ISSN No. 0976-5697

Volume 4, No. 10, September-October 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Presenting a new Algorithm for Management Mutual Exclusion in Distributed Systems by Connecting MESH

Hossein Nick khah*
Department of computer, Science and research branch
Islamic Azad University, Sirjan, Iran
hossein.nick.khah@live.com

Farokh Koroupi
Department of computer
Islamic Azad University, Baft, Iran
farokh.koroupi@iaubaft.ac.ir

Arash Azizi mazrae
Department of computer
Islamic Azad University, Sirjan, Iran
aazizim@iausirjan.ac.ir

Abstract: Algorithm performance in distributed systems directly from the balance between concurrency and mutual exclusion takes effect. The two categories are not mutually exclusive and antagonistic concurrency to reduced efficiency and lack of mutual exclusivity loss of accuracy function ends. So they both exist in the system is essential. Normally communication in distributed systems based on message processing, there is no shared memory. Thus number of messages posted an important criterion for measuring the effectiveness of the proposed algorithms for distributed mutual exclusion is about. In this paper, an algorithm for token-based distributed mutual exclusion problem is presented. This algorithm is a package that permits the processing moves each node receives a packet can take advantage of critical region. The algorithm solves the deadlock and starvation by a mesh topology. In this algorithm, traffic is minimal and the system does not interfere in the job log. Proposed algorithm has capability management different critical regions at the same time. For each critical region of that can be closed in a separate license and permit all packets in the node to the node for the critical region, permits for the package to be another critical region. So act quickly of distributed system is rises.

Keywords: Critical region, depth first traversal, distributed systems, mutual exclusion, mesh topology.

I. INTRODUCTION

The hardware and software systems that are connected to each other and to have shared a long distance are called distributed systems [1]. Obviously, when you use multiple computers from a common source, there is a potential interaction process [2, 3]. Shared areas to manage mutual exclusion problem arises. Mutual exclusion problem was first presented by Professor Joung. Mutually exclusive, shared resources and means to prevent the processes that are going on at one time, work with a common source [3]. To an area where there is risk of collision processes, is called the critical region. Suppose that different processes at the same time they want printed by the printer. Many algorithms for managing distributed systems are critical areas that we describe in this paper a brief overview of each algorithm [4, 5]. The superiority of the algorithm in response time and number of messages to be exchanged between processors is specified. The response time is important because the CPU must be in the best response time and the number of messages is important because the system has little traffic [6, 7, and 8]. May be there is thousands of computers in a distributed system. Traffic is so important. The topology of this algorithm is mesh topology and this topology will look into it, then an algorithm for the management of critical areas in which the connection between computers in a distributed system is a mesh topology, we present. Finally, we compare the proposed algorithm with other algorithms; we demonstrate the advantages of this algorithm.

II. DISTRIBUTED MUTUAL EXCLUSION ALGORITHMS

A. Centralized algorithm:

In this algorithm, there is a central coordinator who is in charge of managing the crisis. Each node is a critical area of application with a request message (Request) sent to the coordinator [9]. Then you can open critical area coordinator a message "ok" and it will be sent to that node, the node can be a critical area to begin his career. But if the coordinator is critical in requesting node does not send any messages to the nodes in the queue to be released to a message critical region at any time "ok" nodes it send a requesting . Send the request of the problems ahead, because the requesting node knows in the waiting queue is destroyed or a coordinator. Perhaps the requesting node to infinity in the critical region is expected. If the coordinator is corrupted, then the coordinator will make the critical area not assigned to him (Because the request queue is cleared) [10]. Furthermore, if the requesting node coordinator believes that it is broken and re-sends your request is likely waiting a long time and now it has been the coordinator node has sent two requests to two critical areas of allocated to that node. After the nodes with critical area critical areas of a release message (Release) to the coordinator sends the completed form to notify the coordinator [9].

B. Distributed Algorithms:

This algorithm has been presented by Mr. Ricart and Agrawala [11]. This algorithm is based on clock synchronization Mr. Lamport is derived from the theory [11]. In this algorithm, there is no central coordinator. All nodes are involved but also to manage the crisis. Politics of this algorithm so that each node has to do with the crisis, make a package with its current network address and time, to be sent to other nodes [11].

All three conditions occur:

a. Nodes receiving the package do not work with critical area after the sender sends the message Ok [12].
b. Nodes receiving the regions critical needs work and has submitted a request packet to all other nodes. Now it's time stamp with the time stamp is received and the time stamp received is less than the requester sends the message Ok [12].
c. After receiving node with the node requesting critical works on the line puts up his own business and then all messages will be sent to the applicant's Ok [12].

