ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2016.0197

A DISTRIBUTED RING ALGORITHM FOR COORDINATOR ELECTION IN DISTRIBUTED SYSTEMS

Shaik Naseera
School of Computing Science and Engineering, VIT University, Vellore, India
E-mail: naseerakareem@gmail.com

Abstract

In distributed systems, nodes are connected at different geographical locations. As a part of effective resource utilization, the data and resources are shared among these nodes. A leader or pioneer is necessary to take care of this resource sharing process by eliminating conflicting among the nodes. The shared resources are to be accessed in a fair and optimal manner among all the nodes in the network. This makes the importance of electing a leader which can coordinate with all the nodes and make fair use of resources among the nodes. As nodes are distributed in different geographical locations and factors influencing its operation make it inevitable that a leader may go down temporarily or permanently. In such case a new leader has to be elected for coordination. The time taken to elect a new leader is one of the crucial factors in improving the performance of the system. In this paper, we propose a new approach for leader election to optimize the time taken for the nodes to elect the leader.

Keywords:

Leader Election, Concurrent Processing, Clustering, Distributed Environment

1. INTRODUCTION

A special hub is required for perfect correspondence among each one of the nodes, it is called “pioneer” or the “leader”. Enrolling a leader is a most fundamental kind of symmetry breaking. The nodes which rely on upon leader races can't show abnormal state of parallelism; consequently these counts encounter the evil impacts of poor time versatile quality. Regardless, from time to time it is still profitable to have a leader to settle on essential options simply.

The system of picking a leader is known as leader race. Regardless of the way that leader race is a clear sort of symmetry cracking, there are some astounding issues that allow us to show striking computing models. Here we take our area on the ring topology. The ring is the “drosophila” of appropriated handling the same number of entrancing challenges starting now reveals the establishment of the issue in the remarkable occurrence of the ring. Giving cautious thought to the ring also nodes well from a sensible area of concentration as some genuine systems rely on upon a ring topology, for example, the token ring standard for neighbourhood.

In the leader race issue, a few nodes, each distinguished by an ID, attempt to choose a hub as a leader. At first (time zero) all nodes are inert and don't not mindful of others. After now and then, a few nodes get to be dynamic and begin to choose a leader. In the fruitful finish of the race, stand out hub chose as a leader and the nodes contributed in the race frame a gathering.

The main aim of all active researches is to select a leader with superior performance. This important problem is mentioned on various topologies and on each topology, the researchers try to use characteristics of topology to solve these problems

2. RELATED WORK

Authors in [1] proposed a leader election protocol for complete networks with a sense of direction. Sense of direction provides nodes the capability of distinguishing between their incident links according to a global scheme.

Garavel et al. [2] proposed a specification and verification of distributed leader election algorithm for a unidirectional ring network. Soundarabai et al. [10] proposed a message efficient leader election algorithm for distributed systems.

Zargarnataj proposed a leader election algorithm which uses an assistant for taking the responsibilities of a leader in case of leader crash. The assistant has a significant responsibility when the size of the network increases [3]. Authors in [7] proposed a new leader election algorithm in which a set of alternatives will be available to take the responsibilities of leader in case of leader crash.

Effat Parvar et al. [7] proposed an improved version of Bully and Ring leader election algorithms in which a heap tree mechanism is used for electing a new coordinator. Authors in [8, 9] have proposed a leader election algorithm for adhoc and colored networks in which the node changes its location.

P Beaulah Soundarabai [11] presents a substitute type of appropriated ring count. This incorporates extensive changes of the present circled ring race count and the relationship of message multifaceted design with the main ring choice estimation. Diversion results show that the displayed estimation minimizes the amount of messages being exchanged picking the leader of the ring. The maker has proposed count which can minimize the amount of messages passed when the leader must be picked in the ring. This proposed count is beneficial in diminishing the amount of messages passed.

In the Ring topology (both bidirectional and unidirectional) the essential issue is hub disappointments (this issue existing in different topologies however in the Ring it is more persuasive). The issue causes a hub can't effectively take an interest in the decision and now and again drives nodes, which can’t impart straightforwardly to others to be in the same gathering [4]. Another critical issue is the loss of messages. Particularly, in the extensive systems may result to pick a wasteful leader. Additionally now and again, perhaps a couple of nodes did not take an interest in the race and in most pessimistic scenario nodes did not get the message of choosing a leader and may run race calculation once more. The ideal leader is the closest hub from all nodes and has the better execution than different nodes in a framework. Let, w_i signifies the aggregate execution at a hub i for the whole framework, and the aggregate evaluated execution indicates w_i. The hub that has $\{w_1$,

w_2, w_3…}, will choose as a leader. After the present leader fizzled, the associate begin to facilitate nodes. The consistency that a right hand turns out to be new leader computed as takes after. The Relation among the disappointment pulse q and the working pulse d of hub r is $d = 1 - q$.

