Donald Disha

Nick Macneil

CSCI-4510

10/15/2017

Project 1 Deliverables

In this project, we attempted to create a distributed twitter service that consists of N sites where users can tweet, block or unblock other users, and view their timeline through the programs UI. Once the program would start, it would check for pre-existing log and dictionary files present and load up the relevant data from those files if they were. Then, it would fork once, and one thread would handle the user input while the other thread handled incoming and outgoing messages. Our design uses a fixed configuration file that contains the IP addresses and ports for all sites on the distributed system and sets all sockets to their according IPs and ports. Once the local socket is bound, if our implementation worked correctly, we would connect to all other sites from that site and once a user inputs a command (tweet, block, unblock) our site would send a log of that event to all other sites after executing the event. Currently, our implementation works like a typical server-client connection that can read inputs and reply to whatever the client sends through the buffer with the same message the client sent.

Custom classes created:

Event class, which contains a constructor and accessors for the type, the originating user, the arguments of the event (message for a tweet, follower for a block or unblock), and the raw system time of the machine which the event was created on. The first three variables are all stored as strings, while the time is stored as a time_t.

Our implementation of the Wuu Bernstein algorithm:

- Instead of lamport timestamps we used raw system time from each individual process because we didn't want to have any issues with regions that were in other places on the earth.
- All tweets, views, block and unblock commands are treated as events with a variable to distinguish these events from one another
- We created a log file that captures events and updates that file for every event the user inputs.
- We create a dictionary file that captures all block relationships between a user and a follower
- Truncating the log is not currently functioning in the program.
- Due to the structure of our event data type, event blocks are handled one a time, therefore we insert at most once whenever we block or unblock.