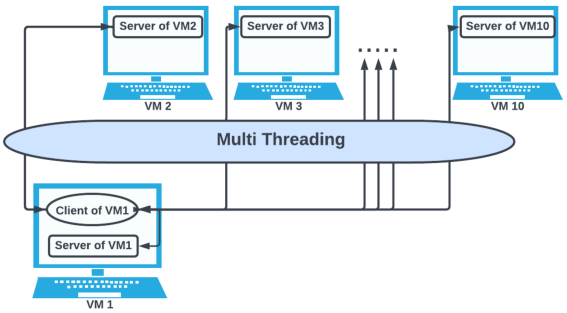
CS425 MP 1 - Distributed Grep

Girija Manoj Kumar Reddy Kalakoti (gmk6), Santosh Kumar Chejarla(santosh8)



I. Design

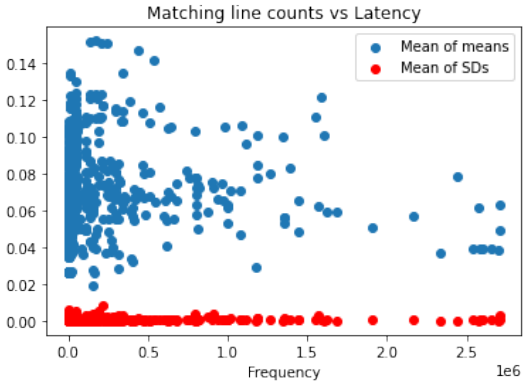
In our design, when a user queries from a machine, it will act as a client and send that specific query to all servers parallelly via multiple threads. Then the individual servers process the query and send the result of matching line counts to the client. The client then displays both individual results and the aggregate to either the terminal or to the webpage as per the request of the user. The query is cached at the server level so that the server gives response to repeated queries quickly. The user can also choose between cached and uncached.

II. Unit Tests

For the unit testing, we used pytest package. We preprocessed the demo log files and got the frequent and infrequent strings. We then validated the answer returned by the server to the expected answer from the original grep. Our 40 unit tests include frequent patterns, infrequent patterns, frequent texts and infrequent texts each having 10 unit tests. We can run these with tests.py on any one of the VM.

III. Performance

We had taken ∼1400 strings and plotted the following graphs. As we can see from the plot of **"Matching Line Counts vs Mean Latency"**, We did not observe any peculiar behaviour in the latency of in the Frequent and Infrequent patterns.



This makes sense because irrespective of the frequency, the grep function of server has to search the entire log file of that particular VM and then the server just sends the matching line count to the client. Given healthy networking situations, time taken for sending the matching line count to client will also be more or less the same. Coming to the plot of **"Length of the query vs Latency"**, We can observe a clear drop in latency as the query length increases, this is in-line with the nature of grep, as it uses the **Boyer-Moore algorithm**. It searches through the lines with size of the pattern length. Standard deviation of both the plots is almost the straight line which indicates that there is not much deviation in latency of the five trails of each query.