Report 4: A Group Membership Service

Lakshmi Srinidh Pachabotla

October 9, 2024

1. Introduction

The main purpose of this assignment is to implement a Group Membership Service (GMS) that provides atomic multicast with consistent communication, leader election, and fault tolerance. One node serves as the group's leader and oversees multicasting messages to other nodes (workers) in the group. When a leader fails, the system chooses a new one while making sure that the states for all nodes sync with one another. Additionally, this assignment demonstrates that the state does not desynchronize or go out of sync dynamically by adding and deleting nodes from the group.

2. Main Problems and Solutions

The problems faced during the implementation of the assignment are:

- **Reliable Multicast:** It involves maintaining message ordering consistency i.e., FIFO and total ordering across all system nodes, regardless of all events of failures. To counter this, the gms3 system tracks a series of messages issued by the leader using sequence numbers. Every message is identified by a sequence number, and in the event of a leader crash, the final message is sent again by the new leader. Every multicast message is broadcast to all the slaves by the leader, who identifies it by assigning a sequence number 'N'. When a message is received by the slaves, they verify the sequence number to avoid processing the same message twice. It is disregarded, if the received message is out of sequence i.e., if I<N.
- **Leader Election:** It is like managing a leader crash while making sure that the group can carry on under a new leader. When the present leader fails, it is imperative to elect a new one. Slaves elect a new leader when the current one fails and use `erlang:monitor/2` function to keep an eye on them. In the event of a leader's failure, the first node in the slave list is chosen to elect a new leader and make sure

that the group keeps running as soon as the new leader assumes control and keeps multicasting messages.

- **Fault Tolerance and Node Failures:** The process is to keep all nodes in a constant condition, when they join, exit, or crash. The leader modifies the group view upon a node's joining and multicasts the updated view to every node. When a node stops or fails, the system keeps working normally as the failing node is removed from the group.

3. Evaluation

Several tests were conducted to ensure the implementation to meet the requirements.

- Several workers have joined the group as well as removed dynamically with the help of `test:add/4` and `test:more/3`. The system updated the group and slave list as expected and appropriately handled the addition and removal of nodes.
- Leader failure was done using `exit/2` function, which will manually eliminate the leader. The system was able to choose a new leader from the remaining slaves. The records were examined, and they confirmed that the organization was still operating under the new leader.
- The worker instructions were used to start changing the system's state, such as transmitting multicast messages. The system remained synchronized, and all nodes implemented the state updates in the right sequence regardless of leader crashes and node deletions.

New leaders were chosen after every setback, and state synchronization was upheld, thus the system was stable throughout the test. Despite many leader elections and extended operations, no node lost synchronization. As anticipated, the freeze and go commands enabled nodes to halt and restart without losing state.

4. Conclusions

This report presents the efficient implementation of Group Membership Service that guarantees reliable multicast, leader election, and fault tolerance. When a node failed or a leader crashed, the system processed it gently and elected a new leader. Additionally, despite repeated additions and deletions of nodes, the state was in sync across all nodes. It can be of great use in real-life systems where it can continue to function consistently and be available even in the event of network breakouts or node failures.