CSCI-1380: Distributed Computer Systems
Homework #3
Assigned: 04/05/2018
Due: 04/19/2018

1 Bugs in Gossip

1. Your company is running gossip with 21 Replica Managers (RM). Each client sends requests to one (and only one) RM – this RM is chosen at random. Each RM gossips with three other randomly selected RMs. The gossip happens every 5 seconds. You've been alerted that, in causal consistency mode, the RMs respond very slowly to client requests. You've done some debugging and the problem is not due to network latency between the client and the RM: the problem is in the gossip protocol running on an RM. The RM appears to be waiting for something before responding. Given the design of the gossip protocol, what could the problem be? How could you solve this problem? What is the downside of this solution?

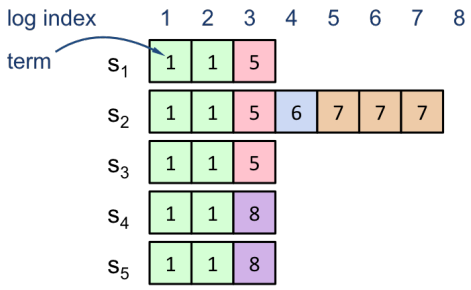
2. You've fixed the above performance problem. Now, an engineer comes to you and claims that when multiple clients perform the same request at the same time to different RMs, e.g., "Get (X)", the clients sometimes receive different values and sometimes receive the same value. The engineer claims that for queries/updates to X, these clients should always receive the same responses. What is the problem? How will you fix this? Explain your answers.

2 Raft

1. Your raft code is **buggy**. You notice that after a leader fails, it often takes a large number of *terms* before a new leader is elected. For example, if a leader fails in term five, the next new leader may not be elected until term two hundred. While debugging, you observe the following:

   - A majority of nodes are participating in the voting process and at least one candidate exists in each term but still no leader is elected.
   - There is no network failure: If a node votes for a candidate, the candidate receives the vote. If a candidate sends a request for votes, all nodes receives it.
   - Nodes are voting correctly — according to the protocol.

   Why is no leader elected? What is one potential probem that could prevent a leader from being elected for many consecutive terms? How would you fix this?



2. Using the figure above. Answer the following questions:

   - If S_2 becomes the leader and successfully replicates index 4 to **all followers**, should it commit and return to the client?

   - For S_2 to become the leader, which followers could vote for S_2?

   - What is the first index that S_2 can successfully commit **and** return to the client?

   - Assume S_2 remains the leader for the rest of this question. What happens to the requests/entries stored in term 8 (index 3) on S_4 and S_5? How does Raft ensure that these requests/entries eventually get committed?

3 Handing In

Once finished, you should hand in a PDF with your answers on Gradescope. Gradescope will allow you to select which pages contain your answers for each part of each question.

Please do not put your name on any page of your handin! This will allow us to do fully anonymized grading through Gradescope.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS138 document by filling out the anonymous feedback form:
http://cs.brown.edu/courses/cs138/s18/feedback.html.