Mehdi Mirakhorli [5] utilizes a few requested nodes as competitor of driving {candid 1, real 2 real to life *k*}. These nodes are utilized for keeping another worldwide race between all nodes. At the point when a hub *L* distinguishes the leader is smashed, it invokes a leader crashed message that maintains the ID of leader that was crashed. In the event that node is up, it tests the ID of leader in got communication with leader ID. On the off chance that those IDs are equivalent, it picks hub *L* as leader. It assigns the ID of hub *L* as latest leader in its current setting and telecasts leader message that contains the ID of hub *L* as latest leader. In the event that two or more nodes at the same time identify the accident of leader and those two perform same occupation, hub 1 picks the hub that its message has been gotten initial one as pioneer. For instance in node1 recognizes the leader is slammed, it sends leader crashed message to candid1 right now, hub 0 at the same time distinguishes the leader is smashed and it sends leader crashed message to hub 1.

3. LEADER ELECTION ALGORITHM

The ring algorithm is regularly credited to LeLann, who formalized it as a strategy to make another leader in a ring algorithm in which the leader has been lost.

In distributed computing, leader race is the strategy of electing out a single methodology as the organizer of some task given among a several nodes of the network. Before the process is begun, all framework node points are either unknown which node point will serve as the "leader" (or pioneer) of the group, or not ready to talk with the present coordinator. After a pioneer race count has been run, in any case, each node point all through the framework sees a particular, stand-out node point as the task of leader.

The entire network nodes pass information on among themselves in order to pick which of them will get into the "leader" state. To accomplish this purpose, they require some methodology with a particular deciding objective to resolve the symmetry between nodes of the network. To quote an example, for every node point has exceptional and proportional characteristics, then the node points can dissect their identities, and pick that the node with the largest identity is the leader.

4. PROPOSED METHODOLOGY

Various nodes are required for shaping a circulated system. Every hub will comprise a specific ID. This is known as the procedure ID of the procedure. This ID is one of a kind in the entire disseminated system. The ID is been whole number worth. All nodes are separated into number of gatherings relying upon the aggregate number of nodes accessible at that specific time.

A few nodes might be slammed or some may leave the system for reasons. So we consider just the nodes that are dynamic at the given time. Partitioning into gatherings should be possible either by utilizing k-implies calculation in light of the separation between the nodes or arbitrarily selecting the nodes and gathering them.

