Gabriel Floreslovo
CSCE 438 - Distributed Systems
Section 500

MP2.1: A Scalable Tiny SNS

Features:

- Coordinator that assigns newly connected clients to an available server cluster
- Clients do not know implementation; only connect to coordinator once

Architecture:

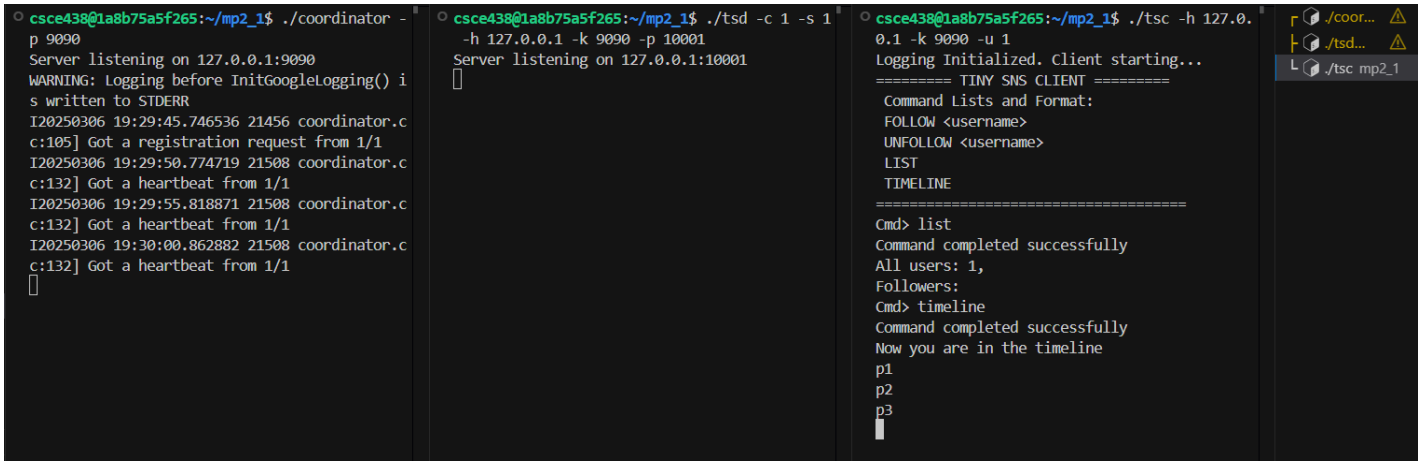
- Coordinator
    - Serves both SNS servers and SNS clients
    - Servers
        - Server opens gRPC channel as a client of the coordinator
        - Server begins sending “Heartbeats” to tell the coordinator that it is still “alive” or active
        - Coordinator stores metadata of all servers on all clusters as a “routing table”
            - This table is updated on every heartbeat (updating current metadata) or when a new server joins the SNS (adding new entry to table)
        - Coordinator periodically checks for “dead” or inactive servers by seeing if their last heartbeat was more than 10 seconds ago
            - Marks a dead server as having missed a heartbeat
    - Clients
        - Client opens gRPC channel as client of the coordinator
        - Client requests server from the coordinator
        - Coordinator assigns client a server, client connects and begins to interact with SNS as normal

Implementation:

