Distributed Log Querier

**Design**

The basic idea is that servers execute command getting from client through RPC, then return result back to client. In each VM, we start a Linux service which runs a bin (server), waiting for RPC with certain port. And in one of VMs, another bin (client) can be executed with a command (it can be any valid Linux command, including grep), and the command will be sent to all known servers through RPC asynchronously. Servers run the command locally and return output back to client and client shows those output.

**Test**

In the Unit test part, since the usage of our distributed log system is from command line, we chose to use exec.Command in the Test function to check if result is desired. First, we generated 10 log files including random words and some known pattern. We set the frequency of occurrence to be 1/5000 for the rare pattern. 1/50 for the somewhat frequent pattern and 1/10 for the frequent pattern. The we put a pattern to only one file, a pattern to some file and a pattern to all files.

We first test the efficiency of grep rare/frequent/somewhatfrequent patterns by comparing our program with local grep performance, and we set our test constraint to be runtime of our program < 2 \* runtime of local, else it fails. All performance tests passed.  
Then we test the accuracy of the grep. We tested grep the pattern that only occurs in one/all file. By analyzing the output of grep. And since occurrence in some files is random, it’s not testable so we just gave it a free pass

**Data/Plot**

Above plot shows the grep average time and standard deviation of pattern with different frequency. Each data points contain 10 trials. We can see that the average time is proportional to frequency and the standard deviations are pretty small. This trend is expected since the time is related to the number of matched lines.