A New Function-based Framework for Classification and Evaluation of Mutual Exclusion Algorithms in Distributed Systems

Leila Omrani *L.Omrani@Qiau.ac.ir*
Department of Computer Engineering
Islamic Azad University of Qazvin
Qazvin , 34185-1416,Iran

Zahra Rafinezhad *Z.Rafinezhad@Qiau.ac.ir*
Department of Computer Engineering
Islamic Azad University of Qazvin
Qazvin , 34185-1416,Iran

Mohammadreza Keyvanpour *Keyvanpour@Qiau.ac.ir*
Department of Computer Engineering
Islamic Azad University of Qazvin
Qazvin , 34185-1416,Iran

Abstract

This paper presents a new function-based framework for mutual exclusion algorithms in distributed systems. In the traditional classification mutual exclusion algorithms were divided in to two groups: Token-based and Permission-based. Recently, some new algorithms are proposed in order to increase fault tolerance, minimize message complexity and decrease synchronization delay. Although the studies in this field up to now can compare and evaluate the algorithms, this paper takes a step further and proposes a new function-based framework as a brief introduction to the algorithms in the four groups as follows: Token-based, Permission-based, Hybrid and K-mutual exclusion. In addition, because of being dispersal and obscure performance criteria, introduces four parameters which can be used to compare various distributed mutual exclusion algorithms such as message complexity, synchronization delay, decision theory and nodes configuration. Hope the proposed framework provides a suitable context for technical and clear evaluation of existing and future methods.

Keywords: Mutual Exclusion, Critical Section, Token

1. INTRODUCTION

The mutual exclusion problem involves the allocation of a single, non shareable resource among n processes [1], by means that just one process can execute in its critical section at a given time. Mutual exclusion problem was first introduced in centralized systems. In these systems mutual exclusion ensure with preserving semaphores and monitors, and one of the nodes function as a central coordinator which is fully responsible for having all the information of the system and processes ask only the coordinator for permission to enter their critical sections. But in distributed systems the decision making is distributed across the entire system and the solution to the mutual exclusion problem is far more complicated because of the lake of common shared memory and physical clock. So obtain a complete knowledge of the total system is difficult.

Lots of algorithms are proposed in distributed systems. They classified into two groups traditionally. One of them is token-based, in this group there is a unique token in the system which ensure mutual exclusion. So the requesting node must have it to enter the critical section. Another one is permission-based group, that requesting node has to ask all other nodes for their permissions to enter the critical section [5]. According to approach of new algorithms, in this paper we proposed a new function-based framework which classified these algorithms in four groups as Token-based, Permission-based, Hybrid and K-mutual exclusion. Also in new approach spite of synchronization delay and message complexity in light and

heavy loads, we introduced two new measures including decision theory and nodes configuration.

This paper is organized as follows: in section 2, presents general model of distributed system and formally describes the mutual exclusion problem. In section 3, introduces the proposed framework and measures with classification of algorithms. Finally in last section concludes our work.

2. MODEL AND PROBLEM DEFINITION

2.1. System Model

In general, most of mutual exclusion algorithms use a common model. In this model, a distributed system is a set of independent and autonomous computers. These computers are called as node or site and connected via a communication network. Each node has abstract view of whole system that communicates with message passing [4]. The most important purposes of distributed system are assigned as providing an appropriate and efficient environment for sharing resource, having an acceptable speed and high reliability and availability.

2.2. The Mutual Exclusion Problem

Mutual exclusion problem in distributed systems has received great consideration in recent 3 decades. This problem ensures that concurrent processes access common resource and data sequentially. In addition each process that executes in its critical section for a finite time, must do without interfering with other processes. Also, when no process is in a critical section, any process that request entry to its critical section must be permitted to enter without delay [3]. Eventually mutual exclusion must be without deadlock and starvation.

3. PROPOSED FRAMEWORK AND CRITERIA FOR CLASSIFICATION OF MUTUAL EXCLUSION ALGORITHMS

By reason of the mutual exclusion importance in distributed system for keeping system consistency and increasing concurrently, various algorithms are proposed. In order to evaluate performance of these algorithms various criteria are defined as synchronization delay and message complexity. Synchronization delay is the average time delay in granting critical section. Message complexity is the number of messages exchanged by a process per critical section entry. Also evaluates message complexity in different heavy and light load of system state.