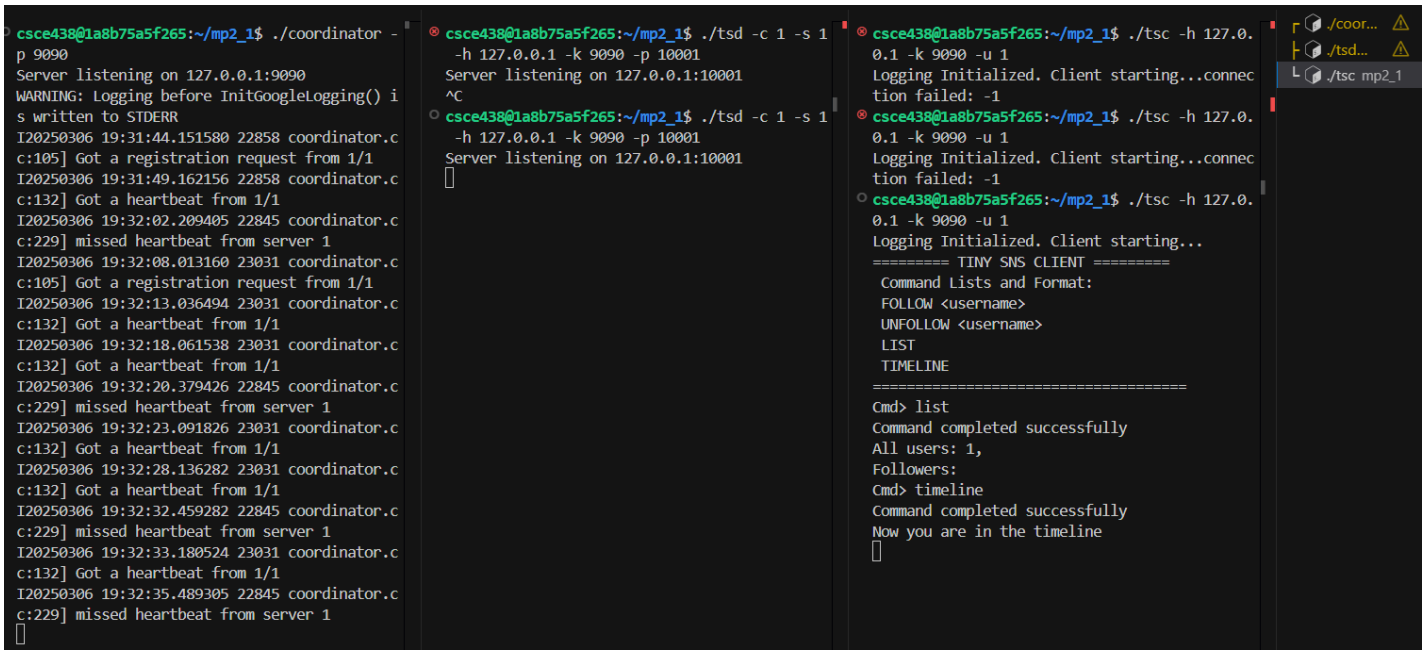
- Server
    - Initialize a server info struct for communication with coordinator in accordance with coordinator service protobuf definition
    - Create stub on coordinator channel
    - Create a heartbeat handler for a thread that will send heartbeat every 5 seconds with thread::sleep_for and chrono::seconds
        - In thread, use stub to send heartbeats to coordinator as rpc’s
        - First heartbeat should be a “registration” heartbeat to create the data for this server on the coordinator’s end
- Client
    - Create stub on coordinator channel
    - Use stub to request a server

    - Use returned server info to create normal server channel and corresponding stub
- Coordinator
    - Hold metadata table as vector of vectors: [clusterid][serverid]
    - Assign clients to clusters by (clientid - 1) mod 3 + 1
    - Use mutex when writing/reading server metadata so as to not create race condition between threads
    - Periodically checks to see if any servers are dead by looping through metadata files in a separate thread

