# Report 4: HW4 Groupy - a group membership service

Jiayi Feng

October 6, 2021

## 1 Introduction

This task will implement a group membership service that provides atomic multicast in different scenarios. There will be 3 erlang files, worker.erl, gui.erl and test.erl, provided, which can be directly used and compiled. Four new erlang files will be constructed as:
gms1.erl for no election scenarios
gms2.erl for election without synchronization scenarios
gms3.erl for election with synchronization scenarios
gms4.erl for receive acknowledgement scenarios

## 2 Main problems and solutions

### 2.1 no election

I personally treated this part as the initial setup part. The initial group, master and slave node are set within the gms1.erl. However, in this scenario, no election will be made if the master node dies. As a result, when the master dies, all the slaves cannot receive information and synchronization information from their master, and slaves' states, colors, will not be changed because the information updated is stopped.

### 2.2 election without synchronization

This part is the improved version of initial setup. In this scenario, master node election is introduced, which means the election of new master node after the old master node dies and slaves receive 'DOWN' message. Tagged slave with smaller number will be the new master nodes. However, asynchronization colors are still exist with the election procedure.

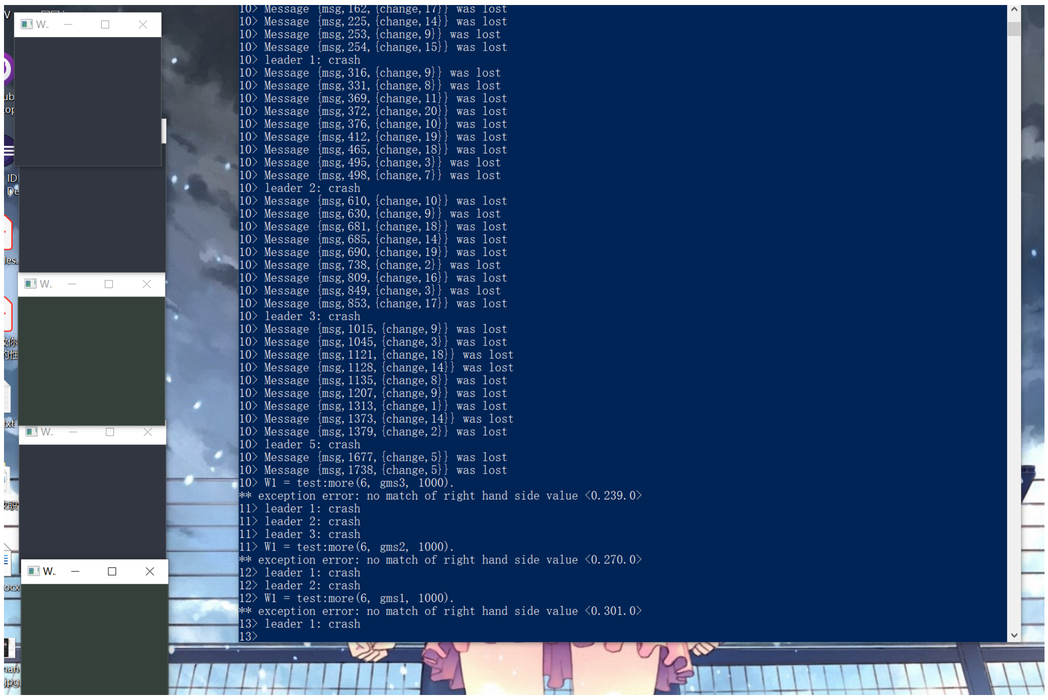
### 2.3 election with synchronization

This part is designed to solve scenario 2's asynchronization colors, even with the election procedure. Reason for this is the master's dying time. When the master dies, not all of the slaves receive its last message. As a result, those who received the message shift to another state, but those who did not have to wait for new master's message, which brings a state delay or gap between the received group and those did not, and also the inconsistency to the multicast communication.
As a result, both tag and storage are introduced for the message in this scenario. When the new master is elected, the master node have to resent the latest recorded message to all the slaves with tagged number. If the resend message's tag is smaller than the slaves, it means the slave has already received the message and the resend one will be removed. As a result, the color has reached synchronization.

