IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727Volume 8, Issue 1 (Nov. - Dec. 2012), PP 23-27
www.iosrjournals.org

Study of various Election algorithms on the basis of message-passing approach

Pooja B. Raval[1], Sanjay M. Shah[2]
1(PG Student of Computer Engineering, Merchant Engineering College, Basna, Visnagar, Gujarat, India)
2(Associate Professor, CSE Department, Government Engineering College, Sector28, Gandhinagar, Gujarat, India)

Abstract: *An important challenge in distributed systems is the adoption of suitable and efficient algorithms for coordination selection. The leader election is important problem in distributed system as data is distributed among different node which is geographically separate. For maintaining co-ordination between the node, leader node have to be selected. The main role of an elected coordinator is to manage the use of shared resources in optimal manner. This paper represents the different election algorithms with their limitations as well comparative analysis of them, efficiency in terms of number of messages exchanged in each case and the complexity of various coordinator selection algorithms in distributed systems.*
Keywords: *Distributed system, Election, Coordinator, Priority*

I. Introduction

In a distributed computing system, a node is used to coordinate many tasks. It is not an issue which node is doing the task, but there must be a coordinator that will work at any time. An election algorithm is an algorithm for solving the coordinator election problem. Various algorithms require a set of peer nodes to elect a leader or a coordinator. Elections may be needed when the system is initialized, or if the coordinator crashes or retires [2].

A **Distributed system** is a collection of autonomous computing nodes which can communicate with each other and which cooperate on a common goal or task.

Tanenbaum and van Renesse: A **distributed system** is one that looks to its users like an ordinary, centralized, system but runs on multiple independent CPUs.[2]

A distributed system is a collection of processors interconnected by a communication network in which each processor has its own local memory and other peripherals and the communication between them is held by message passing over the communication network. [1]

A. Features of Distributed System

1. Inherently distributed applications
2. Information sharing among distributed users
3. Resource sharing
4. Better price performance ratio
5. Shorter response times and higher throughput
6. Higher reliability
7. Extensibility and incremental growth
8. Better flexibility in meeting users needs

B. Need of Election

Several distributed algorithms require that there be a coordinator node in the entire system that performs some type of coordination activity needed for the smooth running of other nodes in the system. As the nodes in the system need to interact with the coordinator node, they all must unanimously who the coordinator is. Also if the coordinator node fails due to some reason (e.g. link failure) then a new coordinator node must be elected to take the job of the failed coordinator.

II. Different Election Algorithms

A. Bully Algorithm by Garcia Molina [2][7]

Bully algorithm is one of the most famous election algorithms which were proposed by Garcia-Molina in 1982.

Assumptions:

1. It is a synchronous system and it uses timeout Mechanism to keep track of coordinator failure detection.

2. Each node has a unique number to distinguish them.
3. Every node knows the node number of all other nodes.
4. Nodes do not know which nodes are currently up and which nodes are currently down.
5. In the election, a node with the highest node number is elected as a coordinator which is agreed by other alive nodes.
6. A failed node can rejoin in the system after recovery.

In this algorithm, there are three types of message and there is an election message (ELECTION) which is sent to announce an election, an answer (OK) message is sent as response to an election message and a coordinator (COORDINATOR) Message is sent to announce the new coordinator among all other alive nodes . When a node P determines that the current coordinators crashed because of message timeouts or failure of the coordinator to initiate a handshake, it executes bully election.

Algorithm using the following sequence of actions

1. P sends an election message (ELECTION) to all other nodes with higher node numbers respect to it. If P doesn't receive any message from nodes with a higher node number than it, it wins the election and sends a COORDINATOR Message to all alive nodes.
2. If P gets answer message from a node with a higher node number; P gives up and waits to get COORDINATOR message from any of the node with higher node number. Then new node initiates an election and sends ELECTION message to nodes with higher node number than that one. In this way, all nodes will give up the election except one which has the highest node number among all alive nodes and it will be elected as a new coordinator.
3. New Coordinator broadcasts itself as a coordinator to all alive nodes in the system.
4. Immediately after the recovery of the crashed node is up, it runs bully algorithm.