If the requesting node to all nodes in the system of the right to enter the message Ok received critical and otherwise wait. This expectation is problematic, because it is a faulty node or nodes, but is unable to send Ok message from that node are waiting, and the system is idle. In a sense, each time the total number of nodes in the voting system should be time stamped messages to be sent and then received the same number is Ok [13]. If there are n nodes in the network must be 2n and the number of messages exchanged messages caused traffic is high.

C. Token ring algorithm:

In this algorithm, nodes are located next to each other in the ring topology cues (i.e. the processors are located on a circle adjacent processors is a direct connection between the processor and between processors non-contiguous Interconnection that there are) [14]. Among processors in a single package, there is a suspended between the processor is running. Each processor that receives the packet, it will be critical if the region is to keep close to the critical region. After finishing with the critical region where the packet leaves the network, the other processors can use the critical region. In this model, if a node has received the packet is damaged, or if the connection between two nodes is cut package is also missing, and the entire node receives the packets are for. After this account, there is failures point in the algorithm, but also reduce the number of messages and network traffic is low [15].

If the package is lost in the network must be built on a new package. It is also difficult because the packet loss is due to late arrival of packets [16]. Perhaps a node receives a packet with the crisis has been a long time, and also the other nodes in the network has a packet and the network is located in the two license packages. A timeframe for building permits is considered. A node during its first period and did not receive the package, the new package will be created in the network [15, 16].

III. MESH TOPOLOGY

The topology of the network nodes on a two-dimensional matrix of rows and columns together, and each node is connected with its neighboring nodes [17, 18, 19, and 20]. It is also possible the communication between nodes by the intermediate node. In this topology, all the nodes are connected. To place the nodes in the topology, nodes with higher priority in the beginning and at the end of the network nodes with lower priority process [19, 20]. Figure 1 is an example of the mesh topology is given.

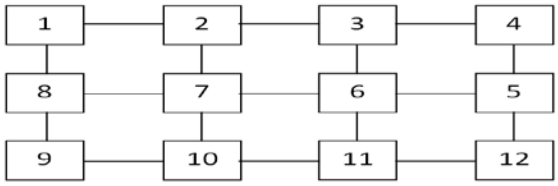

Figure 1. A distributed system that has 12 computers together to form a mesh topology [20].

If the network algorithms by DFS (depth first search) to navigate, it is noticeable that the list is an ordered list. DFS traversal of the nodes connected to the current node, and then navigate to the next level and make it to all nodes in the navigation continues [21, 22].

IV. PROPOSED ALGORITHM

In a distributed system, which we refer to in this article, there is a license package that was suspended between nodes, and each node can receive a packet with the permission of critical work to do. On the other DFS algorithm we described how to traverse the network. It is noted that this topology can be seen in several critical areas for each packet in the network can be licensed. This node can vary in a time critical work areas, and the system efficiency is maximized. When the number of messages exchanged between nodes are allowed to be exchanged two messages. The first message from the source node to the destination node is the destination node indicates that it is a license and permit confirmation message to the source node announces. If the message reaches the first node and the second node, the message is not sent, the source node to the destination node thinks is wrong and will send the license to the next node, but you get the authorization message to the destination node, and the confirmation so it can be a critical area to work with.

If the work is not critical, so the packet to the next node on the network which is derived from the DFS traversal sends, and the waiting area is critical, so keep it closed and its work in the area perform critical. Then send the packet to the next node. When selecting the next node to receive a license package, you must consider that the node has not been visited in the traversal. These lacks of hunger for the algorithm are considered. Because if a node in a traversal of the network, so the algorithms are not licensed, there will be hunger. The algorithm is that a node has received the packet is permitted, the number of packets that can be stored scrolling.

The memory complexity of the algorithm is of order O (n) is. Whenever a network survey was completed, the list of nodes that have received the package licensing arrangements were going to clear up in the next step we traverse the nodes can access it. For a full survey is conducted to determine whether a variable CNT consider that each node has a single packet authorization can be increased. Increasing CNT vary depending on the sender node will be licensed, so if this node is supposed to receive a license package is broken, the CNT variable nodes in to navigation. Finally, it is CNT = N (N is the number of nodes on the network), the network is fully traversal algorithm is then equal to zero and the CNT array traversal nodes are also removed. For example, the network in Figure 2 describes the survey process are.

