Modified HS Algorithm in Distributed System for Synchronous Ring

Dinesh Kumar Yadav
Dept. of CS, RIET, Jaipur,

Vijay Kumar Sharma
Dept. of CS, RIET, Jaipur

ABSTRACT
Leader election is the most critical part of any distributed system and also challenging one. By optimizing the performance of leader election, performance of system can be improved. There is already number of algorithms but in this paper there is a proposal of a new Leader Election algorithm for synchronous ring. In this proposal we are trying to tradeoff among time, message and space complexity. By adding new concept Adoption we can proposed a new ring algorithm which performs better as compare to original classic HS algorithm for ring.

Keywords
Distributed algorithms, Leader election, algorithms, complexity, Distributed Computing, Synchronous

1. INTRODUCTION
In distributed system, processors communicate with each other using shared memory or by exchanging messages with each other. For processors to perform any distributed task effectively the processors require coordination. In a pure distributed network, there is no central controlling processor that arbitrates decisions. Without a central authority or coordinator, any processor has to communicate with all processors in the network to make decision. Often during the decision process, not all processors make the same decision. Communication between processors takes time and further more making the decision takes time. Coordination among processors becomes difficult when consistency is needed among all processors. Centralized controlling processors can be selected among the group of available processors to reduce the complexity of decision- making. By having a centralized authority, decisions can be made in a more serialized fashion, which are simpler to execute..

[1]The problem of leader election is fundamental in the operation of distributed systems. A distributed system is a collection of autonomous computing nodes which can communicate with each other and which co-operate on a common goal or task.

[1]A leader performs a centralized co-ordination after being selected. This may be necessary in some problems where a completely distributed solution is either not available or offers less attractive performance. Whenever some failure occurs it is necessary for the nodes to adapt to the new conditions so that they can continue working. This requires some kind of re-organization. Leaders are elected to manage the re-organization

The leader election problem is supposed to be one of the most prominent ones in distributed computing. It consists of appointing a leader process from an initial configuration in which all processes are in the same state, that is, they all are candidates and can become a leader [10]. In distributed computing leader election is the process of designating a single process as the organizer of some task distributed among several computers (nodes). In many cases we need a coordinator in the network for coordination tasks. When this coordinator crashes, we have to select another process as the substitute

Garcia-Molina's classical Bully Algorithm is a prime solution to leader election in synchronous systems. In this paper there is a proposal of a new algorithm for leader election using priority queue . Priority queue is created using the max heap structure and it keeps the ID of all nodes of network in sorted form . Priority Queue will be dedicated with each node (process). Each priority queue is managed by its node(process) separately. It takes fewer messages as well as less time as compare to other popular leader election algorithm like Bully and Ring.

2. RELATED WORK
There are lot of work already have done in the field of leader election. Several leader election algorithms such as Ref. [6] the Bully algorithm ,[7]Ring algorithm,[8]Chang and Robert' s algorithm ,[9] Peterson's algorithm,[10] LeLann's algorithm, and [11] Franklin's algorithm have been proposed over the years. These algorithms, however, require nodes to be directly involved in leader election. Information is exchanged between nodes by transmitting messages to one another until an agreement is reached. Once a decision is made, a node is elected as the leader and all the other nodes will acknowledge the role of that node as the leader.

[2]A new approach has been proposed for leader election using heap tree method. In this approach formal heap tree method is used.

3. PROPOSED APPROACH
3.1 System Assumptions
Various leader election algorithms have been already proposed. Some examples of Ring algorithms are LCR algorithm, HS algorithm, Peterson algorithm etc. Every algorithm has certain assumption and according to that they are electing leaders by communicating through messages.

In this proposal, there is a concept of adoption. According to this concept the nodes which is not in the race of leader, will adopt the largest ID for this leader election, the node has seen in the process of leader election. The number of message passing can be reduce drastically during leader election

1. Bidirectional communication
2. Ring Size is unknown
3. UIDs with comparison only

3.2 Proposed Approach of Leader Election

In this proposal, Successive doubling strategy: It is used in many distributed algorithms where network size is unknown. Each process sends a UID token in both directions to successive greater distances (double each time).

Going outbound
Token is discarded if it reaches a node whose UID is bigger.

Going inbound
Everyone passes the token back. Process begins next phase only if when it gets both its tokens back. Process that gets its own token in outbound direction elects itself as a leader.

Modification
There will be two categories of the nodes. Active and Non Active; Process which is still in the race of leader are called Active Process and the Process which is not in the race of leader is called Non-Active.

