

CSCE 438: MP 2.1

Kevin Zhang

March 4, 2025

1 Overview

In MP 2.1, our main goal is to develop a coordinator process that coordinates between clients and a *group* of servers. To do so, we need a deterministic way for clients to be assigned to a specific server, and we need the servers to register themselves with the coordinator.

2 Client-Coordinator Interaction

When client processes start up, they immediately contact the coordinator process (whose host and port are known to the client) to get assigned to a cluster/server. The coordinator responds with the IP address and port of the client's assigned server.

The coordinator determine the client's assigned server with the following formula:

$$\begin{aligned} clusterID &= (clientID - 1) \% 3 + 1 \\ serverID &= \text{first server registered to } clusterID \end{aligned}$$

Where *clientID* is the same as the client's username. Here, we perform modulus by 3 because we assume a fixed number of clusters (3). Additionally, for this MP, we assume that each cluster only has up to one server, so we simply return the first server registered under each cluster.

3 Client-Server Interaction

For this MP, the client-server interaction remains largely the same. Once a client is assigned a server, they cannot change which server they are assigned to. Consequently, if the client's server goes down, they cannot access the SNS.

The only difference, which will not be explored in this MP, is that the method the server uses to update client posts is different.

4 Coordinator-Server Interaction

One of the primary jobs of the coordinator process is to register every server in its registration table. The coordinator will use this table to track the liveness status of each server and assign servers to clients.

For a server to register itself, it sends “heartbeat” messages periodically to the coordinator. Each message contains the server’s cluster ID and a **ServerInfo** gRPC object containing the server’s ID, hostname, port, and type (i.e. follower coordinator or server).

On the first heartbeat message the coordinator receives from a server, the coordinator adds the server to its list of registered servers. It now expects a heartbeat at least every 10 seconds for the server to remain “alive.” If 10 seconds elapse since a server sends a heartbeat, the coordinator declares the server “dead.”

5 Test Cases

5.1 Test 1: Sanity check

```
csce438@1656762fa27b:~/mp2_1$ ./coordinator -p 9090
Server Listening on 127.0.0.1:9090

csce438@1656762fa27b:~/mp2_1$ ./tsd -c 1 -s 1 -h localhost -k 9090 -p 10000
Server Listening on 0.0.0.0:10000

csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 1
E20250304 20:55:08.404052 91790 tsc.cc:314] client 1: did not receive 'login' reply from server
connection failed: -1
csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 1
E20250304 20:55:17.852819 91914 tsc.cc:314] client 1: did not receive 'login' reply from server
connection failed: -1
csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 1

===== TINY SNS CLIENT =====
Command Lists and Format:
FOLLOW <username>
UNFOLLOW <username>
LIST
TIMELINE
=====
Cmd> list
Command completed successfully
All users: 1,
Followers:
Cmd> timeline
Command completed successfully
Now you are in the timeline
p1
p2
p3
```

5.2 Test 2: Kill server before client

```
csce438@1656762fa27b:~/mp2_1$ ./coordinator -p 9090
Server Listening on 127.0.0.1:9090

csce438@1656762fa27b:~/mp2_1$ ./tsd -c 1 -s 1 -h localhost -k 9090 -p 10000
Server Listening on 0.0.0.0:10000

csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 1
E20250304 20:55:17.852819 91914 tsc.cc:314] client 1: did not receive 'login' reply from server
connection failed: -1
csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 1

===== TINY SNS CLIENT =====
Command Lists and Format:
FOLLOW <username>
UNFOLLOW <username>
LIST
TIMELINE
=====
Cmd> list
Command completed successfully
All users: 1,
Followers:
Cmd> timeline
Command completed successfully
Now you are in the timeline
p1
p2
p3
```

5.3 Test 3: Kill server after client

```
csce438@1656762fa27b:~/mp2_1$ ./coordinator -p 9090
Server Listening on 127.0.0.1:9090

csce438@1656762fa27b:~/mp2_1$ ./tsd -c 1 -s 1 -h localhost -k 9090 -p 10000
Server Listening on 0.0.0.0:10000

csce438@1656762fa27b:~/mp2_1$ ./tsc -c 2 -s 2 -h localhost -k 9090 -p 10001
Server Listening on 0.0.0.0:10001

csce438@1656762fa27b:~/mp2_1$ ./tsd -c 2 -s 2 -h localhost -k 9090 -p 10001
Server Listening on 0.0.0.0:10001

csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 1

===== TINY SNS CLIENT =====
Command Lists and Format:
FOLLOW <username>
UNFOLLOW <username>
LIST
TIMELINE
=====
Cmd> list
Command completed successfully
All users: 1,
Followers:
Cmd> timeline
Command completed successfully
Now you are in the timeline
p1
p2
p3

csce438@1656762fa27b:~/mp2_1$ ./tsc -h localhost -k 9090 -u 2

===== TINY SNS CLIENT =====
Command Lists and Format:
FOLLOW <username>
UNFOLLOW <username>
LIST
TIMELINE
=====
Cmd> list
Command completed successfully
All users: 2,
Followers:
Cmd> timeline
Command completed successfully
Now you are in the timeline
p4
p5
p6
```