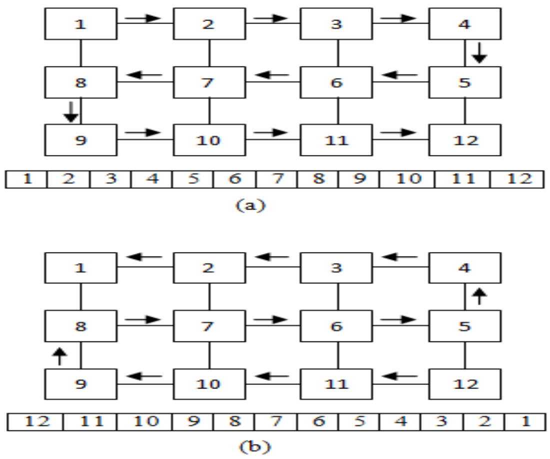

Figure 2. Traverse to Network

At the beginning of a packet network license, the scrolling action begins, and goes up to the net, and when CNT = N, the list is cleared and CNT = 0 is met. Next, scroll, scroll to the end of the license is changed or closed. Meeting list a, b are symmetric exactly, but it's not always because if a node goes down, depending on the direction of change. Suppose package licensing in the way, all nodes are correct and on the way back down to 8 knots. Figure 3 makes navigating the list of networks.

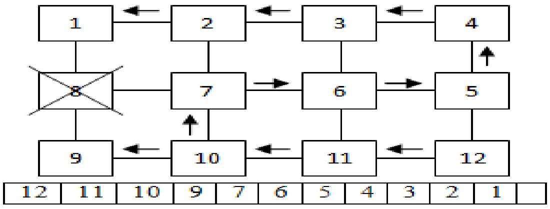

Figure 3. How to navigate the network when a node failure.

Also, if a node went down the road and be right back on track, navigating DFS, it puts the node into the list of recurring appointments, it can also be the node list of nodes using the critical placed. This is why every time you see the list of network-on surveys completed to clean up the list again next survey to be completed. This property makes the algorithm dynamically act.

In mesh networks, each node has at least two neighbors and at most four neighbors, so if a node wants to send packets to all neighbors that fail authorization if the neighboring nodes of the network will be working. To address this problem, in this situation, we consider a time interval in primary node. If the first node in the network after this time period, no packets received, then the network will generate a new package. Also, a node and its neighbors, was shut down in the previous license had already closed and waiting for the new package will be destroyed.

This is due to the elimination of prior authorization package, if the neighbors were true, then closed it and now pending in the network, the network is allowed two bags, and it is possible that the two processors depending on the license they are together entering the critical region, the mutual exclusion problem is a defect. After all nodes have an equal time interval, and the first node of the packet received after this time period, the other is closed, and the rest of the nodes, depending on if it failed to a, then destroy the package. The important thing is timeframe sizes. Consider a small time interval is a problem in the system. It is not complete because the network is still closed to navigation license and goes away this hunger is the last node. Large considering the efficiency of the algorithm reduces the required timeframe. It is possible, then shut down the node that has the permits in hand, will take a long time to make the next batch, and a lot of waiting around for it to stay closed. First, the time interval is very large (infinite). Each time you complete a web survey was conducted on a closed interval of the license, and each node can be considered a license to receive the package, the timeframe to, depending on the period of the license is updated. So every time you navigate to the network, if a node is removed from a scroll or navigate into, the period of time when there is an update, just spent the last navigating.

The algorithm is dynamic in that respect as well. The only disadvantage of the algorithm is responsible for closing the first node and the first node goes down, the whole system is broken. To address this problem, we consider each node at a time. It is enough to generate and send a license. Each node is based on the priorities of the previous node when it is double, and after that time, found that the higher priority nodes are corrupted. So, create a license packet and put in the network. For example, if the waiting time for the package, the first node is two seconds, and the time to generate and send a packet to all nodes 0.5 seconds, In Figure 4, each time the node is displayed.

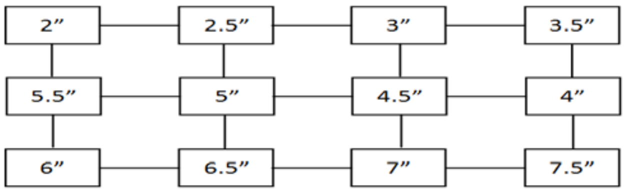

Figure 4. Waiting time to generate a license package

Time license to build and send packets, the first packet by the first node of the first building permit, it is calculated that is sent to each node, that node waiting time for your package to the size of its increases as well. The first time traverse the network in Figure 4 is calculated whenever a node is added to the network when it's time to the size of the node and send a license from the previous calculates. The point of failure in the network is lost.

