

CS 425 MP2

Jiayi Luo (jiayi15)
Zhoutong Jiang (ztjiang30)

1. Summary

In this Machine Problem, we build a distributed group membership system by implementing a SWIM algorithm. Our system has the following functionality:

- 1.1 Display member information (time, address, status)
- 1.2 Detect the failure of other members within a short time via UDP
- 1.3 Write to the log files, which can be used for future debug

2. System Design

The system is designed as follows,

2.1 Member

We consider a member (which has 0 as its member ID) to be the introducer of the group, which is in charge of handling join information. The introducer's IP address will be known by all other members and when any other member wants to join, its information will be sent to the introducer.

2.2 Detection Algorithm

Failure detection: We have a hard criteria at each vm machine. Whenever one of the members doesn't have a newer version of its stats (based on the time information) with in time $F_failure$ (here we set 2s), we will mark it as failed in our own member list and propagate through ack.

Other detection (leave, join, rejoin and detection): This is based on the SWIM algorithm. We set the neighbor as a dynamic and ring-based list. From our calculation, we will need $T_read = 0.6s$ to satisfy the requirement. Here we set the value to be 0.5s in case some small network latency.

2.3 Network Topology

Given the fact that at most 3 VMs will fail simultaneously, each member will Ping a group of members that consists of at most 4 members (one introducer plus three members), which we referred as neighbor members. Each member will select its 3 accessors as its neighbors. For example, member i will select member $i + 1$, $i + 2$ and $i + 3$ (looping around if the number is greater than the number of member list). The neighbor members are dynamically adjusted based on the status of members in the group, for example, in the above case, if member $i + 3$ fails, member i will then try to ping member $i + 4$, etc. If there are not enough functioning members in the group (less than 4), then all the available members are included in the neighbor list. Such design will ensure that any member will be Pinged by 3 members, as a result, failures in any scenario will be detected.

2.4 Action

We define each member will have the following actions: (1) **Ping** a random member in its neighbor list; (2) **Respond** to a Ping message (Ack) by sending its member list; (3) **Update**, update its member list based on the member list from other members; (4) **Join**, send join request to introducer; (5) **Leave**, ask to leave the group.

2.5 Message

When a member ping the other member, the message being sent will include the following information, IP_address, memberID, action and timestamp. All the information all encoded into gob format.

3. Experiment and Performance Evaluation

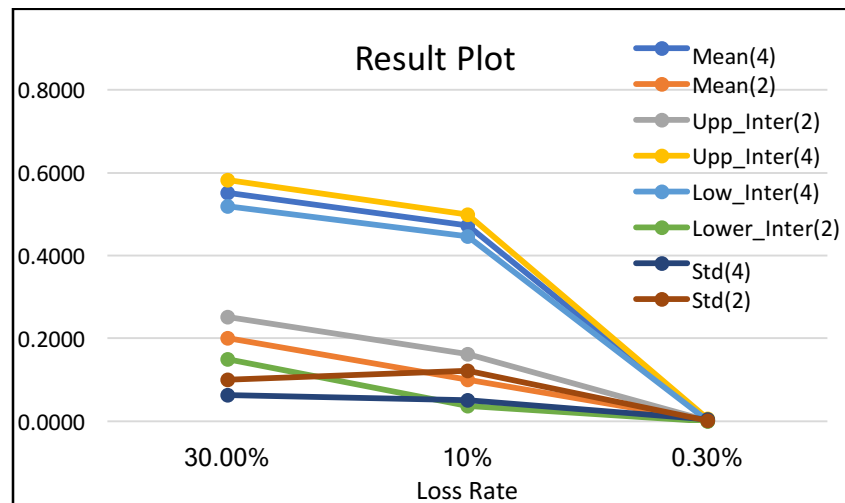
3.1 Bandwidth Usage

We monitor the bandwidth usage at introducer VM (using iftop), and we could see that with the increase of number of VMs, we could see that the increase of VM number, the bandwidth usage increases linearly. This is expected as in our system, the message being sent in the system increases linearly, and is $O(N)$, in which N is the number of members in our system.

VM Number	Normal Bandwidth Usage	Bandwidth Usage(Join)	Bandwidth Usage(Leave)	Bandwidth Usage(Fail)
2	4.6Kbps	9.7Kbps	11.2Kbps	6.5Kbps
4	10.4Kbps	20.2Kbps	23.1Kbps	10.3Kbps

3.2 False Positive Rate

We conduct the experiment as follows, we simulate the information loss by randomly drop the message sent between the members, and then wait for 6 seconds for the information to propogate and then count the number of functioning members that are misclassified and calculate the false positive rate. The result of mean, standard deviation and the 95% confidence interval is shown as follows, as we can see, the false positive rate increase with the number of information loss probability, which is intuitive. Also, we could see that if the member number is larger, the false positive rate is also larger, which results from false message being sent among the group member.



3.3 Utilization of MP1

We use MP1 result such that we will be able to grep the information for all VMs at one VM, which is much more convenient for debug and status check.