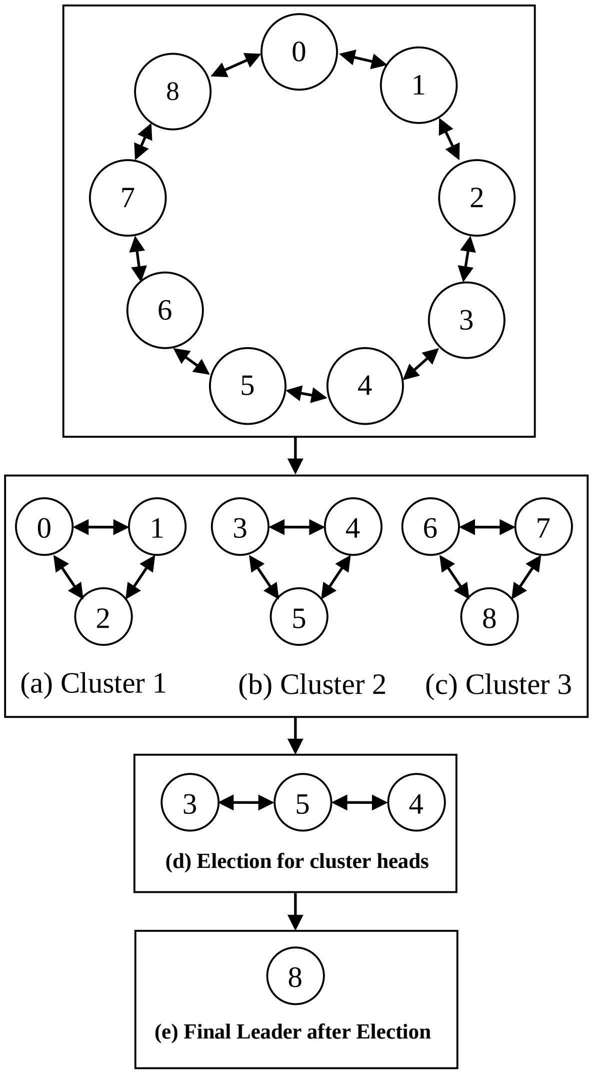

Fig.1. Proposed Distributed Ring Algorithm

We now consider the leader of the gatherings not yet chosen or the current leader has been smashed. So there is a requirement for choosing another leader for the circulated system to perform its election in an ideal way.

Here, decision calculation must be called to choose the new leader of the system. We then utilize the decision calculation for all the individual gatherings in the system, this is executed simultaneously.

At that point every gathering will produce a leader in the meantime for their individual gatherings. For all the chose leaders the decision calculation is prepared among them. Hub that is having the most astounding ID among the gathering leaders will now be chosen as leader for the system. This ID is then coursed to every one of the nodes in the system.

The begin time and end time of this procedure is logged. As in this procedure the time taken for the system to choose a leader is basic and we are making this procedure as an ideal procedure.

Algorithm

1: Create a network of nodes

2: Divide the nodes into sub-groups (clusters) using k-means clustering algorithm.

3: For each cluster in the network, perform the following steps in parallel
  a. Randomly choose a node, and perform the following steps
    i: Send leader message along with its PID to the immediate neighbour
    ii: if neighbour receives a PID > the current PID then forward that message
    iii: else send current PID to immediate neighbour.
  b. When the message reaches to the source node return PID as the group leader.
4: Initiate leader process among the cluster leaders using the ring algorithm.
5: A node with highest PID will be returned as the leader of the network.
6: Circulate the leader information to all nodes in the network.

5. RESULTS AND DISCUSSIONS

The proposed distributed ring leader algorithm has been implemented and simulated using java. We have made various networks for checking the simulation purpose. The nodes are separated into number of clusters relying upon the aggregate number of nodes accessible at that specific time. Division into clusters is made by using k-means clustering algorithm in view of the separation between the nodes or randomly selecting the nodes and clustering them. We then applied the proposed algorithm for all the clusters in the system in parallel. At that point every cluster will elect a leader for their region.

Table.1. Time Taken for Ring Algorithm

S. No.	No. of Nodes	Existing Algorithm	Proposed Algorithm
1	10	5250	4210
2	50	9410	8345
3	100	49880	46254
4	250	78269	71520

The Table.1 gives the time comparison between the existing algorithm and proposed algorithm. The time is given in milli-seconds.

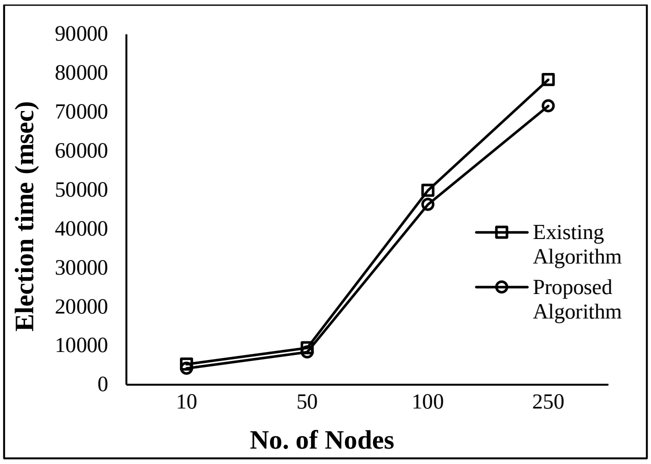

Fig.2. Time Taken by the Algorithms

Table.2. Number of Messages Passed

S. No.	No. of Nodes	Existing Algorithm	Proposed Algorithm
1	10	20	24
2	50	100	109
3	100	200	215
4	250	500	518

From the Table.2 we can observe there is an increase in the number of messages passed through the network. But the time taken to choose the leader is efficient.

For all the leaders of clusters, the leader is processed. Process having the most elevated ID will now be chosen as leader for the system. This ID is then broadcasted to every one of the nodes in the system. The begin time and end time of this procedure is logged. As in this procedure the time taken for the system to choose a leader is basic and we are making this procedure as an ideal procedure.

Let N be number of nodes in the entire network. Let m be number of clusters in the whole network, c_i is the number of nodes in each cluster.

