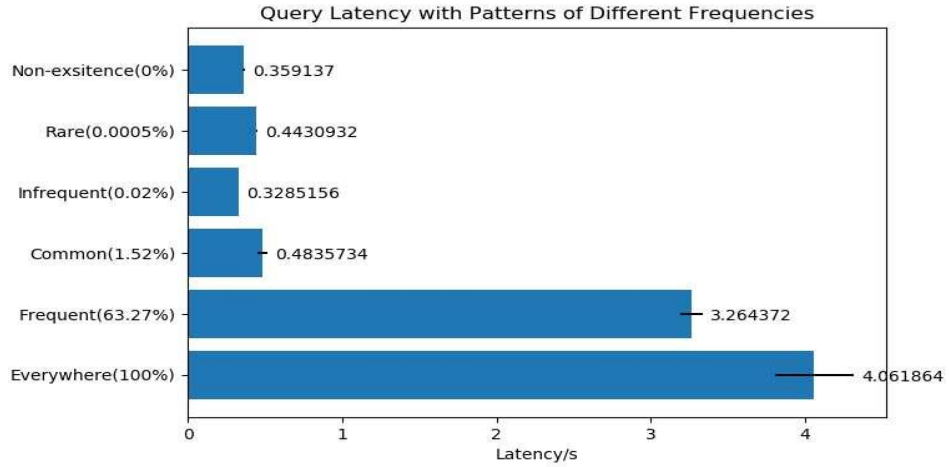
Report of Group 26

Design:

There is a server on each vm, each server keeps listening on a port. Each vm also has a client, which would send the grep request to all the servers. Each server would then run the grep query locally as an external command and send the output back to the client issuing it. Then the client would print the result from vm1 to vm10 to terminal if anything is received from that server. Trials were conducted to make it multithread to have each thread listening to a server, but experiments showed that single thread has about same speed. The application is fault tolerant with regards to certain servers down.

Unit Testing

The unit test we did is that on machine 1-5 there are scripts which randomly generate some files with preset known lines. Then test script grabs all the files from those machines and combine into a large file. The testing script would grep it locally and compare with the result with the distributed grep. There are also greps which only greps on the known lines which will be compared with handwritten answers, so we know for sure that our answer is correct. The unit test tried pattern with different frequencies, such as “ab”, “a*”.



Analysis of Plot:

The time in the plot above refers to the duration from when the client was started to the moment client finished saving the result into a file on disk. The grep pattern has a significant impact on the latency, even when the returned number of line is about the same. For example, we used pattern “abcd” for rare. It is significantly slower than pattern “abc”, even though abc returned 20 times more lines. Checking 60MB of files on each machine took the most amount of time when output size is small. Number of lines returned also significantly influence the latency, because i/o is time consuming. When we returned all the lines the latency is about 10 times higher than returning almost no lines. However, even at this rate we can see that grep pattern’s influence. Pattern ‘ab*’ returned about 63% lines of all log files, costing 80% time of the pattern which returned all log files.