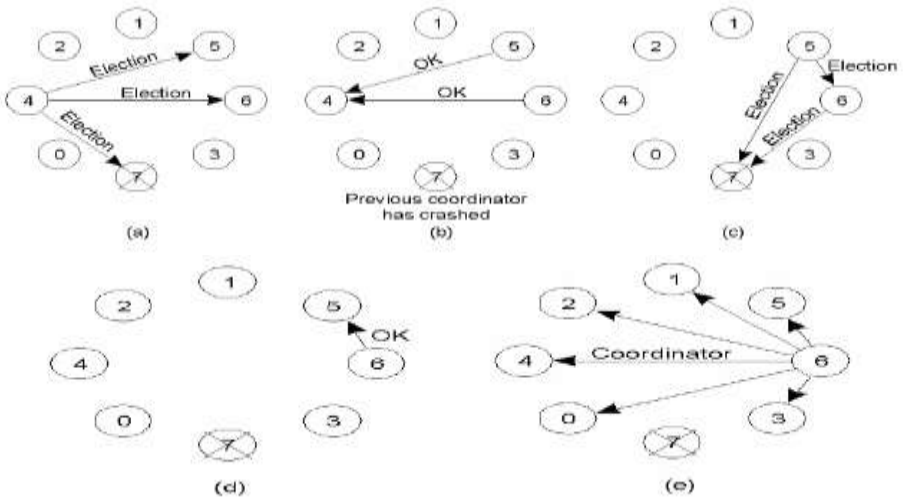

Figure 1: Bully algorithm example (a) process 4 detects coordinator is failed and holds an election, (b) process 5 and 6 respond to 4 to stop election, (c) each of 5 and 6 holds election now, (d) process 6 responds to 5 to stop election, (e) process 6 winds and announces to all.

Bully algorithm has following limitations:

1. The main limitation of bully algorithm is the highest number of message passing during the election and it has order O (n2) which increases the network traffic.
2. When any node that notices coordinator is down then holds a new election. As a result, there May n number of elections can be occurred in the system at a same time which imposes heavy network traffic.
3. As there is no guarantee on message delivery, two nodes may declare themselves as a coordinator at the same time. Say, P initiates an Election and didn't get any reply message from Q, where Q has a higher node number than P. At that case, p will announce itself as a coordinator and as well as Q will also initiate new election and declare itself as a coordinator if there is no node having higher node number than Q.
4. Again, if the coordinator is running unusually slowly (say system is not working properly for some reasons) or the link between a node and a coordinator is broken for some reasons, any other node may fail to detect the coordinator and initiates an election. But the coordinator is up, so in this case it is a redundant election. Again, if node P with lower node number than the current coordinator, crashes and recovers again, it will initiate an election from current state.

B.Ring algorithm by Silberchats and Galvin [2]

Assumptions

1. All the nodes in the system are organized as a logical ring.
2. The ring is unidirectional in the nodes so that all the messages related to election algorithm are always passed only in one direction.

Algorithm using the following sequence of actions

While the message circulates over the ring, if the successor of the sender nodes is down the sender can skip over successor, or the one after that until an active member is located.

When a node n1 sends a request message to the current coordinator and does not receive a reply within a fixed timeout period, it assumes that the coordinator has crashed. So it initiates an election by sending an election message to its successor. This message contains the priority of node n1. On receiving the election message, the successor appends its own priority number to the message and passes it on to the next active member in the ring.

In this manner, the election message circulates over the ring from one active node to another and eventually returns back to node n1. Node n1 recognizes the message as its own election message by seeing that in the list of priority numbers held within the message the first priority number is its own.

Among this list, it elects the node with the highest priority as the new coordinator and then circulates a coordinator message over the ring to inform the other active nodes. When the coordinator message comes back to node n1, it is removed by node n1.