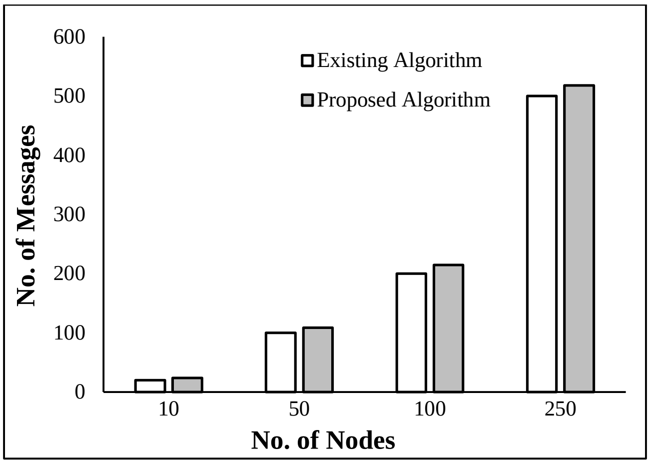

Fig.3. No. of Messages Communicated

The number of messages passed or circulated in the network in parallel using the proposed method is given by $P = \max \{c_1, c_2, c_3 \ldots c_m\}$.

In the conventional ring algorithm, let N be the number of nodes in the entire network. So the number of messages passed will be N. From the results we can show that $P < N$. So time taken to complete the message passing is less for the proposed method.

6. CONCLUSIONS

We have proposed a new distributed ring algorithm for coordinator election for a distributed system. The algorithm is an extension of conventional ring algorithm. The performance of the algorithm is tested on a simulated distributed network and performance comparison is made between the general and the proposed algorithms. The proposed algorithm is fault tolerance and also exhibits better performance over the conventional ring algorithm. The proposed algorithm elects the new coordinator in a distributed manner and the results shows that the time taken by the proposed technique is less compared to the original ring algorithm.

REFERENCES

[1] Gurdip Singh, "Efficient Leader Election using Sense of Direction", *Distributed Computing*, Vol. 10, No. 3, pp. 159-165, 1997.

[2] Hubert Garavel and Laurent Mounier, "Specification and Verification of Various Distributed Leader Election Algorithms for Unidirectional Ring Networks", *Science of Computer Programming*, Vol. 29, No. 1-2, pp. 171-197, 1991.

[3] Mahdi Zargarnataj, "New Election Algorithm based on Assistant in Distributed Systems", *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications*, pp. 324-331, 2007.

[4] G. Singh, "Efficient Distributed Algorithms for Leader Election in Complete Networks", *Proceedings of 11th International Conference on Distributed Computing Systems*, pp. 472-479, 1991.

[5] Mehdi Mirakhorli, Amir Azim Sharifloo and Maghsoud Abbaspour, "A Novel Method for Leader Election Algorithm", *Proceedings of 7th IEEE International Conference on Computer and Information Technology*, pp. 452-456, 2007.

[6] M. Gholipour, M.S. Kordafshari, M. Jahanshahi and A.M. Rahmani, "A New Approach for Election Algorithm in Distributed Systems", *Proceedings of 3rd International Conference on Communication Theory, Reliability, and Quality of Service*, pp. 70-74, 2009.

[7] Mohammad Reza Effat Parvar, Nasser Yazdani, Mehdi Effat Parvar, Aresh Dadlani and Ahmad Khonsari, "Improved Algorithms for Leader Election in Distributed Systems", *Proceedings of 2nd International Conference on Computer Engineering and Technology*, Vol. 2, pp. 2-6, 2010.

[8] Dariusz R. Kowalskia and Andrzej Pelc, "Leader Election in Ad Hoc Radio Networks: A Keen Ear Helps", *Journal of Computer and System Sciences*, Vol. 79, No. 7, pp. 1164-1180, 2013.

[9] Dariusz R. Kowalskia and Andrzej Pelc, "Topology Recognition and Leader Election in Colored Networks", *Theoretical Computer Science*, Vol. 621, pp. 92-102, 2016.

[10] P. Beaulah Soundarabai, J. Thriveni, H.C. Manjunatha, K.R. Venugopal and L.M. Patnaik, "Message Efficient Ring Leader Election in Distributed Systems", *Computer Networks and Communications*, Vol. 131, pp. 835-843, 2013.

[11] Michel Raynal, "*Distributed Algorithms for Message-Passing Systems*", 1st Edition, Springer, 2013.