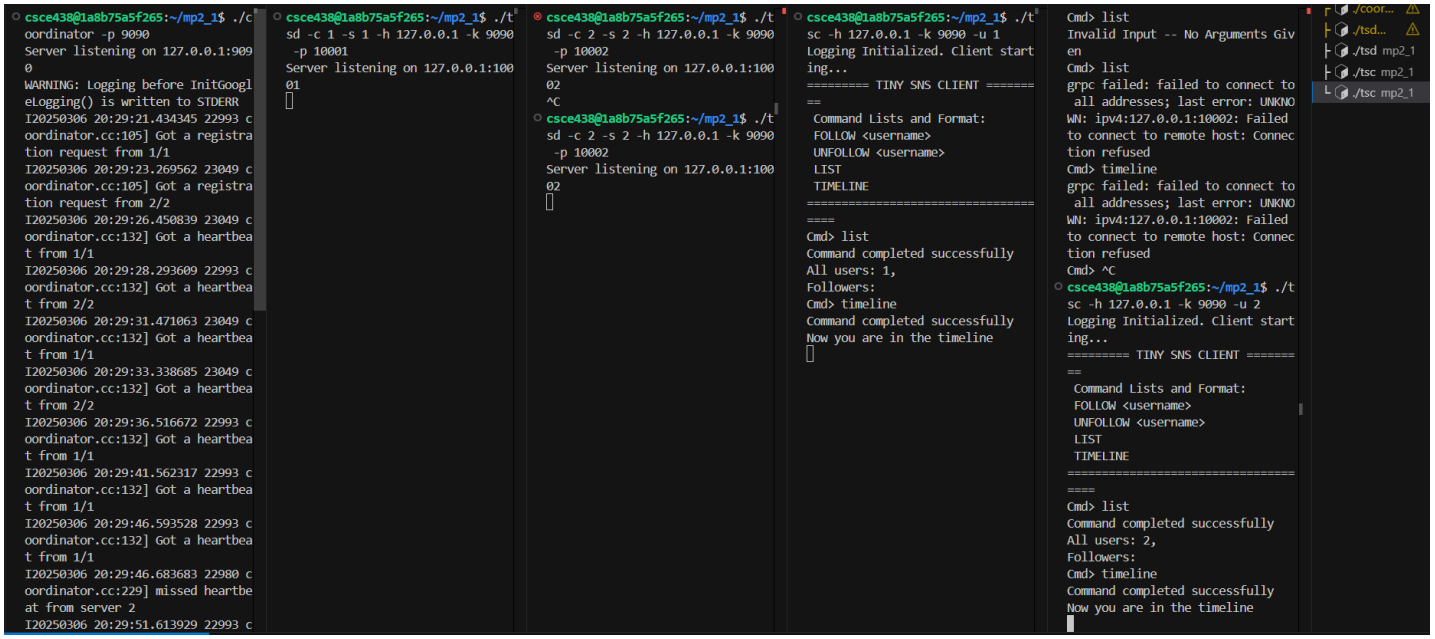
Test 1:



Test 2:



Test 3:


csce438@1a8b75a5f265:~/mp2_1$ ./coordinator -p 9090
Server listening on 127.0.0.1:9090
WARNING: Logging before InitGoogleLogging() is written to STDERR
I20250306 20:29:21.434345 22993 coordinator.cc:105] Got a registration request from 1/1
I20250306 20:29:23.269562 23049 coordinator.cc:105] Got a registration request from 2/2
I20250306 20:29:26.450839 23049 coordinator.cc:132] Got a heartbeat from 1/1
I20250306 20:29:28.293609 22993 coordinator.cc:132] Got a heartbeat from 2/2
I20250306 20:29:31.471063 23049 coordinator.cc:132] Got a heartbeat from 1/1
I20250306 20:29:33.338685 23049 coordinator.cc:132] Got a heartbeat from 2/2
I20250306 20:29:36.516672 22993 coordinator.cc:132] Got a heartbeat from 1/1
I20250306 20:29:41.562317 22993 coordinator.cc:132] Got a heartbeat from 1/1
I20250306 20:29:46.593528 22993 coordinator.cc:132] Got a heartbeat from 1/1
I20250306 20:29:46.683683 22980 coordinator.cc:229] missed heartbeat from server 2
I20250306 20:29:51.613929 22993 c
csce438@1a8b75a5f265:~/mp2_1$ ./tsd -c 1 -s 1 -h 127.0.0.1 -k 9090 -p 10001
Server listening on 127.0.0.1:10001
csce438@1a8b75a5f265:~/mp2_1$ ./tsd -c 2 -s 2 -h 127.0.0.1 -k 9090 -p 10002
Server listening on 127.0.0.1:10002
^C
csce438@1a8b75a5f265:~/mp2_1$ ./tsd -c 2 -s 2 -h 127.0.0.1 -k 9090 -p 10002
Server listening on 127.0.0.1:10002
csce438@1a8b75a5f265:~/mp2_1$ ./tsc -h 127.0.0.1 -k 9090 -u 1
Logging Initialized. Client starting...
========= 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
Cmd> list
Invalid Input -- No Arguments Given
Cmd> list
grpc failed: failed to connect to all addresses; last error: UNKNOWN: ipv4:127.0.0.1:10002: Failed to connect to remote host: Connection refused
Cmd> timeline
grpc failed: failed to connect to all addresses; last error: UNKNOWN: ipv4:127.0.0.1:10002: Failed to connect to remote host: Connection refused
Cmd> ^C
csce438@1a8b75a5f265:~/mp2_1$ ./tsc -h 127.0.0.1 -k 9090 -u 2
Logging Initialized. Client starting...
========= 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
./coor...
./tsd...
./tsd mp2_1
./tsc mp2_1
./tsc mp2_1