

MP3 Design Document

Client-Master/Slave Interaction:

The client will begin by creating a stub and reaching out to the coordinator via RPC. The coordinator will return to it the port of the current master of the server that it belongs to according to the client's ID. The client will then use this port to create a stub for communicating with the master process of the server. All client choices will then be communicated to the server via RPCs through that port. The master/slave processes themselves will maintain files storing the changes of the clients belonging to their server.

master_{id} (OR slave_{id}) / {client ID}_following.txt : stores the clients that a client follows

master_{id} (OR slave_{id}) / {client ID}_feed.txt : stores the messages that a client writes in timeline mode

master_{id} (OR slave_{id}) / server_{id}_clients.txt : stores the clients that belong to the particular server

These files will only store local changes so master/slave will be the ones to maintain them, however they will be monitored by the synchronizer in order to disseminate the proper changes.

Synchronizer-Synchronizer Interaction:

The synchronizer for a particular server will be monitoring the local files that the master/slave processes are maintaining. If it notices a change has occurred in the last x seconds, it will first set up a stub to the coordinator and make an RPC in order to find the port of the necessary synchronizer. It will then create a stub to that synchronizer and pass along the changes that were made locally via RPC. The synchronizers will maintain files in their servers that store the global information. These are the files that will be altered during the RPCs.

master_{id} (OR slave_{id}) / {client ID}_followers.txt : stores the clients that follow a local client

master_{id} (OR slave_{id}) / {client ID}_timeline.txt : stores the messages that a client's followers write in timeline mode as well as what it writes.

master_{id} (OR slave_{id}) / all_clients.txt : stores a list of all of the clients in the system.

These files store the changes that relate to local clients by clients in other servers, so they are maintained by the synchronizer and accessed by the master/slave processes.

Coordinator-Master/Slave Interaction:

All master, slave, and synchronization processes will begin by setting up a stub and calling an RPC to the coordinator containing their information for the coordinator to store and give to clients. The master and slave processes will call a separate RPC every 10 seconds that the coordinator will store the timing of. The coordinator will continuously check this timing. If a process did not make one of these calls in the previous 20 seconds, that process will be made inactive.

Master-Slave Interaction:

The master process will begin its execution by creating a stub to the coordinator and getting the port of the slave process for its server via RPC. It will then create a stub to the slave process and retransmit the commands that it gets from the clients to the slave process. This will ensure that all local files are consistent between the master and slave. If the master were to go inactive, the client would then connect to the slave process and all information would be accurate and immediately available.

Timeline Decisions:

1. Every time the timeline for a client that is in timeline mode is changed, the most recent 20 messages will all be written to the client and displayed.
2. When a client follows another client with a non-empty feed (that client has written in timeline mode before), all posts from that client's feed will be added to the following client's timeline in chronological order with the previous posts in the timeline.

Other Notable Decisions:

1. Client IDs are used as usernames. An example of an output in timeline mode is "1 > hello 1"
2. The use of file storage was preferred over caching