Summary

This project integrates Paxos in our key-value storage system. Paxos is a great solution to solve inconsistency problem among replicas of servers. Paxos ensures that all the replicas have the same data at last even if some of the servers go down or messages arrive at servers at different time. In this implementation, we have five servers and one client. When the client communicates with a server, that server acts as a proposer and the others act as acceptors. Then the proposer acts as coordinator and the accepters act as learners.

Technical Impression

In order to integrates Paxos in our key value storage system, we need to add a proposal number to our request body. I assume that only one proposal with the highest proposal number would be executed at a time, and other attempt would be rejected. Another assumption is that the server will not actually go down or lose connection, because the port numbers are hard coded and the RMI connections are established as soon as servers starting. But I implement a method to simulate server failures. To do that, random failures will be triggered at prepare and accept stages, by randomly rejecting requests. In other word, the proposer would act as if some of the acceptors are dead and decide whether consensus has been reached based on other acceptors. The “dead” acceptors would get data synchronized at last by PUT operations.