Distributed System MP2: Distributed Group Membership

Group 41 Cheng-Hsuan Huang(chhuang5) Yen-Chen Yeh(yy63)

Design

We implement C++ based distributed group membership with gossip style heart beating. We utilized UDP communication between members and create a Membershiplist class to store the id, heartbeat, incarnation number and the suspicion status of each member. Each member sends its list to random b members. The id of each member contains time, hostname and port. That way it can easily distinguish a member if it rejoins.

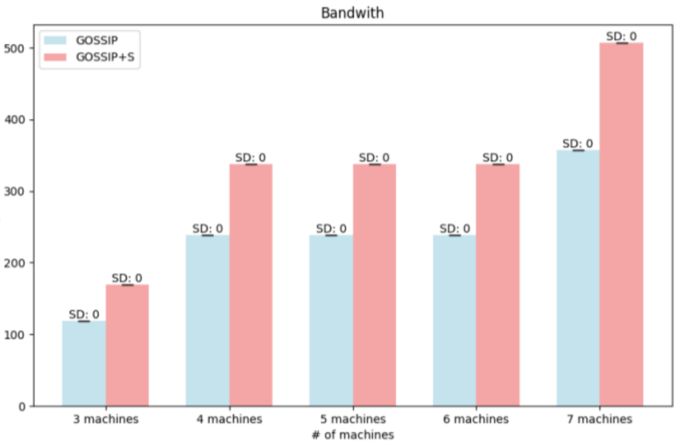
Join/Leave:

Introducer will always be vm1. Each member's client program will send message to vm1 to join the group and update its own membership list. To leave the group, member will stop sending and receiving, and then delete their membership list.

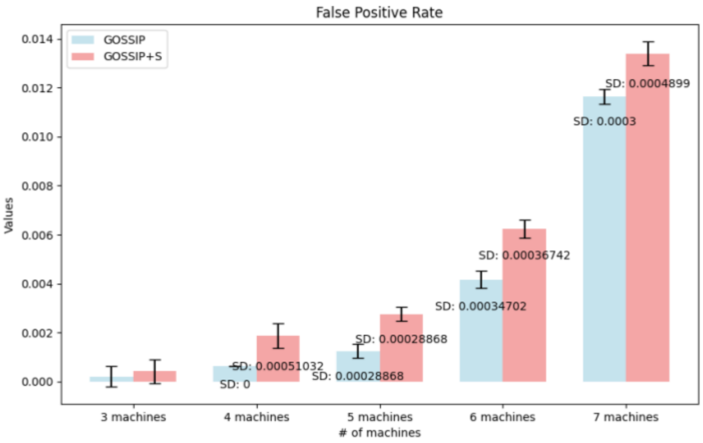
Failure detector:

There are two modes in this design: gossip & gossip+S. In gossip mode, each member checks other members' heartbeats, if one doesn't increase for more than T_{fail} time, then it is seen as fail. In gossip+S mode, each member will suspect another if its heartbeat doesn't change for more than $T_{suspect}$ time, once one is suspected, it should add its incarnation number, and send its list to others to prove that it is alive. If receive in T_{fail} time, then it is rejuvenated, or else, it is seen as fail.