V. MANAGEMENT ALGORITHM FOR MUTUAL EXCLUSION

Now that all the issues raised optimization algorithm, pseudo code for this algorithm can be cited. This code presented in figure 5

```
Procedure (Mesh algorithm)
Wait for receive package
CS ()
DFS ()
While receive<>"ok" do
Send package to node [index+1]
If remaningtime=timeslot then
   Delete package
    Break
End procedure
Procedure (Receive package)
Send ok to package.sender
End procedure
Procedure (Make a package)
If remaningtime=timeslot then
    Create a package
   Call (mesh algorithm)
End procedure
```

Figure 5. The pseudo code for proposed algorithm

Mesh algorithm permits will initially function for receiving packets, and the packets get your work done with the critical region. Then again a DFS traversal of the network and the node sends the packet to reach a proper knot. If all the nodes are damaged and eliminate the waiting time to finish packing and other nodes in the network, and the network using the other packages are makepackage. Receive the license you received the package, then a message "ok" to permit the sender sends the packets are not sent to other nodes. Receive makepackage functions and must be constantly running, and if any provision of its operations are conducted. So each must be considered in a thread, that thread is constantly running and if the condition algorithms makepackage, receive, these algorithms runs. The algorithm only works when dealing with the crisis started. We can vary depending on the algorithm for each shared resource management is used to take. It does not require nodes to forward packets which are not working as well as system efficiency goes up.

VI. CONCLUSIONS

Since the proposed algorithm uses a mesh topology, any disruption of distribution rules is respected. Because the system will retain the flexibility to easily add or remove a node can work without harming the system is. Reliability is respected because of deterioration or malfunction in the communication line between two nodes of a node, the system will not fail mode and finds another way to navigate a network. Also, because each traversal, we traverse all nodes are available, then there is the issue of hunger processors, and all processors have access to critical areas of the fair.

As noted earlier in network traversal algorithm is DFS. If you want to permit packets from the first node to the last node must be the number of n-1 node to node to traverse to reach the final. The algorithm in the worst case n-1 times Message must send the packet to the other node that receives a message packet "ok" to send the package to the sender. In the worst case, depending on the license is the first node to the end of the 2 (n-1) messages to be exchanged.

The direction if the nodes in the network and shut down the license could not be sent between nodes, the first node to receive packets waiting period expired license and did not receive a license package, depending on the network there make sends. After doing this great failure of processors in the system does not work. The license package is not missing any time. Previous node, depending on the license and managed to send it is not closed after the waiting period is calculated for each processor in the network survey, removing defects, while the package eliminates the interference 2 there is depending on the license.

In the license package, we have defined a variable called CNT.Variable CNT with a single traversal of each node is added. Network traversal algorithms work with DFS, until it is done, the CNT value is smaller than the number of processors in the system or processor who holds a license Package, and cannot you send the packet to another node. When the value is zero in the CNT network traversal in the opposite direction, i.e. from the beginning to the end of the network, the network is looking for. With this interpretation, a node in a navigation bar, two packets will not receive the license, or a node receives a packet cannot be excluded licenses. Then there is the issue of hunger, processors and navigation network at any time, there is equality between processors.

License package includes an array of length n where n is the number of processors in the system. The array of network navigation displays, and license packages are sent to nodes that are not within the array. When scrolling the grid once completed CNT = n, then the array will be erased and CNT = 0.

Finally, in Table I a comparison of the proposed and existing algorithms for distributed mutual exclusion problem, we present the advantages and disadvantages of the algorithms are the points table indicates.

Table I. Comparison of the algorithm with other algorithms

Algorithms	Number of message for services to all nodes	Time of message	Disadvantage
Centralized algorithm	3	2	Coordinator crashed
Distributed algorithm	2(n-1)	2(n-1)	Point of failure
Token-ring algorithm	n-1	1 to ∞	Lost package, point of failure
Proposed algorithm	2(n-1)	2(n-1)	Requires memory for storing traversal history

VII. REFERENCE

[1]. Sau-Ming Lau, Qin Lu, Kwong-Sak Leung, Adaptive load distribution algorithms for heterogeneous distributed systems with multiple task classes, Journal of Parallel and Distributed Computing, Volume 66, Issue 2, February 2006, Pages 163-180.

[2]. Sheng-Hsiung Chen, Ting-Lu Huang, A tight bound on remote reference time complexity of mutual exclusion in the read-modify-write model, Journal of Parallel and Distributed Computing, Volume 66, Issue 11, November 2006, Pages 1455-1471.

[3]. Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, Petr Kouznetsov, Mutual exclusion in asynchronous systems with failure detectors, Journal of Parallel and Distributed Computing, Volume 65, Issue 4, April 2005, Pages 492-505.

[4]. Guo-Zhong Tian, Jiong Yu, Jing-Sha He, Towards critical region reliability support for Grid workflows, Journal of Parallel and Distributed Computing, Volume 69, Issue 12, December 2009, Pages 989-995.

