DS PROJECT REPORT

Bully Leader Election

Pranav Kamojjhala 201501213

Overview:

Bully Algorithm is an algorithm found in distributed computing. The algorithm is used to elect a leader or a coordinator from a group of distributed computer processes. This algorithm was presented by Hector Garcia-Molina in 1982 where he also described the invitation algorithm.

Assumptions:

- Each process has a unique number(ID) to distinguish them.
- Each process knows the process ID of all other processes.
- In the election, the process with the highest process ID is elected as the coordinator.
- A crashed process can rejoin the system after recovery.
- The model is time bounded

.

The Algorithm:

The algorithm uses the following message types:

- Election Message: Sent to announce election.
- Answer (Alive) Message: Responds to the Election message.
- Coordinator (Victory) Message: Sent by winner of the election to announce victory.

When a process P recovers from failure, or the failure detector indicates that the current coordinator has failed, P performs the following actions:

1. If P has the highest process id, it sends a Victory message to all other processes and becomes the new Coordinator. Otherwise, P broadcasts an Election message to all other processes with higher process IDs than itself.
2. If P receives no Answer after sending an Election message, then it broadcasts a Victory message to all other processes and becomes the Coordinator.
3. If P receives an Answer from a process with a higher ID, it sends no further messages for this election and waits for a Victory message. (If there is no Victory message after a period of time, it restarts the process at the beginning.)

4. If P receives an Election message from another process with a lower ID it sends an Answer message back and starts the election process at the beginning, by sending an Election message to higher-numbered processes.
5. If P receives a Coordinator message, it treats the sender as the coordinator.

Analysis:

Bully algorithm has following disadvantages.:-

- It is required that every process should know the identity of every other process in the system which results in large memory usage.
- It has high number of messages being passed during communication which causes heavy traffic . The message passing is of the order $O(n^2)$ in the worst case scenario.

Bully algorithm has a best case message passing of $\Omega(n-1)$ where as ring leader election has a constant 2n messages being passed. There are improved versions of the original Bully Algorithm described in Garcia-Molina's paper. These try to address the time bounding issues as well as the memory usage. The Bully Algorithm described by Garcia-Molina was one of the most widely used election algorithms and is still one of the most prominent ones used to date.

We get the worst case when the node initiating the algortihm is the node with the lowest ID. This means the said node would have to trigger n-2 elections which would mean roughly n^2 messages being passed overall.

We get the best case when the node initiating the election is the node with the highest ID. The election happens immediately because all other nodes have lower priority. The total number of messages being passed is n-1.

Project Scope and Implementation Details:

In this project I have done a simple python implementation of the original Bully algortithm. In order to set up servers for communication i have used the zerorpc library. Different instances of the terminal are connected to diffrent ports to simulate the algorithm. This algorithm has been implemented assuming a synchronous system.

Link to Github repo containing code for implementation:
https://github.com/IceIceRabbit/Bully_leader_algorithm