CSCI-1380: Distributed Computer Systems
Homework #4
Assigned: 04/29/2020
Due: 05/05/2020 11:59 PM ET

1 Consistency Models

Your team has informed you that the cluster is designed for the strongest consistency model available. The cluster is set to a size of 7.

1. If your team is expecting non-byzantine failures, what is the max number of simultaneous node failures that this cluster can tolerate? What happens to the system if your cluster experiences more failures? Explain.

Given the timeline below in Figure 1, where a mobile client and laptop client are interacting with a distributed hash table system where each key is replicated on three servers. Answer the following questions given the assumed consistency models.

- Assuming a Linearizable model, what value does the laptop client get?
- Assuming Eventual consistency model, what values does the mobile client get?
- Assuming Eventual consistency model, what values does the laptop client get?

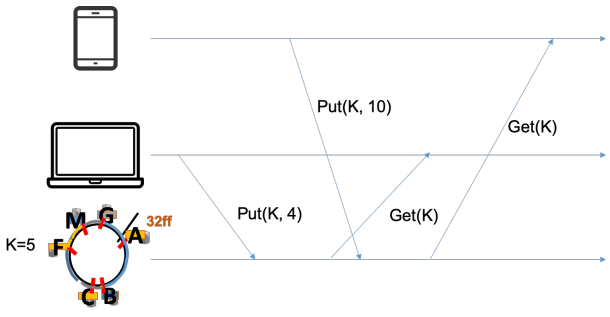

Figure 1: Consistency Timeline.

2 Practical Distributed File Systems

Your company is currently running Brown File System, BFS, (a distributed File system with opportunistic locking) and your workloads are 95% writes and 5% reads. You have client-side caching enabled and thus the opportunistic locks are meant to improve performance while maintaining strong consistency. You observe that most of the writes are to the same files: at any point in time multiple clients are simultaneously writing to the same files — usually at any time, there are usually 5 or more clients writing to a file. The reads, on the other hand, are localized: usually at most one client is reading a file. Your files are all large over 10GBs in size.

Due to unique conversations, you have setup BFS such that only one client can hold any locks (read or write) for a file. To read or write, clients must acquire an opportunistic lock.

Over the weekend, you decide to turn off opportunistic locking (this is the only change to your file system). What is the anticipated impact of this design choice? Explain.

3 Eventual Consistency: Dynamo and Cassandra

Your team is deploying Dynamo with a replication factor of 3. The ID of the Dynamo nodes are in a space of 0-255. The IDs of the Dynamo nodes are: 3, 60, 29, 192, 299, 230, 185, 320.

1. Please identify the set of servers where the following objects are stored:

Keys	Servers
45	
90	
4	
170	

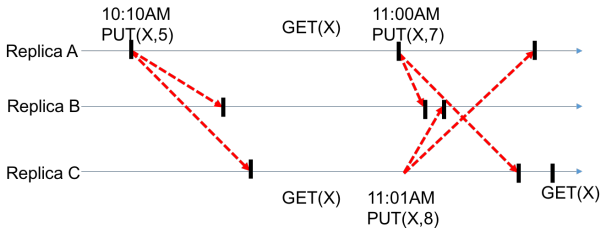

Figure 2: Timeline.

2. In Figure 2, we present the timeline of the interactions between two clients and two replicas (A and C). The figure also shows replication between the replicas (dashed lines).

   - If this system is using Cassandra, what is the value returned by Replica C for the last Get? Why?
   - If this system is using Dynamo, what is the value returned by Replica C for the last Get? Why?

4 ExtraCredit: Consensus Services: Zookeeper

Your team is using Zookeeper as the consensus service. Figure **??** shows the current code for checking and acquiring locks. How would you change the code to use Zookeeper while ensuring that you avoid busy-wait and herd problems. You want a version that uses ZooKeeper and minimizes CPU overheads. Once your code has the lock, it must call the function "doAmazingWork()". Once "doAmazingWork()" returns, you should relinquish the lock. You goal is to replace "getLock()" and "releaseLock()" with Zookeeper specific concepts.

```
doneWork := false
for !doneWork {
    if getLock() {
        doAmazingWork()
        releaseLock()
        doneWork = true
    }
}
```

5 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 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/s20/feedback.html