

Executive Summary

The purpose of this project was to extend Project #3 which was to establish an efficient RPC connection between a client and multiple servers, using two-phase commit protocol with the remote servers being available/deployed in the virtual cluster, with each server on a separate virtual node. In this project we extended the Project #3 by adding fault tolerance and achieving consensus of updates amongst replicated state machine KV-store servers by implementing PAXOS algorithm.

In this project we nominate one of the servers as LEADER. When the clients send the request for transaction to the leader the leader will send the value and a PROPOSAL number to all the peers to check the availability and priority of tasks that the peers are working upon. The peers (ACCEPTORS) will acknowledge and send back the current proposal number they are working upon. The leader will create the BALLOT and validate the majority decision and enforce necessary action.

Most of the challenges while working on the project were focused on PAXOS communications setup. The communication setup was necessary for the maintenance of consistency between all five KV stores situated on all server nodes. To achieve this consistency, the leader made calls to each server port's remote methods (put, get and delete); if the connection was successful, the method would proceed to make calls to the next server port. A C-sharp library named *remote methods* has been implemented for the actual manipulation of the KV store on each server node. Another challenge we faced while working on the project was debugging this PAXOS setup. As the code was being executed on multiple servers, debugging at each server was a hassle. So to debug the program, several test cases needed to be setup at several locations to identify the errors.