In addition, two new measures are proposed as decision theory and nodes configuration. In the first one, if each node need not keep information about the concurrent state of the system, the algorithm will be called static. On the other hand the algorithm is called dynamic. Also, in nodes configuration if nodes are assumed to be arranged in logical configuration, the algorithm is called structural, otherwise is called nonstructural. In next sections, the proposed framework is presented for classifying mutual exclusion algorithms with their evaluation as follows: 1.Token-based, 2.Permission-based, 3.Hybrid, 4.K-mutual exclusion.

3.1.Description of Token-based Approach

In this approach the right to enter a critical section is produced by a unique message, named token. The concurrent owner of the token chooses the next token owner and sends it the token. So granting the privilege to enter the critical section performs by the owner of the token. In 1985, Suzuki and Kasami [5], presented an algorithm that token by means of privilege message transmitted to requesting process based on sequence number.
Some of the algorithms use a logical structure like in Raymond [6]. In this algorithm, nodes (each process performs in a node) arranged in rootless tree structure and every node is related only to its neighbors and is aware of their information. In Naimi and Trehel algorithm [7], each node sends its request only to another one that knows as a current root and waits for its permission. In addition this algorithm uses two data structures, one of them is a queue for keeping requests and the other is a logical rooted dynamic tree for assigning token. But this algorithm is so sensitive to node failure and recovery. In consequence in [8] presented a dynamic algorithm which is able to failure detection, regenerates lost token and robust against failures.

In 2006 a new algorithm presented as queue migration [9], that nodes arranged in a distributed and fully connected network. In this network, nodes in logical group can communicate directly together for the purpose of entering the critical section. One node from each group is selected to form part of the global group this node is called a link node. Link node collects logical and global requests and sends them to the owner of token. So token transmits among local and global groups for ensuring mutual exclusion.

3-1-1. Evaluation of Token-based Approach

The most advantage of Token-based approach is simplicity. In this approach, for example if the logical structure of algorithm is a ring then the token transmits from a node to another continuously. Table1 represent the comparison and evaluation of well-known algorithms which mentioned in previous section according to proposed measures. Under light load, this method is so expensive because token is broadcasted without using by any nodes. But under heavy load it's so efficient. According to results that show in table1 Suzuki and Kasami's algorithm [5] has least message complexity under light load and queue migration algorithm [9] has least message complexity under heavy load. As you see in table1, the algorithm which has a structural configuration and dynamic decision theory can has less synchronization delay.

Token-based approach is so capable to lose token and deadlock occurs if it can't regenerate token. But if algorithm assumes the existing token is lost and regenerate another one, it will violate mutual exclusion [10]. Another problem of this approach is low scalability because all nodes arranged in a logical structure. So, when the number of nodes increment, the average of waiting time increase. Most of these algorithms have a structural configuration and decision theory make dynamically.

3.2.Description of Permission-based Approach

In Permission-based approach, requesting node asks to obtain permissions from a set of nodes in the systems. A priority or an order of events have to be established between competing requesting nodes, so only one of them receives permission from all other nodes in the set [5]. After receiving permission from a sufficient number of nodes, it is allowed to enter the critical section. When a node is completed its execution in critical section must sends release message to the other nodes. The main problem is finding a minimal number of nodes from which a node has to obtain permission to enter its critical section. Many algorithms have been developed to find this minimal, such as Lamport algorithm [11]. This algorithm uses a mechanism based on logical clocks for the total ordering of requests in the system.

Algorithm name	Evaluation measures					Description
	Message complexity		synchronization delay	configuration	decision theory	
	Heavy load	Light load				
Suzuki-Kasami	N	0	_	Non structural	dynamic	Token as a privilege message
Raymond with tree structure	O(log N)	O(log N)	((log N)/2)T	structural	static	Nodes in rootless tree
Naimi-Trehel	O(log N)	O(log N)	T	structural	dynamic	Use two structures :queue& logical tree
Dynamic tree	O(log N)	O(log N)	T	structural	dynamic	With failure detection & recovery
Queue migration	2	$O(\sqrt{N})$	_	structural	dynamic	With global & local groups

TABLE 1: comparison and evaluation of Token-based approach

In Ricart and Agrawala algorithm [12], when a node receives a request compares its sequence number with previous request and allows the request with smallest sequence

number to enter the critical section. After that in [13], both of above algorithms exchanged and introduced a new algorithm which is the best known algorithm that guarantees fairness in the same sense. It means, when there is a high priority request to do, the low priority request is delayed.

3.2.1. The Group Mutual Exclusion Problem

In this problem, every request for a critical section is associated with a type or group. It means at any time none of two processes which have requested critical sections belonging two different groups in their critical sections are simultaneously. In addition it's free of starvation it means a process wishing to enter critical section succeeds eventually.

