Ragini Gupta, NETID: raginig2
Yinfang Chen, NETID: yinfang3
Team Number: **42**
CS 425 MP-1

Description: Client.py and Server.py

Client.py is the querier and Server.py queries the log files using local grep PATTERN. Client.py takes two arguments; flag (-Ec or -c) and the PATTERN. The Client.py aggregates the output returned by each querier to return total matching lines from all log files and number of matching lines from each log file VM.

Log files generation and Unit testing: LogGen_UnitTest.py is used for generating Log files for frequent, infrequent, somewhat frequent, unique (only on one machine), and common patterns (pattern on all machines logs). To control occurrence of each pattern, probability distribution is used. A count of each type of pattern is maintained in the Patterns_Count.txt file to verify the output of distributed log querier. UnitTest.py is used for writing test cases using *Unittest* Package for Python. It verifies the output (# of matching lines) returned to the Client querier with the count maintained in Patterns_Count.txt file.

Latency Performance: We ran the query for three different patterns across the provided log data files (size ~60MB for each of 10 VMs). The time cost for searching and fetching each query output from 10 VMs is calculated in *Client.py* file. Each pattern is queried 10 times.

Pattern	**Latency (seconds)**	**Average**	**Standard Deviation**
Frequent Pattern: String used for Testing- "http://www.thompson.com" *This pattern exists in all VMs, count of this pattern=3295*	[0.2618, 0.1961, 0.1740,0.1823, 0.1780, 0.1768,0.1778,0.1724, 0.2055,0.2020]	0.19267	0.027
Infrequent Pattern: String used for Testing- "http://king.com/privacy.asp" *This pattern exists in VM7, VM5, VM10, VM9. Total Count of this pattern= 5*	[0.0940,0.0873,0.0987,0.0809, 0.0882,0.0783,0.0734,0.0744, 0.0810,0.0776]	0.08338	0.008145
Regex Pattern: "-Ec ([tT]hompson)|(sawyer).com*"	[0.2413, 0.2218, 0.2231, 0.2215, 0.2239, 0.2210, 0.2240, 0.2255, 0.2221,0.2317]	0.22559	0.006325

From the above table, we can infer that for infrequent pattern the average latency is the smallest whereas for the regex pattern the average latency is the highest. This is expected since the occurrence of infrequent pattern is much lesser than the frequent pattern across all log files. Also, the infrequent pattern starts with http://king instead of http://www. Therefore, when the initial of the log entry does not match with the initial of the infrequent pattern which is http://king, the program skips the remaining search for it and moves to the next entry. This saves the search/matching time drastically for infrequent pattern. Regex pattern has the highest average latency as it searches for the presence of different combination of strings i.e. Thomson OR thompson OR sawyer from the log files.

GitLab: https://gitlab.engr.illinois.edu/yinfang3/cs425-mp.git