### 2.4 bonus

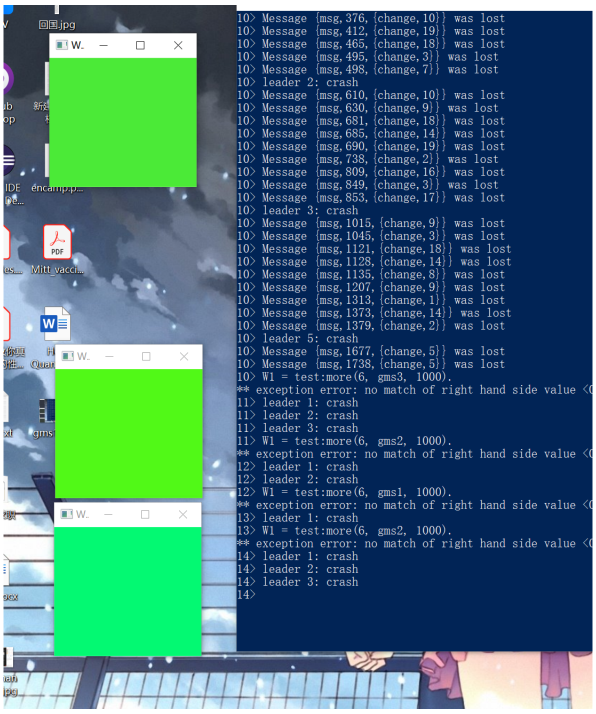
What actually guarantees the Erlang system sending messages as expected. The specifications only guarantee that messages are delivered in FIFO order, not that they actually do arrive. We have made the hypothesis that the system is perfect.
As a result, the acknowledgement process is introduced for the slave. Then the slave will send an acknowledgement to the leader once they receive the message from the master node. Before the acknowledgement confirmation received by the master node, it will wait for their response. If there is no response, then the master will send the recorded message again.

## 3 Evaluation



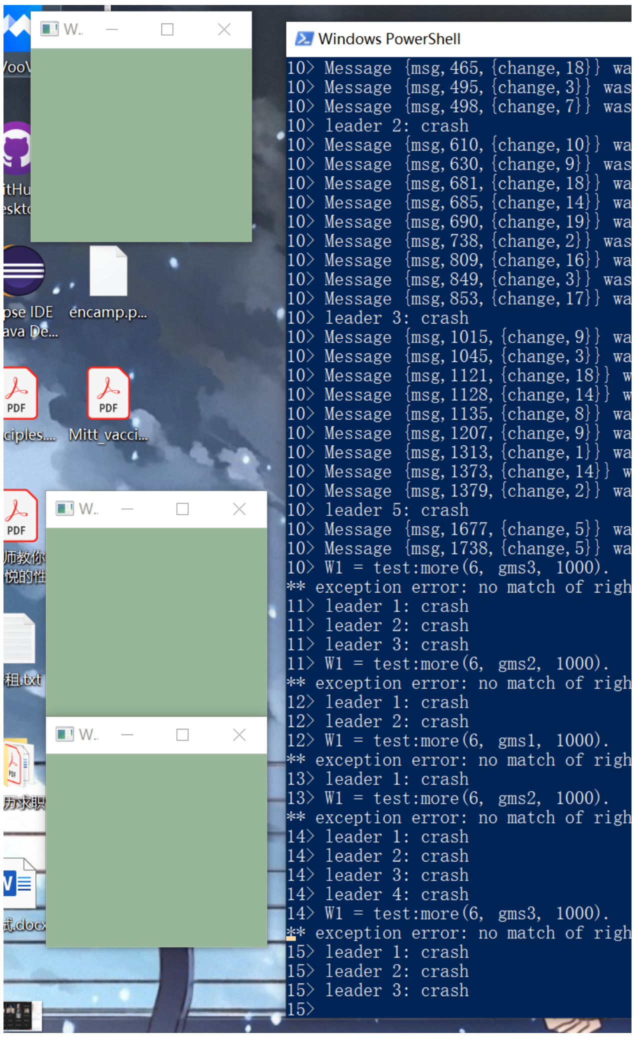
```
10> Message {msg,162,{change,17}} was lost
10> Message {msg,225,{change,14}} was lost
10> Message {msg,253,{change,9}} was lost
10> Message {msg,254,{change,15}} was lost
10> leader 1: crash
10> Message {msg,316,{change,9}} was lost
10> Message {msg,331,{change,8}} was lost
10> Message {msg,369,{change,11}} was lost
10> Message {msg,372,{change,20}} was lost
10> Message {msg,376,{change,10}} was lost
10> Message {msg,412,{change,19}} was lost
10> Message {msg,465,{change,18}} was lost
10> Message {msg,495,{change,3}} was lost
10> Message {msg,498,{change,7}} was lost
10> leader 2: crash
10> Message {msg,610,{change,10}} was lost
10> Message {msg,630,{change,9}} was lost
10> Message {msg,681,{change,18}} was lost
10> Message {msg,685,{change,14}} was lost
10> Message {msg,690,{change,19}} was lost
10> Message {msg,738,{change,2}} was lost
10> Message {msg,809,{change,16}} was lost
10> Message {msg,849,{change,3}} was lost
10> Message {msg,853,{change,17}} was lost
10> leader 3: crash
10> Message {msg,1015,{change,9}} was lost
10> Message {msg,1045,{change,3}} was lost
10> Message {msg,1121,{change,18}} was lost
10> Message {msg,1128,{change,14}} was lost
10> Message {msg,1135,{change,8}} was lost
10> Message {msg,1207,{change,9}} was lost
10> Message {msg,1313,{change,1}} was lost
10> Message {msg,1373,{change,14}} was lost
10> Message {msg,1379,{change,2}} was lost
10> leader 5: crash
10> Message {msg,1677,{change,5}} was lost
10> Message {msg,1738,{change,5}} was lost
10> W1 = test:more(6, gms3, 1000).
** exception error: no match of right hand side value <0.239.0>
11> leader 1: crash
11> leader 2: crash
11> leader 3: crash
11> W1 = test:more(6, gms2, 1000).
** exception error: no match of right hand side value <0.270.0>
12> leader 1: crash
12> leader 2: crash
12> W1 = test:more(6, gms1, 1000).
** exception error: no match of right hand side value <0.301.0>
13> leader 1: crash
13>
```