Adoption
A non-Active process will perform the adoption concept during this leader election It will adopt the largest UID as its own UID, which this node has seen till that time and during this leader election it will use a adopted value for various comparisons which would be performed. During the leader election the process will be Non-Active.

In the case of active process it will work normally and adoption will not follow. So adoption case will follow only in the case of Non-Active process.

4. PROPOSED ALGORITHM

1. Only active processes will be in the race of leader election. Initially each process is active process. So each active process in every phase l sends a UID token in both directions to successively greater distances 2^l for leader election.(double each time for l=0,1,2,3……, where l is a phase). If active process receives its own UID from both direction then it remain active process and proceeds to next phase for leader election.

If active process does not receive its own UID from both directions it becomes non active process and now it performs adoption.

2. If Going outbound then Token will be sending to next process (node) according to hop count if received token (UID) is greater than receiving process (node) UID and after this receiving node will become non active process and perform adoption. Token is discarded if it reaches a (Active/Non-Active) node whose UID is less than received UID. If equal then there may be two cases either it will be non-active node or active node.

i. The case of non-active it will perform according to hop count either send message to next node or sent it back.

ii. In 2nd case in the case of active process goto last step (3), where active process, which received its own token in outbound direction, elects itself as a leader.

3. If In inbound then every process will relay the received UID to its originating process.

4. When a process received a UID which is equal to its own UID, in outbound direction then process elects itself as a leader.

ring).

These are steps of proposed algorithm named as Modified H.S. algorithm. It is basically the modified version of old H.S. algorithm.
As does the LCR algorithm, the HS algorithm elects the process with the maximum UID. Now every process, instead of sending its UID all the way around the ring as in the LCR algorithm, sends it so that it travels some distance away, then turns around and comes back to the originating process. It does this repeatedly, to successively greater distances. The HS algorithm proceeds this way.

In this proposal ,there are few modifications have done which produce great results in terms of message passing and It performances better for larger value of n (number of processes in ring).

5. ANALYSIS OF PROPOSED ALGORITHM

Complexity of Proposed algorithm (Modified HS Algorithm)

Complexity is the parameter which used to analysis any algorithm for the increasing value of n. For distributed system algorithm message complexity is one of the major factor to analysis any distributed system algorithm.

Time complexity: The last round is the major and deciding factor for time complexity of this proposed algorithm, which is equal to 4n,so time complexity will be same as original HS algorithm O(n).

Message Complexity: Message complexity is the key, which reduced and reduced more for larger value of n . Message is largely depends on number of active process in each phase and in proposed algorithm, the active processes are reduced due to the adoption concept, and this is the main reason why this algorithm perform better as compare to old HS algorithm and perform even more improves for larger value of n .So, The message complexity of proposed algorithm will be θ(nlogn) as compare to HS algorithm O(nlogn).

6. CONCLUSION AND FUTURE WORK

Selection of leader election algorithm in a distributed system plays a vital role in the performance of the system and there should be a proper trade-off among space, time and message complexity. Selection of leader election algorithm in a synchronous ring of distributed system plays a vital role in the performance of the system and there should be a proper tradeoff between time and message complexities. The proposed algorithm of synchronous ring is an attempt to improve leader election algorithm (using adoption method) by using modified HS algorithm. Lesser complexity and less number of message send in modified HS method shows that it will perform better than earlier proposed algorithms in ring network.

It has already proved that Modified HS perform better than HS for large number of data items. By using proposed algorithm proper balance between time and message can be obtained. Proposed algorithm performance improves for large value of n and messages sent during leader election is quite less from earlier proposed algorithms and this proposed algorithm perform even better in term of message complexity as compare to earlier algorithms of synchronous ring.

As similar to the original H.S. algorithm, the new proposed Modified H.S. algorithm also works without modification in

