MP1: Design Document for SNS using GRPC

Nachiket Umesh Naganure
UIN: 532008698
CSCE662: Distributed Processing Systems

System Design

The goal of the machine problem was to design a simple social network service using GRPC and c++.

1. **Requirements :**
   (a) This network will have multiple users who can follow each other.
   (b) The users can post their updates on their timelines which in turn show up on timelines of users which follow them.
2. **Design approach Part 1: High Level Design**
   (a) We make use of GRPCs to satisfy our requirements. Mainly Unary RPCs and bidirectional RPCs.
   (b) A server is used to maintain the general registry of users and coordinating messages between users.
   (c) Clients use a ”Login” RPC when they first spawn up. The ”Login” RPC will help register the client with the server.
   (d) The server has a structure to maintain information of each client. Things like the follower_list and following_list corresponding to the client is maintained in the server.
3. **Desgin Approach Part 2: Low Level Design**
   (a) This won’t be like a LLD, but we will discuss a few interesting aspects of the system’s low level design.
   (b) FOLLOW, UNFOLLOW, LIST use basic Unary RPCs. And the server updates Client info accordingly.
   (c) In case of TIMELINE, we use a BiDirectional RPC, where the client initiates a RPC and two streams are maintained till the client volunteerly dies.
   (d) Here the client both writes to client stream and reads from server stream. And the server does the same.
   (e) Client writes to stream when the user posts something, while the servers writes to server stream when the current client is follower of some other client and the other client has posted something.
   (f) We have to maintain the Client Stream in the server(which will be used to send messages from server to client). This is done when the client initiates the BiDirectional RPC for the first time.
   (g) We use Threads to parallelise the reading and writing of messages on client side.
   (h) We also use threads to parallelise the writing of incoming messages to files on the server side.

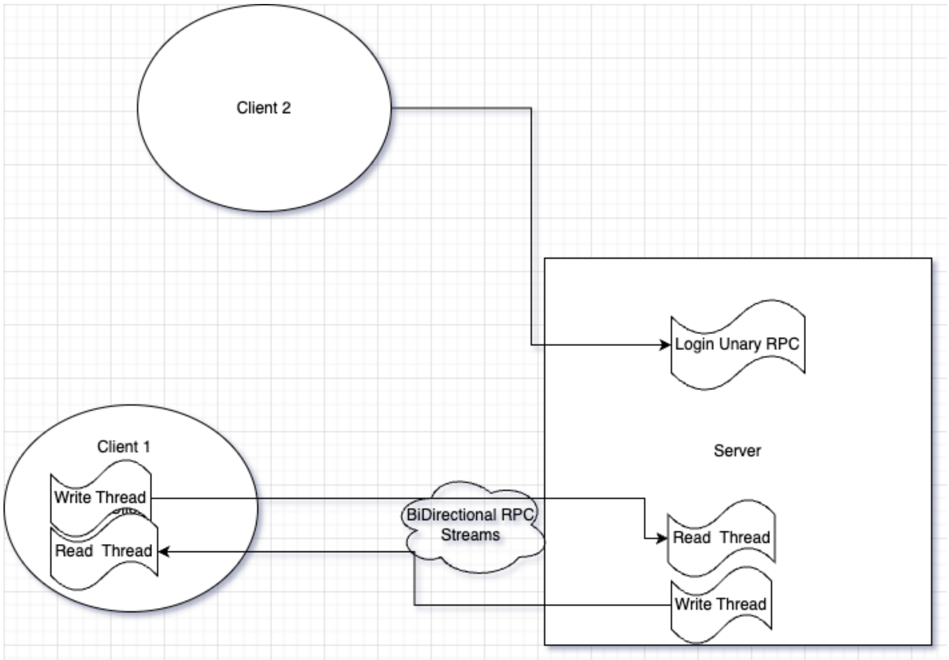

Figure 1: High Level System Diagram