Design Doc: MP3

Rohit Barichello

Server

The server code will largely stay the same as before. It will include a few alterations

- It will sort client text files into directories that are named appropriately so that the follower synchronizer can find it
- It will create a stub to communicate with the follower synchronizer and the coordinator to send occasional heartbeats
- It will also check the text files for the timeline and follower lists periodically to see if there are external updates. If there are, it will update the following list for its clients accordingly

Client

The client will be the same except it will use a separate coordinator stub to communicate with the coordinator to find out it's cluster. I'll also be adding a function to the sns.proto file for this action

Coordinator

- the coordinator will implement three new functions in the proto file.
    - One to receive heartbeats from the server and slave servers to make sure they're staying active
    - One to tell the clients what cluster they need to go to
    - One to tell the follower synchronizers which synchronizers to go to get the information they need
- The address tables for the servers and synchronizers will be stored as unordered_maps
- A vector of timer threads will spin off on execution to store information about the last ping from the servers. The threads will check for a ping every 10 seconds. If there's no ping, the slave will take over as the main server

Follower Synchronizer

- the synchronizer will use a timer thread similar to the coordinator one which will check for file updates every 30 seconds. If the files have been updated, then line by line those files will be sent to the corresponding follower synchronizers for updating on their clusters
- Each synchronizer will need to contact the coordinator using a special proto function to find the address of the new synchronizer. Then it will create a stub for that synchronizer and use grpc to send over the correct information