Groupy: A Group Membership Service

Nagasudeep Vemula

October 2, 2019

1 Introduction

In this assignment we implement a group membership service that is used to demonstrate reliable multicasting. The prime purpose is to have multiple application layer processes to be in sync whilst passing messages i.e they should all perform the same sequence of state changes. The problem to solve is ensuring the synchronization of nodes despite the addition of new and removal of existing nodes from the network. GUI is given to us to show the states in terms of colours being displayed at the same time.

2 Main problems and solutions

The problem to solve is the synchronization between the members for the state changes and this is solved in the form of 3 implentations which show how the system reacts on addition of the synchronization algorithm and then reliable multicast for the group membership system(gms).

2.1 First Implementation: gms1

Following the instructions provided in the assignment report, the first implementation consists of the functionality to add nodes to the network with a leader election algorithm that communicates with the GUI to show different colours. This implementation has no fault tolerance check and it consists of a group which essentially has a leader and several slaves. In our next implementation we will add the crash functionality to get an understanding of the problem.

2.2 Second Implementation: gms2

In this implementation the system is modified to introduce crash functionality and re-election of the leader occurs upon this. The leader is selected as the node that appeared next to the leader in the list of peers. This detection is done by using the bif erlang:monitor/2 in the initialization of slave. Furthermore a new clause is added in the state of slave:

{’DOWN’, _Ref, process, Leader, _Reason} ->
election(Id, Master, Slaves, Group);

On testing this implementation we see that when the leader dies the slaves go out of synchronization and display different colours. This can be explained with an example using the bcast/3 method. For instance if the leader sends a message to the first slave and crashes, the second slave as a result does not receive any updates. Due to this slave 1 has received a message slave 2 hasnt and this causes them to go out of sync.

2.3 Third Implementation: gms3

In this system we implement a reliable multicast. Effectively this is seen during testing when the nodes dont go out of synchronization upon the leader crashing. Here, each message has a number and each message contains in its state the last message it has seen. To avoid duplicate messages we check if the number of messages needed is not inferior to the number of last message seen:

{msg, I, _} when I < N ->
slave(Id, Master, Leader, N, Last, Slaves, Group);

The crucial part is the election procedure where the elected leader will forward the last received message to all the peers in the group. This should ensure that the slaves stay synchronized.

3 Evaluation

For the evaluation I will be showing the difference in output between the 3 implementations. These are shown with embed videos

3.1 gms1

In gsm1 there is no crash functionality introduced and as a result the group is synchronized throughout:

https://drive.google.com/open?id=14rYfTgWQoISHMxbF9b821Fk0eeKyp2BA

3.2 gms2

In gsm2 the crash functionality is implemented and it can be seen that the slaves are not synchronised when the leaders crash:

https://drive.google.com/open?id=1n-bSiI149pn8BsV3YZHVOrpvqv2MCh9a

3.3 gms3

This implementation introduces the reliable multicasting solution and as a result it can be seen that the slaves are synchronised even when the leaders crash:

https://drive.google.com/open?id=1IkgcZCS6Z9xYSvac3QcsyPLsOnQ1vEV-

4 Conclusions

This assignment has given me insights on dealing with problems related to synchronization in a real world system albeit at a much smaller level. I got to learn more about reliable multicasting and how it can be implemented and also the difficulties of message passing in a distributed system .In terms of Erlang it was an introduction to using a GUI with this language.