Parallel Programming INF-3200

Assignment 2

Peer-to-Peer leader election

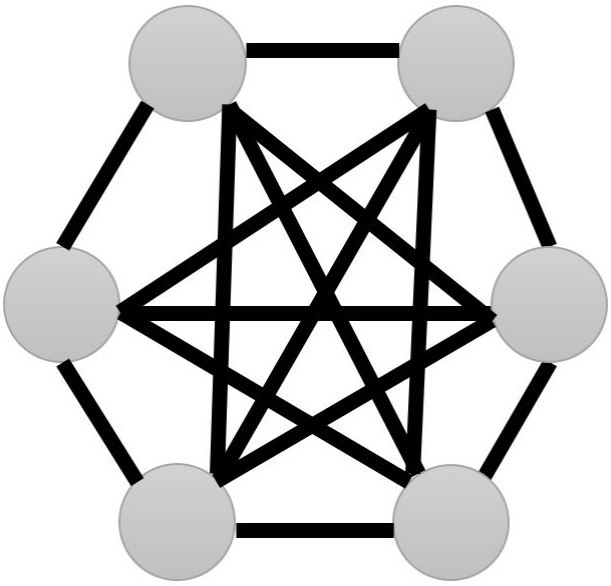
1.0 Introduction

This report describes the design and implementation of a simple Peer-to-peer leader election algorithm used within a number of nodes in a cluster, we will look at how the algorithm is implemented as well as different fault tolerances.

2.0 Technical Background

When using a distributed system it is normal to assign a leader, a leader serves as a organizer and distributes tasks among the different nodes, When the nodes are booted the nodes are unaware of who the leader is, thus different leader election algorithms are used of various complexity, such as a ring network, mesh network, complete network, and so on.

3.0 Design



For this implementation i chose to go for a rather simple design i chose a variation of the bully algorithm, the bully implementation uses a complete network structures, where all nodes are connected to each other, for the author's implementation all the nodes are made known of which nodes exist, and the network is more or less set at start, however after a get command is sent every node checks their connected nodes and return only the ones alive, there is also an nextnode variable in place for more functionality. [1]

The bully algorithm itself uses a hierarchy for the nodes in the system, where the coordinator is a master to the slaves, hench the name Bully, each node is assigned a number, and the process with the highest ID is always chosen as the coordinator. Whenever a node wants to check on its coordinator a get message is sent, if the return message is positive everything is good and well. However if the return message is negative it means that the coordinator node has terminated and a new coordinator has to be elected. The slave then sends a election broadcast to all the nodes that has a higher value, when these nodes gets the message they also send get commands to their higher nodes until there's only one left. if however none of the higher nodes respond it means they've also been terminated and the new coordinator is the node that initially sent out the message.

4.0 Implementation

This simple Distributed key-Value Store was implemented using Python and compiled in the Terminal.

5.0 Discussion

As mentioned in the discussion part the implementation uses the bully algorithm, the fault testing for this algorithm was done by running different nodes: 1-1, 1-2, 3-2, and checking if the leader election and next nodes were correct, which signals a correct implementation of the leader election, however this implementation does not broadcast, as i could not figure out the specifics of it in the limited amount of time, rather every node checks individually for a new leader if the current one is dead. In the initial implementation the leader node is set to null, to showcase the leader algorithm.
Not having broadcasting hurts the flexibility of the system, providing a lot more bytes to be sent than necessary, in a large scale cluster this would have to be fixed for optimal results. By showcasing that a new leader can be elected reliably the implementation also supports leaving nodes. However if a node with the ID of 2 is the current leader, and a new node joins with the ID 10, a new leader is not elected before node 2 dies, if the leader node dies the node with the highest id will be chosen, and in this case id 10. I have implemented both a next node system and a complete network system, the next node system has a node only point to one specific node, always being the higher Id node, if none are available the lowest node is chosen. While the complete network system checks the return value for every other node, if they return True they are included in the return value. I tested the next node part and the complete system by having a number of nodes available, such as 1-1, 3-2, 4-1, and they all pointed to each other and the next node was always correct. When running three nodes and a fourth comes online each of the already active nodes automatically links to it.shutting down nodes, everything is already implemented just not the respectable function, as i could

only get “ungraceful” shutdowns. For adding nodes outside of the array a PUT command could be sent to all the nodes already in the list and it would work.

6.0 Conclusion

In this report the author has shown how the simple Leader election algorithm is implemented, it is fully working on both lab computers and at home.

7.0 References

Pictures:

[1] https://en.wikipedia.org/wiki/Leader_election

Books:

Distributed new international Edition, Distributed systems principles and paradigms, Andrew S. Tanenbaum, Maarten Van Steen - Second edition