Distributed Systems
Report 4: Groupy

Yerrapragada Akhil

October 2, 2019

1 Introduction

This seminar is intended to explain the process of creating a group membership service using Erlang and describe the functionality of it. The key technical aspects implemented in groupy are: worker, gui, gms1, gms2, gms3 and test.

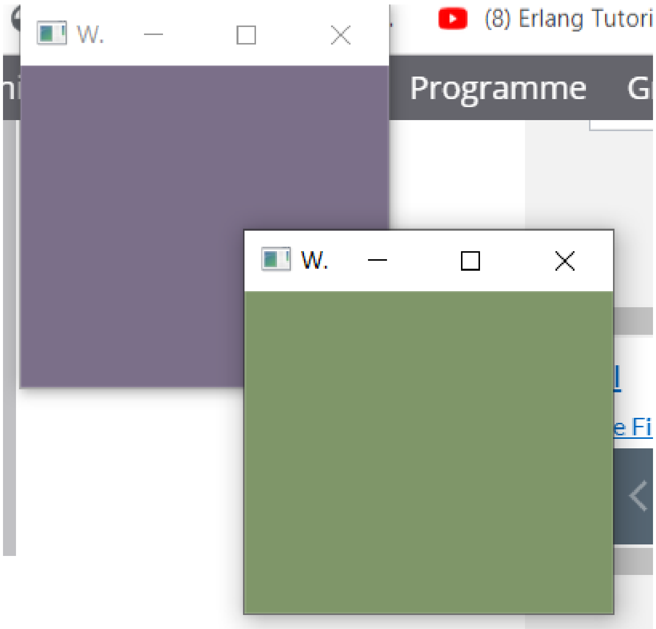
Here, all the nodes that wish to multicast a message will send the message to the leader and the leader will multicast the message to all the members of the group. If a new node wishes to enter the group, it will contact any of the node in group and request to join the group. The leader will determine when the node is to be included. The application layer process must also be prepared to decide if a new node should be allowed to enter the group and decide the initial state of this node. The application layer process will use the group membership to synchronize their states. Each application process will have its own group to communicate.

2 Main problems and solutions

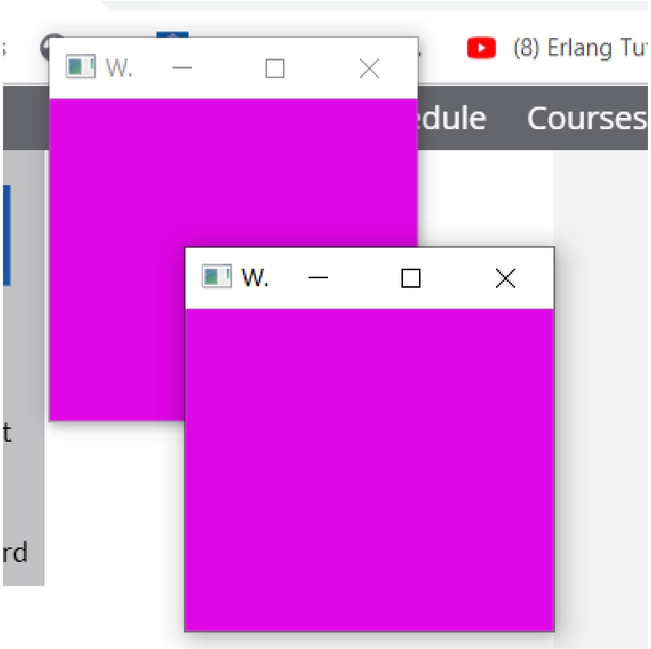
When slave receives a message from application layer, it transmits the state to the leader and the leader multicasts messages to all the members of group and their states will be synchronized. But when the leader gets crashed, the other slaves stop to display the colors. Therefore, we implement the re-election procedure to make sure that the slaves display different colors but why not same? This is because, say the leader sends a message to first salve and then dies, the slave 2 goes out of sync from salve 1 as the message was not sent to slave 2. To make sure all are in sync, we implement reliable multicast. Here, If the message has failed, the new leader sends out the last message it has seen to all the members to keep them in sync.

3 Evaluation

1) When the leader crashes we can observe the non-synchronization in state. The below test case was implemented using gms2.



2) We can observe the synchronization in state when we implement the crash handling technique



4 Conclusions

Overall, we provide a solution to implement a group membership service to send messages to nodes even when the leader crashes. GUI concept and certain API's such as "apply/3" were used in the implementation of the code which are exciting to know.