leiserson, Charles E. Rivest, Ronald L. Stein, Cli ord, MIT Press and McGraw-Hill, 2009,ISBN 0-262-03293-7.

[11] Andrew S.Tanenbaum, Maarten Van Steen, Distributed systems,Pearson Education, 2007, ISBN 0-13-239227-5

[12] Erkin, Reginald L. Lagendijk ,Efficient Privacy Preserving K-means Clustering In a Three-Party Setting Michael Beye, Zekeriya Information Security and Privacy Lab, Faculty of EEMCS, Delft University of Technology 2628 CD, Delft, The Netherlands. 978-1-4577-1019-3/11/$26.00 ©2011 IEEE

[13] J Vaidya and C. Clifton, Privacy preserving association rule mining in vertically partitioned data. In Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002, pp.639-644.

Mr. Anurag Singh has completed his bachelor degree in mathematics and master degree in computer science and applications. Mr. Singh has Brilliant academic record with good project supervision. He has more than five master students supervised in various project. He has good grip in various software language ie. Java, C,C++ etc. Currently he is PHD Scholar from Babu Banarsi Das University Lucknow.

Dr. Amod Tiwari: Born in 1974 in Kannouj district (Uttar Pradesh). He acquired his Bachelor degree in Mathematics and Science from CSJM Kanpur University Kanpur and master degree in Computer Science and Engineering from Bilaspur Central University Bilaspur (CG) in India. His Academic excellence shines further with PhD in Computer Science and Engineering from Indian Institute of Technology Kanpur awarded with UPTU Lucknow. He worked for reputed firm like LML Scooter India Ltd, Kanpur, at senior level more than two years. He has been associated with Indian Institute of Technology Kanpur from 2005 to 2010. He has worked as Associate professor and Dean Academic and Affairs in PSIT Kanpur from 2009 to 2013. Currently working as Director in “Bhabha Institute of Technology” Kanpur. Dr. Tiwari has more than 45 Publications in his credit.