Design

The high level structure is a client server model for communication. Each program has a server thread that creates threads to execute commands upon receiving a client connection request. Each program has a client object which facilitates starting and restarting worker threads that connect to pre-specified server addresses upon receiving a grep command. The main thread of execution handles input of the command and the output of the received results. The design is fault tolerant because the client workers are started by a starter thread. Whenever the worker threads fail, it sets a failed flag and notifies the starter to restart itself.

Result

My result for the 100MB machine.log is as follows:

4 machines:

grep –v Josephine 7059.18ms returned 413736 lines of results

grep –k Josephine 2953.31ms returned 413738 lines of results

grep –v noexist 332.42ms returned 0 lines of results

grep –v she 12810.50ms returned 1654956 lines of results

grep –k she 17862.82ms returned 2482423 lines of results

grep –v told 3292.22ms returned 413736 lines of results

grep –k it 3600.68ms returned 413736 lines of results

grep –v Josephine[[:space:]]who 6647.61ms returned 413738 lines of results

The average query latency is **6819.84ms.**

From the results shown, the query performance is quite related to the lines of results. The more lines in the result, the longer time it needs to complete. Also the key searches are generally faster than value searches. The reason might be that the keys are relatively shorter strings in the log file, so the regex search task is simpler.

Unit Tests

The unit test covers the correctness of log generation and implements the cases mentioned in the MP spec. I also added tests regarding the robustness of grep commands.