ECE428 Homework 5

Due: 11:59 p.m. on Monday 9th May, 2022

This assignment has 2 questions with 40 points in total. The solutions must be typed, and submitted via Blackboard. However, the diagrams can be hand-drawn. You must acknowledge any sources used to arrive at your solutions, other than the course materials and textbook. All homework assignments are expected to be an individual work, so no collaborations are allowed.

Question 1: Distributed Transactions *[22 points]*

Consider a distributed transaction with a coordinator and two participants. The coordinator and the participants are each implemented as a 3-node Raft cluster. During the transaction, the client communicates with the leader nodes of each of the clusters. At the end of the transaction, the coordinator performs a two-phase commit by having the leader of the coordinator cluster communicate with the leaders of the participant clusters. The distributed system considered is shown in Figure 1.

Assume that (1) the one-way delay between nodes in the same cluster are 2 ms and the one-way delay between nodes in different clusters is 5 ms, (2) the replica leader has been elected and is known to the other participants, and(3) there are no communication or node failures, unless otherwise specified.

(a) (2 points) How long does it take to achieve a consensus between the cluster members? Assume that the leader immediately sends AppendEntries rather than waiting for the next heartbeat interval, and that the followers reply immediately.

(b) (2 points) At what point(s) in the two-phase commit protocol do the nodes need to run the Raft consensus protocol (i.e., a commit event)?

(c) (2 points) Calculate the delay from the start to the finish of the two-phase commit protocol using the specified delay parameters.

(d) (4 points) At what point in the two-phase commit protocol can the coordinator reply to the client that the transaction is successfully committed? Calculate the delay from the start of the protocol (at the coordinator) until this point.

(e) (4 points) At what point in the two-phase commit protocol can each participant release transaction locks? Calculate the delay from the start of the protocol until this point.

(f) (2 points) How long would a consensus take if Paxos were used instead of Raft? Assume that all nodes act as acceptors; additionally, the leader node also acts as the distinguished proposer and learner.

(g) (6 points) Now consider using Raft again but with a different delay configuration. The leaders of the clusters are placed nearby each other, so that the communication between any two leaders has only a 2 ms delay. Each cluster has a local follower, in a different rack, with a delay of 5 ms delay. The other follower is in a backup data center, with a 20 ms delay. See Figure 2 Recompute the delays in parts (c), (d), and (e) under this configuration.
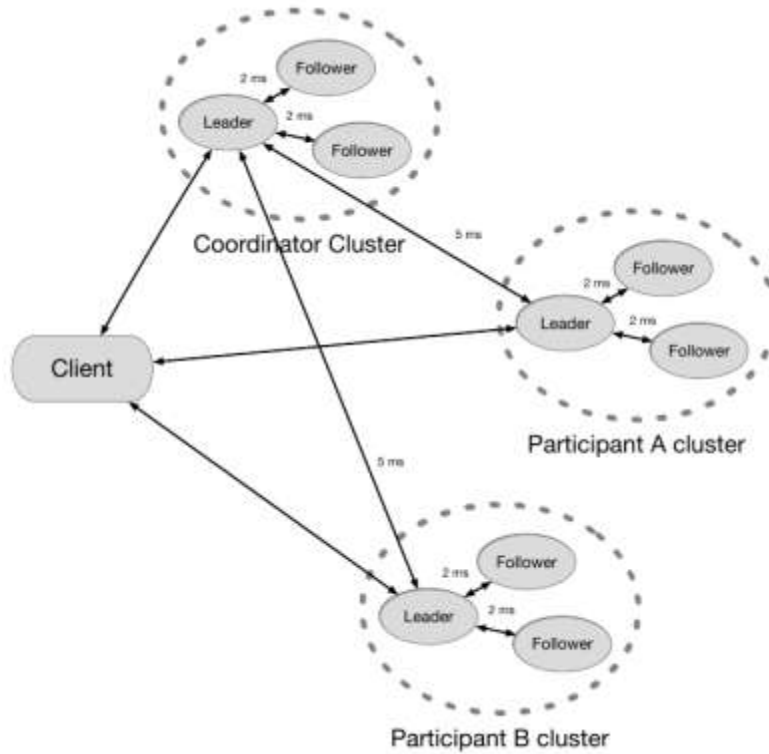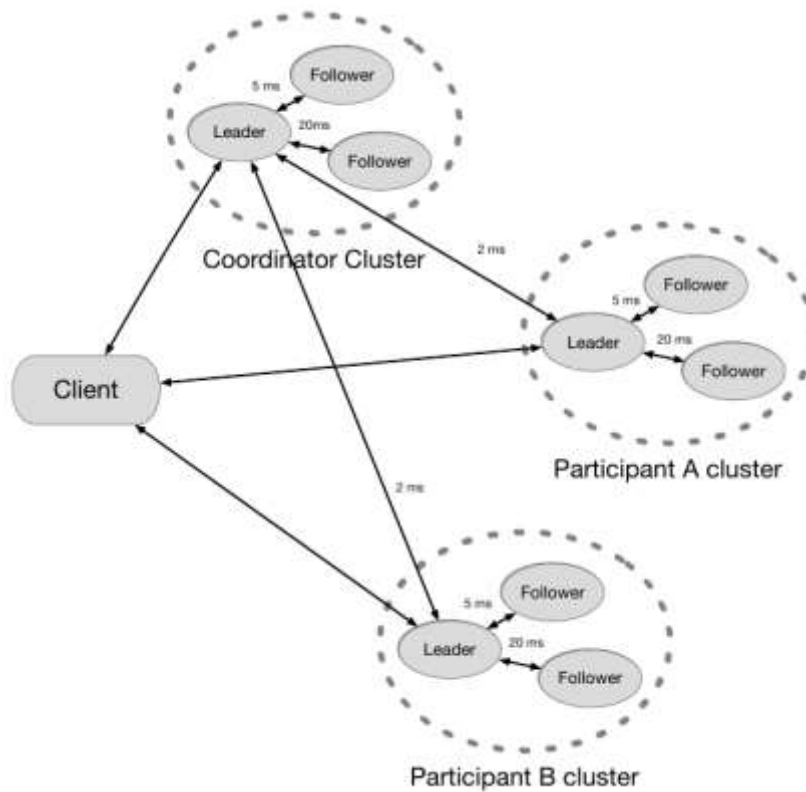
Figure 1: Node configuration for Question 1.