Also, concurrent entry property is the most important issue in group mutual exclusion, it means if all requests are for critical sections belonging to the same group, then no requesting process should wait for entry in to its critical section until some other processes have left it [14]. The concept of quorum is used to solve this problem. In fact quorum is a subset of processes. Each process enters its critical section only after it has successfully locked all nodes in its quorum. There are two properties in the concept of quorum which can ensure mutual exclusion requirements: the first one is intersection and the second one is minimally [14].

At first Maekawa in 1985[15], used the concept of quorum in his algorithm. In this algorithm requests serviced based on their sequence numbers. After that another algorithm [16] represented by using the concept of dynamic quorum. The purpose of this algorithm was high scalability and low message complexity. This algorithm acts independent on its quorum system. In Maekawa algorithm a node never changes its quorum or requests from the other quorums but here a node can link to member of other quorums and respond all of its requests by making some changes during execution. Hence the quorums change dynamically. Then, the delay optimal algorithm was presented in [17]. This algorithm has least delay with fix message complexity among of above algorithms.

3.2.2.Evaluation of Permission-based Approach

Fairness is a very important measure for solutions to the most contention problems. In the concept of mutual exclusion is that requests for access to the critical section are satisfied in the order of their timestamps. This concept is obvious in the algorithms which mentioned above, clearly [13].

Table2 presents the comparison and evaluation of well-known algorithms which mentioned in previous section according to proposed measures. Beside the number of messages are exchanged for accessing the resources capture the overhead imposed on the system. In the various algorithms are tried to get an optimal value. This value in group mutual exclusion algorithms is limited to the number of quorum members [16].

As mentioned in section 3.1.1, most of the token-based algorithms configuration was structural, which decreased scalability. Also detection of lost token and regenerate were the other problems of this approach. But in permission-based approach, are tried to solve these problems. Here configuration of nodes in most of the primary algorithms was nonstructural. However after introducing the group mutual exclusion, is imposed the logical structure to the system and the configuration is became structural. But as show in table2 synchronization delay increased when the configuration of algorithms became structural. So delay optimal algorithm solved this problem.

3.3. Description of Hybrid Approach

Providing deadlock-free distributed mutual exclusion algorithms is often difficult and it involves passing many messages, so the hybrid algorithms are introduced. Since call such algorithm hybrid that uses both Token-based and Permission-based approach for mutual exclusion assurance, simultaneously. One of the algorithms of this group is proposed by Paydar et al[10]. In which sets n nodes in 2-dimension array. This array is composed of $\sqrt{n}$ rows and $\sqrt{n}$

Algorithm name	Evaluation measures					Description
	Message complexity		synchronization delay	configuration	decision theory	
	Heavy load	Light load				
Lamport	3(N-1)	3(N-1)	T	Non structural	static	Give priority with timestamp
Ricart-Agrawala	2(N-1)	2(N-1)	T	Non structural	dynamic	Get n-1 permissions
fair	2(N-1)	(N-1)	T	Non structural	dynamic	Give priority with FIFO
Maekawa	$5\sqrt{N}$	$3\sqrt{N}$	2T	structural	static	Use quorum
quorum Dynamic	O(q)	O(q)	3T	structural	dynamic	Generate dynamic quorum
Delay optimal	$6(\sqrt{N}-1)$	$6(\sqrt{N}-1)$	T	structural	dynamic	Least synchronization delay

TABLE 2: comparison and evaluation of Permission-based approach

columns. Every node i (i=1,2,3,…,n) with the other $\sqrt{n}-1$ nodes in the same row and the other $\sqrt{n}-1$ nodes in the same column form the quorum Q_i. Every node can enter to critical section when it obtains the permissions of all nodes in the same quorum and gets the token too.

Kakugawa et al [18] presented an algorithm that used the coterie concept. Coterie is a set of quorums, each of which is a subset of the process set and any two quorums share at least one process. In coterie concept both intersection and minimally properties are satisfied. This algorithm uses two classes of token, main-token and sub-token. If each process of a coterie requests to enter the critical section, the owner of the main-token by generating necessary sub-tokens can respond their requests.

Then an algorithm based on Suzuki and Kasami's algorithm [27] is proposed in [19,20]. This algorithm uses non-uniform groups, in which, the same groups set in one session. Also, uses two kind of tokens to enhance concurrency of Suzuki and Kasami algorithm: primary and secondary. The owner of the primary token can grant the secondary token to others.

