Nicholas Soliman
Aaryan Shenoy

MP3 Design Document

To start this machine problem we first break down the project into three main components: clients, coordinator, and server clusters. The order that we are going to create each of these components is by the order of compilation. So the main steps that take place are listed below in chronological order:

1. Coordinator is created and server runs
2. Master server for each cluster is ran, appending its corresponding IP and Port in Coordinator database upon start up
3. Slave server for each cluster is ran, appending its corresponding IP and Port in Coordinator database upon start up
4. Client is ran and immediately contacts the coordinator to request an available IP and Port
5. Coordinator responds with IP and Port. Client creates stub and connects to that address
6. Interactions between client and master/slave server are essentially the same as MP2
7. Master and Slave mimic processes and if master fails to contact the coordinator after 20 seconds then the slave takes over
8. The Follower Synchronizer also contacts the coordinator to fill corresponding routing tables
9. Then it synchronizes the following information and the timeline information through a multithreaded implementation
10. In order for the synchronizer and server interactions to be correctly implemented we ensured that the synchronizer wrote to files that the server would only read from and vice versa

The following processes communicated via GRPC:

- Client —> Coordinator (Retrieves available IP and Port)
- Master/Slave —> Coordinator (Sends IP and Port to fill routing tables)
- Client —> Master/Slave (Calls functions like List, Follow, etc..)
- Master —> Slave (Mimics processes and Slave replaces Master if it fails)
- Follower Synchronizer —> Follower Synchronizer (To send follower information from server cluster to server cluster)
- Follower Synchronizer —> Coordinator (Fills routing tables)

Nicholas Soliman
Aaryan Shenoy

The commands used to run each aspect of this program for a successful run through are listed below:

```
./coordinator -p 8080
./tsd -h 0.0.0.0 -c 8080 -p 1260 -i 1 -t master
./tsd -h 0.0.0.0 -c 8080 -p 1261 -i 1 -t slave
./tsd -h 0.0.0.0 -c 8080 -p 1262 -i 2 -t master
./tsd -h 0.0.0.0 -c 8080 -p 1263 -i 2 -t slave
./tsd -h 0.0.0.0 -c 8080 -p 1264 -i 3 -t master
./tsd -h 0.0.0.0 -c 8080 -p 1265 -i 3 -t slave
./tsc -h 0.0.0.0 -c 8080 -p 1266 -i 1
./tsc -h 0.0.0.0 -c 8080 -p 1267 -i 2
./tsc -h 0.0.0.0 -c 8080 -p 1268 -i 3
./tsc -h 0.0.0.0 -c 8080 -p 1269 -i 4
./tsc -h 0.0.0.0 -c 8080 -p 1270 -i 5
./tsc -h 0.0.0.0 -c 8080 -p 1271 -i 6
./synchronizer -h 0.0.0.0 -c 8080 -p 1272 -i 1
./synchronizer -h 0.0.0.0 -c 8080 -p 1273 -i 2
./synchronizer -h 0.0.0.0 -c 8080 -p 1274 -i 3
```

Tips if error occurs:

-Ensure that the program has enough time for the periodic functions to update timelines and followers (can sometimes be upwards of a minute)

-If program fails to connect immediately after disconnecting, try again

Finally, I would like to mention that every portion of this assignment was completed fully and accurately. However, due to lack of time (working 3-9 hours a day since week 1), we were unable to fully implement the list function to display all_users from other server_clusters. We have written pseudocode to do so, but were unable to complete this portion.

GitHub link: https://github.com/nbsoliman/CSCE438/tree/main/MP3_Team