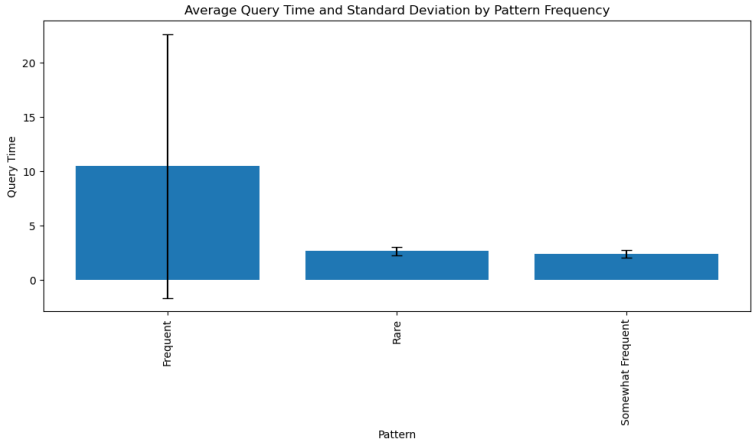
MP-1 Report by Colin Zhou (colinz2) and Keyang Xuan (keyangx3)

Algorithm: Our group decided to have two files; client.py and server.py. The client file will be responsible for sending the query to the server file. It will also be responsible for receiving the results from the server files and then printing it out in the terminal. But most importantly, the client will be launching separate threads to each host to initiate the socket connections. The server will be running on each host, searching the log files for the specified query pattern, and then sending the result back to the client. It will also dynamically add in flags depending on what is specified in the query.

Tests: We used a separate file (with a random number generator) to generate log files for testing. Five log files were generated at random on five different virtual machines. two of the VM's had frequent patterns, one had rare patterns, and the other had a somewhat frequent pattern. All patterns were log files that had a random string of alphabetical characters and numbers of length 40, but the number of numerical digits in every string were changed for each pattern frequency. Five patterns of each frequency were generated and then tested on all of the five running VM's.

	Pattern_Frequency	mean	std
0	Frequent	10.494345	12.166450
1	Rare	2.685382	0.387022
2	Somewhat Frequent	2.420276	0.349252



The deviation of the frequent pattern was much larger, possibly because the frequent patterns took much longer to execute on certain log files, and also that a pattern that was frequent in one file may have unexpectedly not yielded a lot of results in certain files. The average latency time for the rare pattern was expected, since a rare pattern will not occur in most files. The somewhat frequent pattern was surprisingly similar to the rare pattern, possibly due to the fact that not all logs would have that kind of pattern.