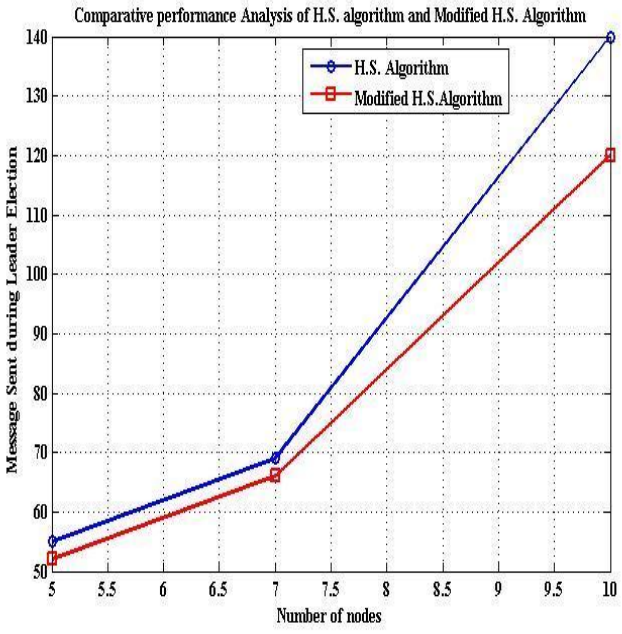
the version of the synchronous model with the variable start time.

In future fault tolerance, leader failure and other features can also be added and this approach can be used in ad hoc and sensor environment

In future, it will be also a subject of research that how the message complexity will be affected if the sequence of connected nodes in a ring network changed, and if affected then at which sequence the nodes should be arranged for best and worst message complexity

Table 1 Performance of HS and Proposed Algorithms in Leader Election

Number of processes(nodes) n	HS algorithm (Message sent)	Modified HS algorithm (proposed algorithm) (Message sent)
5	55	52
7	69	66
10	140	120
20	360	280
30	500	390



6. REFERENCES

[1] Princy Francis and Sanjeev Saxena ,IEEE conference ,1998,Optimal Distributed Leader Election Algorithm For Synchronous complete Network .

[2] Mohammad Reza EffatParvar , Nasser Yazdani, Mehdi EffatParvar , Aresh Dadlani and Ahmad Khonsari,IEEE conference,2010, Improved Algorithms for Leader Election in Distributed Systems .

[3] Vandana Sharma, Parvinder S. Sandhu, Satwinder Singh, and Baljit Saini, World Academy of Science, Engineering and Technology 42, 2008,Analysis of Modified Heap Sort Algorithm on Different Environment

[4] Xio Dong Wang, Ying Jie Wu, Journal of Computer Science and Technology. 22(6): 898-903 An improved heap sort algorithm with nlogn –0.788928n comparisons in worst case

[5] McDiarmid C J H, Journal of Algorithms, 1989, 10(3): 352~365,Reed B A. Building Heaps Fast.

[6] H. Gracia-Molina, IEEE Trans. on Computers, vol. C-31, no. 1, Jan. 1982 "Elections in a distributed computing system"

[7] N. Fredrickson and N. Lynch,Journal of ACM, vol. 34, no. 1, pp. 98-115", 1987 "Electing a leader in a synchronousring"

[8] E. Chang and R. Roberts, Communications of the ACM, vol. 22, no. 5, pp.281-283, May 1979 "An improved algorithm for decentralized extrema-finding in circular configurations of processes".

[9] G. L. Peterson, ACM Trans. Programming Languages and Systems, pp. 758-762, Oct. 1982 "An O(n log n) unidirectional algorithm for the circular extrema problem".

[10] G. LeLann, Information Processing Letters, pp. 155-160, 1977 "Distributed systems - towards a formal approach".

[11] W. R. Franklin, Communication of the ACM, pp. 336-337, 1982 "On an improved algorithm for Misra and Kate N. Nagaraj,"Security in Wireless Ad Hoc

[12] H. GaFcia-Molina." IEEE Trans. on Computers, Vol C-31, Jan 1982, 48-59 "Elections in a Distributed Computing System"

[13] G. Fredrickson and N. Lynch, in Proc. 16th ACM Symp. on Theory of Computing, Washington, USA, pp. 493-503, 1984 "The impact of synchronous communication on the problem of electing a leader in a ring"

[14] E. Korach, S. Moran, and S. Zaks, in Proc. 3rd ACM Symp. on Principles of Distributed Computing, Vancouver, Canada, pp. 199-207,Aug. 1984, "Tight lower and upper bounds for some distributed algorithms for a complete network of processors".

[15] Gonnet G H, Munro J I, 1986, 15(6): 964-971, Heaps on Heaps. SIAM Journal on Computing.

[16] P. M. B. Vitanyi, "Distributed election in an Archimedean ring of processors", USA, pp. 542-547, 1984,in Proc. 16th ACM Symp. on Theory of Computing, Washington

[17] Thomas H. Cormen,Charles E. Leiserson, Ronald L. Rivest, Clifford Stein,"Introduction to Algorithms,Second Edition"