3.3.1. Evaluation of Hybrid Approach

In algorithms of this approach, every node begin its function in a permission-based manner but continues in token-based. This approach can overcome token-based problems, Because of using both token-based and permission-based techniques.

Table3 represents the comparison and evaluation of well-known algorithms which mentioned in previous section according to proposed measures. Most of the algorithms in this group improve message complexity and increase the degree of concurrency. This issue avoids unnecessary blocking [19,20], also they can increase fault tolerance by using failure recovery and failure detection mechanisms.

According to table3 these algorithms because of using quorums have structural configuration and their decision theory is dynamically, so these features help them to overcome the only token-based or permission-based algorithm's problems.

3.4. Description of K-mutual Exclusion Approach

The K-mutual exclusion problem is a fundamental distributed problem that completes the mutual exclusion issue. It guarantees the integrity of the k units of a shared resource by restricting the number of processes that can access them simultaneously [21]. Likewise these algorithms divided in to token-based and permission-based groups. In token-based group, k tokens are generated to let requested processes to enter their critical sections. But in permission-based group a node gets in to the critical section only after sending requests to

Algorithm name	Evaluation measures					description
	Message complexity		synchronization delay	configuration	decision theory	
	Heavy load	Light load				
Paydar algorithm	$4\sqrt{N}$	$4\sqrt{N}$	–	structural	Dynamic	Quorum contains of same column and row nodes
Kakugawa	5|Q|+1	0	3T	structural	Dynamic	With coterie and main token & sub token
Non-uniform groups	2N-1	2N-1	T	structural	Dynamic	With non-uniform groups

TABLE 3: Comparison and evaluation of Hybrid approach

the $n-1$ other nodes and receiving permission from $n-k$ nodes [22].

One of the algorithms of this group that uses queue migration is extended on [9]. In this algorithm for ensuring mutual exclusion, the link node is the owner of parent token which has the capacity of generating $k-1$ tokens. In [23] the Raymond's algorithm [6] extended. Although in this algorithm each node has a sequence number, it has to obtain $n-k$ permissions to enter the critical section. In [21] the Raymond's algorithm is extended again and it increases fault tolerance to $n-$ nodes, in which in Raymond's algorithm it was $k-1$ nodes. So it ensures that even occurs failure, k processes can execute simultaneously.

3-4-1.Evaluation of K-mutual Exclusion Approach

Most of the k-mutual exclusion algorithms focus on fault tolerance. They provide using more resource at the same time. For example, some of these algorithms can prevent the system security until occurs $k-1$ failures. It means when a node fails, It is impossible to any others can collect $n-k$ permissions and enter to the critical section. But after $k-1$ failures, the most number of processes that can execute simultaneously reach to one. As a result the algorithm's performance decreases. Also, if the number of the received permissions decreased by occurring failures dynamically, the value of fault tolerance will reach to $n-1$ nodes. So it ensures that even occur failures, k processes can execute in their critical sections simultaneously [23,24].

Table 4 represents the comparison and evaluation of well-known algorithms which mentioned in previous section according to proposed measures. As you see decision theory in these algorithms are dynamically.

Algorithm name	Evaluation measures					description
	Message complexity		synchronization delay	configuration	decision theory	
	Heavy load	Light load				
k-queue migration	$O(\sqrt{N})$	0	4T	structural	dynamic	Use parent token & generate k-1 tokens
Raymond with multi entries	2N-1	2N-k-1	_	Non structural	dynamic	K resources & n-k replies
Obtain n_i-k replies from n_i correct nodes	2N-1	2N-k-1	_	Non structural	dynamic	Extended of Raymond algorithm

TABLE 4:comparison and evaluation of K-mutual approach

4. CONCLUSION

According to importance of mutual exclusion in achieving goals of distributed systems, various approaches are proposed. In this research try to make a brief analyzing on most

common distributed mutual exclusion algorithms. Also, present a new classification based on their functions in four groups: Token-based, Permission-based, Hybrid and K-mutual exclusion. This framework helps novice researchers to set each new algorithm in specific category. To achieving this purpose focus on four measures such as message complexity, synchronization delay, decision theory and nodes configuration for comparison and evaluation of them. Hope the proposed framework makes a convenient way for future researches**.**

5.REFERENCES

[1] N.A.Lynch . "*Distributed Algorithms"* , Morgan Kaufmann Publishers, pp.255-327,(1996)

[2] P.C.Saxena, and J.Rai. "*A Survey Of Permission-based Distributed Mutual Exclusion Algorithms*" . Computer Standards & Interfaces*, 25*: 159-181, 2003

