CS578CC Programming Assignment 3.2

Instructor: Prof. V. Arun
CICS, UMass Amherst

Due: Dec 4, 2023

Name: Riya Deshpande   Name: Riya Danve
Student ID: 34011024   Student ID: 34038293

1 Zookeeper: Overview

To implement a fault-tolerant system in addition to the consistent system that we built earlier, Zookeeper can be used as a coordinator to make the system fault-tolerant. Unlike PAXOS, zookeeper does not enable replicated state machines on the servers. So, zookeeper would help us to create z-nodes and coordinate the requests among all replica servers.

2 Architecture

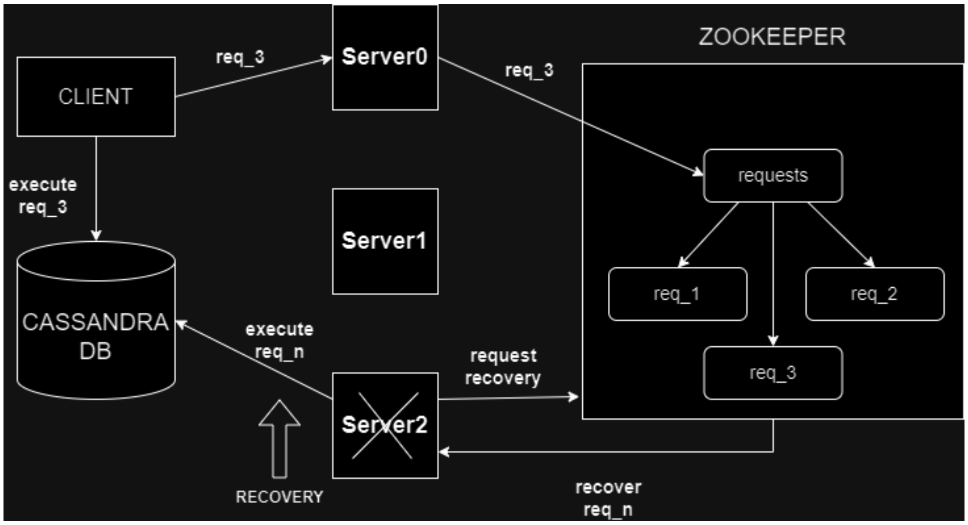

Figure 1: Zookeeper Implementation Architecture Diagram

The above diagram represents a high level design of our proposed system implementing zookeeper. Here server 2 crashes and recovers.

3 Request Handling

We run a single zookeeper instance. When a client sends a request to a server that request will get stored as a child to a znode called 'requests'. This request after that will then get executed on the Cassandra database. Since our approach is to keep only a single zookeeper instance a global write order gets followed automatically. Due to this we do not need a separate methodology for replica coordination or leader election.

4 Recovery

After a server crashes and recovers, first its constructor will get called. This constructor will maintain the crash recovery logic.
As part of the crash recovery, the server will first reach values of all the znodes under the parent node 'request'. These children nodes, as described above are nothing but the requests executed so far. This recovered server will then parse each request to fetch its id and will check if the key is present in its database. If it is not present, the server will execute that request and if it is not present, it will continue reading other requests from zookeeper's child znodes. Only after all the requests get parsed will that server accept new client requests. Keeping the recovery logic as part of the constructor achieves this automatically.