Figure 1: gms1



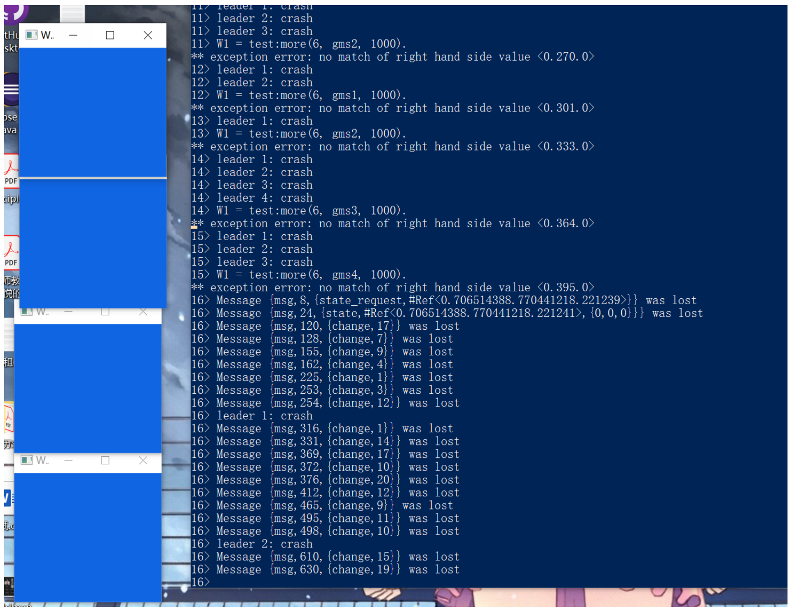
```
10> Message {msg,376,{change,10}} was lost
10> Message {msg,412,{change,19}} was lost
10> Message {msg,465,{change,18}} was lost
10> Message {msg,495,{change,3}} was lost
10> Message {msg,498,{change,7}} was lost
10> leader 2: crash
10> Message {msg,610,{change,10}} was lost
10> Message {msg,630,{change,9}} was lost
10> Message {msg,681,{change,18}} was lost
10> Message {msg,685,{change,14}} was lost
10> Message {msg,690,{change,19}} was lost
10> Message {msg,738,{change,2}} was lost
10> Message {msg,809,{change,16}} was lost
10> Message {msg,849,{change,3}} was lost
10> Message {msg,853,{change,17}} was lost
10> leader 3: crash
10> Message {msg,1015,{change,9}} was lost
10> Message {msg,1045,{change,3}} was lost
10> Message {msg,1121,{change,18}} was lost
10> Message {msg,1128,{change,14}} was lost
10> Message {msg,1135,{change,8}} was lost
10> Message {msg,1207,{change,9}} was lost
10> Message {msg,1313,{change,1}} was lost
10> Message {msg,1373,{change,14}} was lost
10> Message {msg,1379,{change,2}} was lost
10> leader 5: crash
10> Message {msg,1677,{change,5}} was lost
10> Message {msg,1738,{change,5}} was lost
10> W1 = test:more(6, gms3, 1000).
** exception error: no match of right hand side value <
11> leader 1: crash
11> leader 2: crash
11> leader 3: crash
11> W1 = test:more(6, gms2, 1000).
** exception error: no match of right hand side value <
12> leader 1: crash
12> leader 2: crash
12> W1 = test:more(6, gms1, 1000).
** exception error: no match of right hand side value <
13> leader 1: crash
13> W1 = test:more(6, gms2, 1000).
** exception error: no match of right hand side value <
14> leader 1: crash
14> leader 2: crash
14> leader 3: crash
14>
```