When a node n2 recovers after failure, it creates an inquiry message and sends it to its successor. The message contains the identity of node n2. If the successor is not the current coordinator it simply forwards the enquiry message to its own successor. In this way, the inquiry message moves forward along the ring until it reaches the current coordinator. On receiving the inquiry message, the current coordinator sends a reply to node n2 informing that it is the current coordinator.

C.Modified Bully Algorithm [6]

Modified Bully algorithm, an efficient version Bully algorithm to minimize redundancy in electing the coordinator and to reduce the recovery problem of a crashed process.

Assumptions:

There are five types of message. An election message is sent to announce an election, an ok message is sent in response to an election message, on recovery, a process sends a query message to the processes with process number higher than it to know who the new coordinator is, a process gets an answer message from any process numbered higher than it in response to a query message and a coordinator message is sent to announce the number of the elected process as the new coordinator.

Algorithm:

a) When a process p notices that coordinator is down, it sends an election message to all processes with higher number. If no response, p will be the new coordinator.

b) If p gets ok message, it will select the process with highest process number as coordinator and send a coordinator message to all process.

c) When a crashed process recovers, it sends query message to all process with higher process number than it.

d) And if it gets reply then it will know the coordinator and if it doesn't get any reply it will announce itself as a coordinator.

The limitations are given below:

a) On recovery, it sends query message to all processes with higher process number than it, and all of them will send answer message if they alive. Which increases total number of message passing and hence it increases network traffic.

b) It doesn't give guarantee that any process p will receive only one election message from processes with lower process number. As a result there may be q different processes with lower process number can send election message to p and p will send ok message to all of them. This increases number of election and also number of message passing.

c) It doesn't give any idea if p will crash after sending an election message to all processes with higher process number.

d) It also doesn't give any idea if a process with the highest process number will crash after sending ok message to p.

D.Election algorithm using election commission [6]

Algorithm:

a) When process P notices that the coordinator is down, it sends an ELECTION message to Election Commission.

b) FD of Election Commission verifies ELECTION message sent by P. If the sending notice of P is not correct, then Election Commission will send a COORDINATOR message to P with process number of the current coordinator.

c) If the sending notice of P is correct and if the highest process number is P, then Election Commission will send a COORDINATOR message to all processes with process number of P as a new coordinator. If the highest process number is not P, Election Commission will simply find out the alive process with the highest process number using HP and sends a COORDINATOR message to all processes with the process number of that process as a new coordinator.

d) If any process including last crashed coordinator is up, it will send a QUERY message to the Election Commission. If the process number of the newly entranced process is higher than the process number of the current coordinator, Election Commission will send a COORDINATOR message to all processes having the process number of new coordinator. e) If not, Election Commission will simply send a COORDINATOR message to newly entranced process having process number of the current coordinator.

f) If more than one process sends ELECTION message to Election Commission at the same time, then Election Commission will consider the process with higher process number which ensure less message passing to find out the highest process number using HP.

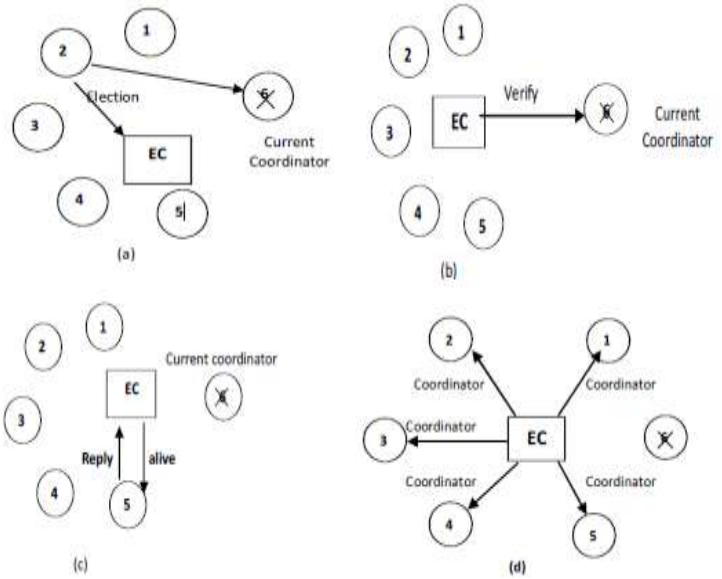

Figure 2 : Election Procedure: (a) Process 2 detects current coordinator is down and sends an election message to EC, (b) EC verifies either the coordinator is really down or not,(c) EC finds the alive process with highest number using alive message,(d) EC sends coordinator message to all process having process number of currently won.

