Submitted by: chagari2, ntalasu2

REPORT - CS425 - MP1 - Group 59

Design

This program is implemented using a client-server architecture. One of the VMs acts as a client while the remaining VMs act as servers. Both the client and server machines implement multithreading which makes the program execution faster as a result of concurrent programming. It is possible that multiple VMs can act as clients.

The program starts when the user enters a grep query in the client machine. The client generates multiple threads, where each thread is connected to a unique server. The query is executed on each server, and the matching lines are sent back to the client where the results from each server are stored in separate log files.

Unit tests

Unit tests have been written to check for frequent pattern, infrequent pattern, regex and a failure test case. No testing framework has been used. Instead, function calls have been used. Client specifies a list of servers to test, and the data to be sent to the server. Each server generates a log file based on this input. The client then sends the grep command to each server to test a predefined pattern, and compares the results with pre-existing values present with the client.

Query Latency

The graph below shows the query output against the time taken for the query to execute. It is visible that greater the number of lines returned by the query, higher is the execution time. This is understandable as the program runs using socket connections, and higher the output lines, the more time it takes for the lines to be transferred over a socket network.

When running the program for a regular expression, the query runtimes are higher, as the parsing is done through each token of the log file as opposed to chunks of characters for string texts. The query point (x=4075) is an example of regex which deviates from the expected behaviour.

For the results below, the queries were run against 4 VM servers, each containing log files of ~70MB.