[5]. Madjid Allili, David Corriveau, Sara Derivière, Marc Ethier, Tomasz Kaczynski, Detecting critical regions in multidimensional data sets, Computers & Mathematics with Applications, Volume 61, Issue 2, January 2011, Pages 499-512.

[6]. Shenghai Liu, Suili Feng, Wu Ye, Hongcheng Zhuang, Slot allocation algorithms in centralized scheduling scheme for IEEE 802.16 based wireless mesh networks, Computer

Communications, Volume 32, Issue 5, March 2009, Pages 943-953.

[7]. K. Christodoulopoulos, V. Sourlas, I. Mpakolas, E. Varvarigos, A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks, Computer Communications, Volume 32, Issues 7–10, May 2009, Pages 1172-1184.

[8]. Peyman Neamatollahi, Hoda Taheri, Mahmoud Naghibzadeh, Info-based approach in distributed mutual exclusion algorithms, Journal of Parallel and Distributed Computing, Volume 72, Issue 5, May 2012, Pages 650-665.

[9]. Christian Lavault, Mario Valencia-Pabon, A distributed approximation algorithm for the minimum degree minimum weight spanning trees, Journal of Parallel and Distributed Computing, Volume 68, Issue 2, February 2008, Pages 200-208.

[10]. Sagar A. Tamhane, Mohan Kumar, A token based distributed algorithm for supporting mutual exclusion in opportunistic networks, Pervasive and Mobile Computing, Volume 8, Issue 5,October 2012, Pages 795-809.

[11]. Stefan Dobrev, Paola Flocchini, Rastislav Královič, Nicola Santoro, Exploring an unknown dangerous graph using tokens, Theoretical Computer Science, Volume 472, February 2013, Pages 28-45.

[12]. Hoda Taheri, Peyman Neamatollahi, Mahmoud Naghibzadeh, A hybrid token -based distributed mutual exclusion algorithm using wraparound two-dimensional array logical topology, Information Processing Letters, Volume 111, Issue 17, September 2011, Pages 841-847.

[13]. Fabio Martignon, Stefano Paris, Antonio Capone, Design and implementation of MobiSEC: A complete security architecture for wireless mesh networks, Computer Networks , Volume 53, Issue 12, August 2009, Pages 2192-2207.

[14]. Okechukwu E. Muogilim, Kok-Keong Loo, Richard Comley, Wireless mesh network security: A traffic engineering management approach, Journal of Network and Computer Applications, Volume 34, Issue 2, March 2011, Pages 478-491.

[15]. Paolo G. Franciosa, Giorgio Gambosi, Umberto Nanni, The incremental maintenance of a Depth-First - Search tree in directed acyclic graphs, Information Processing Letters, Volume 61, Issue 2, January 1997, Pages 113-120.

[16]. Christophe Duhamel, Philippe Lacomme, Caroline Prodhon, A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems, Engineering Applications of Artificial Intelligence, Volume 25, Issue 2, March 2012, Pages 345-358.

[17]. Ajay D. Kshemkalyani, MukeshSinghal, Efficient distributed snapshots in an anonymous asynchronousmessage-passing system, Journal of Parallel and Distributed Computing, Volume 73, Issue 5, May2013, Pages 621-629.

[18]. Roberto Baldoni, Silvia Bonomi, Adriano Cerocchi, Leonardo Querzoni, Virtual Tree: A robust architecture for interval valid queries in dynamicdistributed systems, Journal of Parallel and Distributed Computing, Volume 73, Issue 8,August 2013, Pages 1135-1145.

[19]. Qinma Kang, Hong He, Jun Wei, An effective iterated greedy algorithm for reliability-oriented taskallocation in distributed computing systems, Journal of Parallel and Distributed Computing, Volume 73, Issue 8,August 2013, Pages 1106-1115.

[20]. Gang Wu, Huxing Zhang, MeikangQiu, Zhong Ming, Jiayin Li, Xiao Qin, A decentralized approach for mining event correlations in distributedsystem monitoring, Journal of Parallel and Distributed Computing, Volume 73, Issue 3,March 2013, Pages 330-340.

[21]. HosseinSaiedian, Gabe Wishnie, A complex event routing infrastructure for distributed systems, Journal of Parallel and Distributed Computing, Volume 72, Issue 3,March 2012, Pages 450-461.

[22]. HaiyingShen, A P2P-based intelligent resource discovery mechanism in Internetbaseddistributed systems, Journal of Parallel and Distributed Computing, Volume 69, Issue 2,February 2009, Pages 197-209.