Project Report
CS425-MP2

Thread

We use 10 virtual machines and each node holds 4 threads: 1. Run a UDP listener to listen for messages about nodes' joining, leaving, crashing and heartbeats; 2. Receive user's command. There are 4 patterns of user command("leave", "join", "lsm" for member list, "ip"); 3. Send heatbeats to its three neighbors(one predecessor and two successors) every second; 4. Use the logic of MP1 to grep MP2 log files to debug with the grep pattern of "", "", ""

Logic

Among these 10 vms, we choose vm01 as the introducer that is responsible for helping other nodes to join the group. To be more specific, once the introducer receives a joining request, it would add this node to its member list and then notify all the members in this list, also, the introducer would send heartbeats to its 3 neighbors to help them and these neighbors would do the dissemination subsequently. For the member list, there are three components: ip, heartbeat count and local timestamp. Once a neighbor receives these messages, it would update its member list according to the heartbeat count in local member list and current member list. If the current one is bigger, member list would be updated. If a node has not been updated for more than 5 seconds, we would mark it as fail. When a node is failed, no heartbeats would be sent to others. Also, when a node voluntarily leaves the group, it would send the leaving message to others through the member list and no heartbeats would be sent after it leaves. In this way, its neighbors would notice that this node is dead, no matter it fails or it voluntarily leaves. So the neighbors would remove this node from the member list and push it into the recently removed list for cooling down. After 5 seconds of cooling, the removed node is allowed to rejoin the group.

Design

- **Marsheled message format**

Ip: Address(String); heartbeat count: HeartbeatCount(int);
timestamp: UpdateTime(int64)

- **Machine scale**

We use ring heartbeating so that no matter how large is the machine scale, each node is still connected with its 3 neighbors.

- **Failure**

We choose 3 neighbors in the virtual ring for each node to guarantee that at most three machines can fail simultaneously.

- **MP1 usage**

With the logic of MP1, we do not need to log into each machine and check whether the log result match our expectation complicatedly, we could just see the result in one machine through the grep command. It is very convenient and time-saving.

Measurement

- **Bandwidth usage**
  Background bandwidth usage: message size / 1s * 3 for one node, * 6 for all nodes
  Bandwidth usage when one node joins: 1) for introducer: message size/ 1m
  2) for other nodes: message size / 1s * 3
  Bandwidth usage when one node leaves: (message size / 1s * 3) / 5s
  Bandwidth usage when one node fails: (message size / 1s * 3) / 5s

- **False positive rate**
  1. When packet loss increases, false positive rate increases accordingly. It matches our expectation because when pack loss is higher, there would be more lost message of one node during a specific time period.
  2. The false positive rate of N=4 is higher than N=2, since more nodes in the group mean more message delivery, which would lead to a high possibility of message loss.