Answer 3

* Do actors ever receive messages originating from a given actor out of order?

Yes the actors receive messages out of order

* What if the messages are forwarded through an intermediary?

The intermediary could have enforced the delivery order by organizing the messages by timestamp

* What if two actors multicast to the same group? Does each member receive the messages in the same order?

All the messages would reach the group members but messages are not guaranteed to be in order as it depends if the actor is available to process the request/message

* Do actors ever receive messages for groups "late", after having left the group?

Yes, the actors indeed receive messages ‘late’ after having left the group

* How does your choice of weights and parameters affect the amount of message traffic?

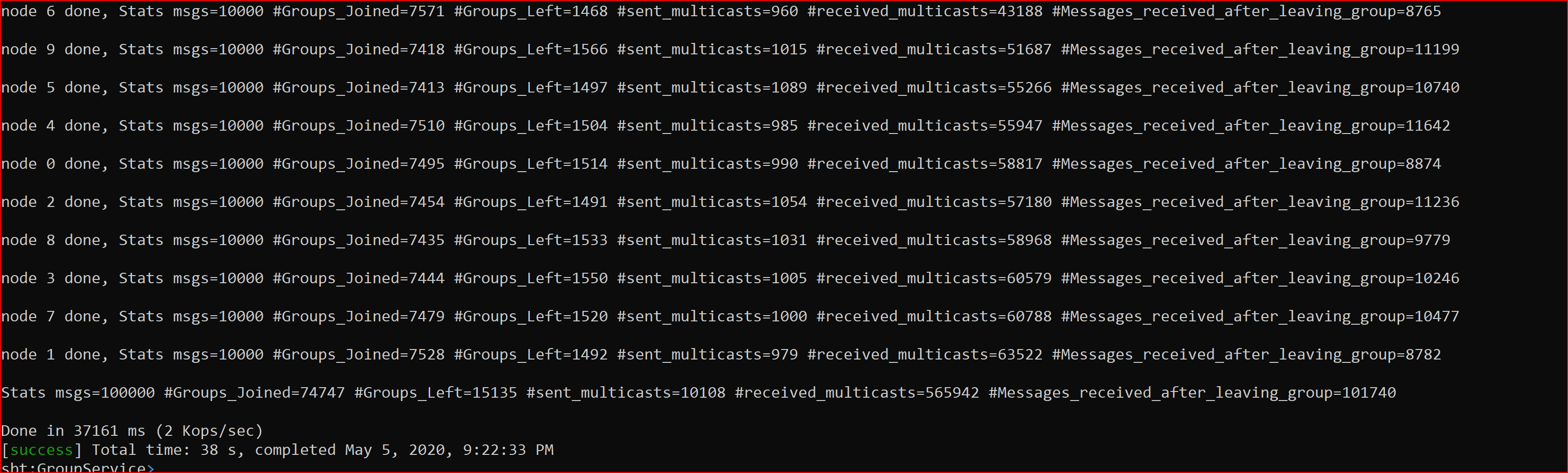
I used maxweight to be 100 and groupjoin weight is 60%, Leave group is 15% and multicast is 25% of the maxweight respectively, it leads to more traffic towards joining the group, followed by # of multicast messages and least is actors leaving the groups

* How can you be sure your code is working, i.e., the right actors are receiving the right messages?

The gameservers are able to join and leave the group  
Any multicast message is received by all members of the group.  
The gameservers are also able to report that the message from a group was received after it left the group

FEW NOTES

Set WRITETOFILE=true in GroupService.scala to generate the attached log files



Brief Design Detail

* KVAppService generates a bunch of GroupServer instances
* For each instance of the GroupServer, the LoadMaster sends the below message
  + Prime – doesn’t do much
  + Command –

Upon receiving the message the GroupServer instance randomly performs the below

* + - * Join a group - with a probability of 60%  
        randomly generates a number between 0 and numNodes where each number is considered to be a GroupID
      * Multicast a message - with a probability of 25%
      * Leave A Group - with a probability of 15%
  + Group Membership is maintained in KVStore where

key: BigInt,

value: GroupMembership

The LoadMaster prints the output once all nodes/GroupServer are done with processing