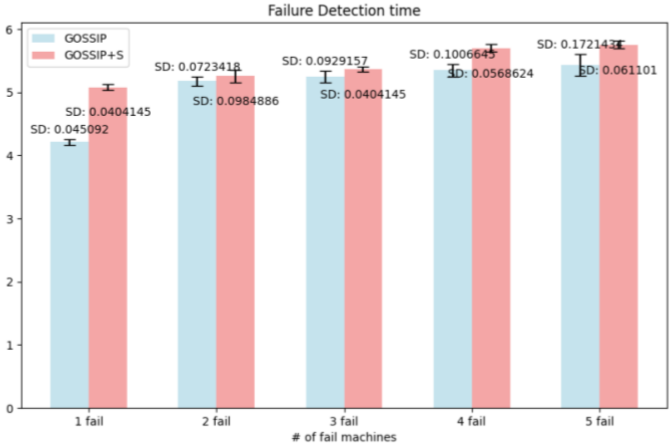
Analysis



Plot (a) Bandwidth for 3~6 machines

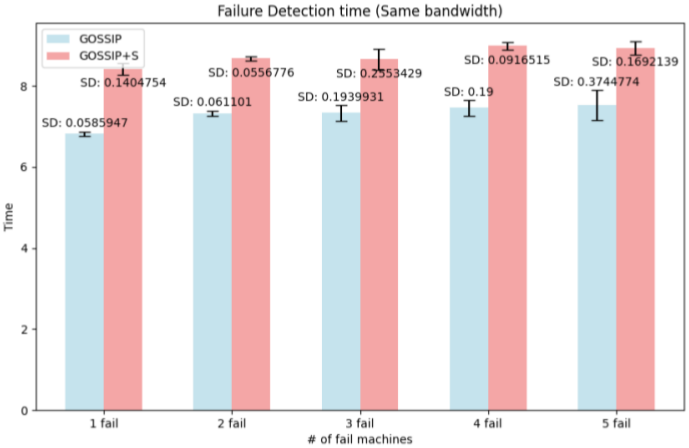


Plot (b) FPR for 3~6 machines at drop rate=16%

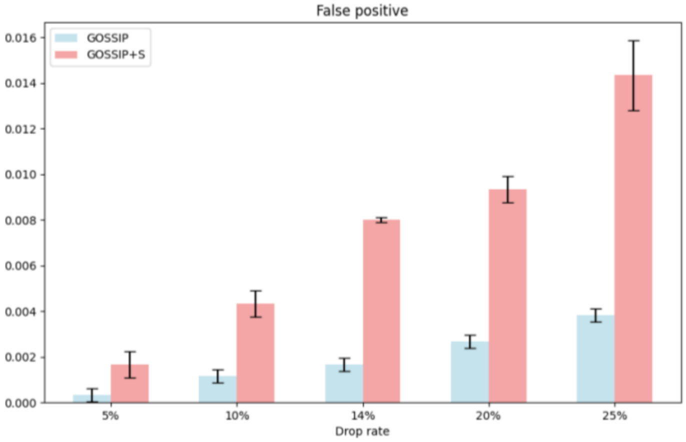


Plot (c) Detection time for simultaneous failures

1. Fixed detection time (~5 seconds)
    a. To keep the detection time fixed at around 5 seconds, we must adjust the disseminate node per round (b), which would directly interfere with the bandwidth.
    We assumed that to keep the detection time fixed, for a larger group, we must increase b, and that is also exhibited in Plot(a).
    b. We assumed that the FPR would increase with the size of the group because the increase of the bulk of gossip messages would also be affected by the fixed message drop rate. Also, we speculate that because GOSSIP+S mode has generally more bandwidth, the FPR would also be higher of that of the regular GOSSIP mode. Plot(b)
    c. The result is close, we think that GOSSIP+S takes more time averagely because after Tfail it still must wait for the suspicion time to time out. Plot(c)



Plot (d) Detection time for simultaneous failures



Plot (e) FPR for different message drop rate

2. Fixed base bandwidth
    a. For a fixed bandwidth, the detection time of GOSSIP+S mode is significantly longer than the normal mode. GOSSIP+S has a bigger gossip amount, limiting the bandwidth means we must decrease its overall dissemination nodes per round. Plot(d)
    b. Similar to 1.(b), GOSSIP+S also has a higher FPR for the same drop rate. However, we think that it is significantly higher because of the bandwidth tradeoff. Plot(e)
    c. The average gossip amount for our GOSSIP mode is around 118 bytes per member, and it is 169 bytes per member for the GOSSIP+S mode. There for we kept our dissemination per round to a ratio of 3:2 to match the bandwidth of the two modes. That makes the bandwidth ratio to 354:338, which is within 5% of each other.