III. Comparisons Of Different Algorithms

In **Bully algorithm[7]**, when the process having the lowest priority number detects the coordinator's failure and initiates an election, in a system of n processes, altogether (n-2) elections are performed. All the processes except the active process with the highest priority number and the coordinator process that has just failed perform elections. So in the worst case, the bully algorithm requires O(n2) messages. When the process having the priority number just below the failed coordinator detects failure of coordinator, it immediately elects itself as the coordinator and sends n-2 coordinator messages. So in the best case, it has O(n) messages.

During recovery, a failed process must initiate an election in recovery. So once again, Bully algorithm requires O(n2) messages in the worst case, and (n-1) messages in the best case.

In **ring algorithm**, on the contrary, irrespective of which process detects the failure of coordinator and initiates an election, an election always requires 2(n-1) messages. (n-1) messages needed for one round rotation of the ELECTION message, and another (n-1) messages for the COORDINATOR message.

During recovery, a failed process does not initiate an election on recovery, but just searches for the current coordinator. So ring algorithm only requires n/2 messages on average during recovery. [3]

For the case of **modified bully algorithm [3]** there will be need of or O(n) message passing between processes. In worst case that is the process with lowest process number detects coordinator as failed, it requires 3n-1 message passing. In best case when p is the highest process number, it requires (n-p) + n messages.

For the case of **election algorithm with election commission[5]** there will be need of 1 election message to inform EC, 2 verify message to ensure the failure of coordinator, and say r is the highest alive process then alive and reply message to find out the highest alive process and so total or O (n) message passing between processes. If the process with lowest process number detects coordinator as failed it will not change total message. In worst case it may happen that our algorithm needs to check up process to p+1 to find out highest alive process. Only at that case it requires message passing between processes. However, in best case, our algorithm may find the highest alive process with only one alive and one reply message that is highest alive process in the system is process with process number n-1. In that case, our algorithm requires only 1+2+2+n messages. When p is the highest process number, it requires only 1+2 + n messages.

If a process crashes and recovers again, it sends a query message to all processes higher than that process to know the current coordinator which requires 2*(n-p) message passing. But in our algorithm, any process after recovery will only send query message to EC and EC will send a coordinator message having process number of current coordinator which requires only 2 messages passing.

IV. Conclusion

In this paper, we have shown the comparison between various election algorithms in a distributed system. The comparison is done on the basis of number of messages passed and time complexity parameters of algorithms. Some algorithms overcome the overhead the sending of number of messages. This paper also focuses on limitations of different algorithms for coordinator selection.

References

[1] Sinha P.K, Distributed Operating Systems Concepts and Design, Prentice-Hall of India private Limited, 2008.
[2] Tanenbaum A.S Distributed Operating System, Pearson Education, 2007.
[3] Heta Jasmin Javeri, Sanjay Shah ―"A Comparative Analysis of Election Algorithm in Distributed System‖ IP multimedia communication", a special issue from *IJCA*
[4] Sachi Choudhary, Dipesh Sharma―"A Comparative Analysis in Terms of Message Passing & Complexity of Different Coordinator Selection Algorithms in Distributed System", a special issue from *IJARCET*, Volume 1, Issue 7, September 2012
[5] Deepali P. Gawali―"Leader Election Problem in Distributed Algorithm", a special issue from *IJCST Vol*. 3, Issue 1, Jan. - March 2012
[6] Muhammad Mahbubur Rahman, Afroza Nahar―"Modified Bully Algorithm using Election Commission"
[7] H.Garcia Molina "Election in a distributed Computing System". IEEE Trans. Comp, 1982, vol31, no. 1, pp. 48-59. 1982