Figure 2: gms2



```
10> Message {msg, 465, {change, 18}} wa
10> Message {msg, 495, {change, 3}} was
10> Message {msg, 498, {change, 7}} was
10> leader 2: crash
10> Message {msg, 610, {change, 10}} wa
10> Message {msg, 630, {change, 9}} was
10> Message {msg, 681, {change, 18}} wa
10> Message {msg, 685, {change, 14}} wa
10> Message {msg, 690, {change, 19}} wa
10> Message {msg, 738, {change, 2}} was
10> Message {msg, 809, {change, 16}} wa
10> Message {msg, 849, {change, 3}} was
10> Message {msg, 853, {change, 17}} wa
10> leader 3: crash
10> Message {msg, 1015, {change, 9}} wa
10> Message {msg, 1045, {change, 3}} wa
10> Message {msg, 1121, {change, 18}} w
10> Message {msg, 1128, {change, 14}} w
10> Message {msg, 1135, {change, 8}} wa
10> Message {msg, 1207, {change, 9}} wa
10> Message {msg, 1313, {change, 1}} wa
10> Message {msg, 1373, {change, 14}} w
10> Message {msg, 1379, {change, 2}} wa
10> leader 5: crash
10> Message {msg, 1677, {change, 5}} wa
10> Message {msg, 1738, {change, 5}} wa
10> W1 = test:more(6, gms3, 1000).
** exception error: no match of righ
11> leader 1: crash
11> leader 2: crash
11> leader 3: crash
11> W1 = test:more(6, gms2, 1000).
** exception error: no match of righ
12> leader 1: crash
12> leader 2: crash
12> W1 = test:more(6, gms1, 1000).
** exception error: no match of righ
13> leader 1: crash
13> W1 = test:more(6, gms2, 1000).
** exception error: no match of righ
14> leader 1: crash
14> leader 2: crash
14> leader 3: crash
14> leader 4: crash
14> W1 = test:more(6, gms3, 1000).
** exception error: no match of righ
15> leader 1: crash
15> leader 2: crash
15> leader 3: crash
15>
```

Figure 3: gms3



```
11> leader 1: crash
11> leader 2: crash
11> leader 3: crash
11> W1 = test:more(6, gms2, 1000).
** exception error: no match of right hand side value <0.270.0>
12> leader 1: crash
12> leader 2: crash
12> W1 = test:more(6, gms1, 1000).
** exception error: no match of right hand side value <0.301.0>
13> leader 1: crash
13> W1 = test:more(6, gms2, 1000).
** exception error: no match of right hand side value <0.333.0>
14> leader 1: crash
14> leader 2: crash
14> leader 3: crash
14> leader 4: crash
14> W1 = test:more(6, gms3, 1000).
** exception error: no match of right hand side value <0.364.0>
15> leader 1: crash
15> leader 2: crash
15> leader 3: crash
15> W1 = test:more(6, gms4, 1000).
** exception error: no match of right hand side value <0.395.0>
16> Message {msg,8,{state_request,#Ref<0.706514388.770441218.221239>}} was lost
16> Message {msg,24,{state,#Ref<0.706514388.770441218.221241>,{0,0,0}}} was lost
16> Message {msg,120,{change,17}} was lost
16> Message {msg,128,{change,7}} was lost
16> Message {msg,155,{change,9}} was lost
16> Message {msg,162,{change,4}} was lost
16> Message {msg,225,{change,1}} was lost
16> Message {msg,253,{change,3}} was lost
16> Message {msg,254,{change,12}} was lost
16> leader 1: crash
16> Message {msg,316,{change,1}} was lost
16> Message {msg,331,{change,14}} was lost
16> Message {msg,369,{change,17}} was lost
16> Message {msg,372,{change,10}} was lost
16> Message {msg,376,{change,20}} was lost
16> Message {msg,412,{change,12}} was lost
16> Message {msg,465,{change,9}} was lost
16> Message {msg,495,{change,11}} was lost
16> Message {msg,498,{change,10}} was lost
16> leader 2: crash
16> Message {msg,610,{change,15}} was lost
16> Message {msg,630,{change,19}} was lost
16>
```

Figure 4: gms4

## 4 Conclusions

synchronization and acknowledgement have been the key points in gsm3 and gsm4. This homework has taught us the multicast communication step by step. The whole story starts with the initial settings among group, master and slave nodes, and their abstract states are illustrated in visualized color instead. After I found that the initial scenario was breakdown because of the master node's death, the second scenario has naturally turned to the master node selection. Asynchronization colors from scenario 2 are the reasons to have scenario 3, where the message is tagged and recorded.
Bonus part introduced a very interesting point, that is the imperfection of the system. Solution of this will be acknowledgement process.