[3] M.G.Velaquez. "*A Survey Of Distributed Mutual Exclusion Algorithms*". Technical Report CS. Colarido state university, September 1993

[4] W.Stallings. *"Operating Systems Internals and Design Principls*", Prentice Hall, pp.205-261 (2009)

[5] I.Suzuki,and T.Kasami."*A Distributed Mutual Exclusion Algorithm*", ACM Transactions On Computer Systems, Vol.3(No.4): 344-349, November 1985

[6] K.Paymond. "*A TreeBased Algorithm For Distributed MutualExclusion*",ACM Transactions On Computer System*,* Vol.7(No.1): 61-77, February 1989

[7] M.Naimi, M.Trehel, and A.Arnold. "*A Log(n) Distributed Mutual Exclusion Algorithm Based On The Path Reversal*", Journal Of Parallel And Distributed Computing*,34(1)*: 1-13 April 1996

[8] J.Sopena , L.Arantes, M.Bertier, and Pierre Sens. "*A Fault-tolerant Token-based Mutual Exclusion Algorithm Using A Dynamic Tree*". Euro Par.LNCS 3648, 2005

[9] P.chaudhuri, and Tomas Edward. "*An* $O(\sqrt{n})$ *Distributed Mutual Exclusion Algorithm Using Queue Migration*". *Journal Of Universal Computer Science,* Vol.21(No.2):140-159, 2006

[10] S.Paydar, M.Naghibzadeh , and A.Yavari. "*A Hybrid Distributed Mutual Exclusion Algorithm*", IEEE International Conference On Emerging Technologies In Pakistan, November 2006

[11] L.Lamport. "*Times,Clocks,And The Ordering Of Events in a Distributed System*". Communications OF The ACM, Vol.21(No.7): 558-565, July 1978

[12]G.Ricart, and Ashok.K.Agrawala. "*An Optimal Algorithm For Mutual Exclusion in Computer Networks*". *Communication of The ACM*, Vol.24(No.1):9-17, January 1981

[13] S.Lodha, and A.Kshemkalyani. "*A Fair Distributed Mutual Exclusion algorithm*".IEEE Transactions On Parallel And Distributed Systems. Vol.11(No.6), June 2000

[14] R.Atreya , and N.Mittal. "*A Dynamic Group Mutual Exclusion Using Surrogate-Quorums*". Proc,IEEE Int'l Conf. Distributed Computing System, June 2005

[15] M.Maekawa. "*A*$\sqrt{N}$ *Algorithm For Mutual Exclusion In Decentralized Systems*". ACM Transactions On Computer Systems , Vol.3 (No.2):145-159, May 1985

[16] R.Atreya , and N.Mittal. "*a Quorum-based Group Mutual Exclusion Algorithm For A Distrinbuted System With Dynamic Group Set*". IEEE Transactions On Parallel And Distributed Systems, Vol.18(No.10), October 2007

[17] G.Cao , and M.Singhal. "*A Delay-optimal Quorum-based Mutual Exclusion Algorithm For Distributed Systems*". IEEE Transactions On Parallel And Distributed Systems, Vol.12(No.12), December 2001

[18] H.Kakugawa , S.Kamei, and T.Masuzawa. "*A Token-based Distributed Group Mutuaal Exclusion Algorithm With Quorums*", IEEE Transactions On Parallel And Distributed Systems, Vol.19(No.9), 2008

[19] N.Mittal,and P.Mohan."*An Efficient Distributed Group Mutual Exclusion Algorithm For Non-uniform group access*", proceedings Of The IASTED International Conference On Parallel And Distributed Computing And Systems , Phoenix,Arizona,USA, 2005

[20] N.Mittal,and P.Mohan. "*A Priority-based Distributed Group Mutual Exclusion Algorithm When Group Access is Non-uniform*". Journal Of Parallel And Distributed Computing, No.67: 797-815, March 2007

[21] M.Bouillaguet,L.Arantes,and P.Sens. "*Fault Tolerant K-mutual Exclusion Algorithm Using Failure Detector*" . International Symposium On Parallel And Distributed Computing, 2007

[22] P.Chaudhuri, and T.Edward. "*An Algorithm for K-mutual Exclusion In Decentralized Systems*". Computer Communications 31: 3233-3235, 2008

[22] K.Raymond. "A Distributed Algorithm For Multiple Entries To A Critical Section", Information Processing Letters, North-Holland, No.30, February 1989