Project 3

Osamah Alzacko & Anurag

# Server Implementation

The server is working in a multi-threaded architecture as follows:

* Server main socket listener
  + Keep listening to new connections and reply to any incoming command.
* Recovery thread
  + A separate thread that handles the connection to two servers before the host and two servers after, and handles saving the stream of keys from each server.
* Command thread
  + A separate thread that handles user commands directly on the server like read a value, write a value, print all key:value in the server, recover key store from other servers, and a kill switch to quit the application.

## Features

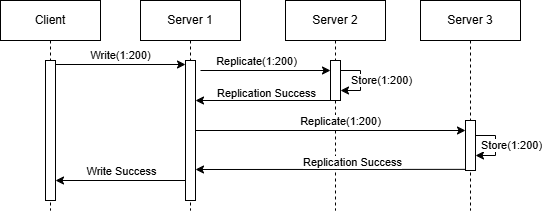
1. Recovery: The server supports recovery on startup, each host will try to connect to the 2 predecessor servers and 2 successor servers in the ring and will get all key:value store that is related to the host using the Hash function to determine if the key should be stored in the host.

A diagram of a server

AI-generated content may be incorrect.

Server recovery sequence.

1. Replication: Servers support replication of write commands and each write would be executed in the host server and on 2 successor servers for replication. When a write command is executed, the connection with the client is blocked until the write command is executed and replicated (if the replication was successful with 1 or 2 replications) then write will be executed on the server and an acknowledgement is sent to the client.



Replication process