Figure 2: Delay configuration for Question 1(g).

Question 2: MapReduce *[18 points]*

For each of the below parts, write either a MapReduce job or a chain of MapReduce jobs to solve the question. Write down brief pseudocode (Python-like, preferably) for each of the map and reduce tasks. Also write a brief description that explains the type and meaning of each of the intermediate key-value sets produced by a map task or a non-final reduce task in a chain. Here is an example specification of a chain finding common words:

```
# map1 input:
#   key: input line number
#   value: input line text
def map1(k, v):
    for word in v.split():
        emit(word, 1)

#   key: word name
#   value: always 1

def reduce1(k, vv):
    emit(k, sum(vv))

#   key: word
#.  value: count of word

def map2(k, v):
    emit(v, k)

#   key: count
#   value: word with that frequency

def reduced2(k, vv):
    emit(k, concatenate(vv))
```

(a) (6 points) For contact tracing purposes, find people who have spent 15 minutes or longer in the same location. The input will be of the form:
• Key: location identifier
• Value: a triple (name of person, entry time, exit time)

The output should be of the form:
• Key: name of person
• Value: list of names of contacts

Note that any contact should only appear in the list once.

(b) (6 points) Multiply two matrices $M^{(1)}$ and $M^{(2)}$. The input will be of the form:
• Key: triple *(i, j, n)*, where *n = 1* or *2*, representing the element at row *i* and column *j* of matrix *n*
• Value: a floating point number *x*; $x = M^{(n)}_{i,j}$

The output should be of the form:
• Key: tuple *(i, j)*
• Value: a floating point number *y* representing the value at row *i* and column *j* of the result matrix, $y = P_{i,j.}$

Recall that $P_{i,j} = \sum_k M^{(1)}_{i,k} \times M^{(2)}_{k,j}$

You may assume that the dimensions of the matrices are known, fixed, and compatible.

(c) (6 points) For each node in an undirected graph, find its 4-hop neighborhood; i.e., all nodes that are distance of at most 4 from the given node. Each node should be included in its own 4-hop neighborhood. The input will be of the form:

• Key: node name u

• Value: v such that (u, v) is an edge in the graph

You should assume that you can compare node names using the operator $<$. Since this is an undirected graph, each edge will only be represented once, and you can assume that $u < v$.

The output should be of the form:
• Key: node name u
• Value: the list of 4-hop neighbor nodes, $v_1, \ldots, v_m$