CS425: Distributed Systems – MP1 Report

Gopalakrishna Holla V (hollava2)

Alok Tiagi (tiagi2)

Solution overview:

Assume that a client has connected to one of the servers of a distributed system (with N machines). When the client fires a query, the following events occur

1) The receiving server (S) processes the query.
2) Processed query is sent to the N - 1 peers of the server. The connections are threaded.
3) Each peer executes the query on its local log and stores the output in a file after tagging.
4) Each peer sends his tagged output file to S. The threads, which sent out the commands, receive the file too.
5) S executes the query on its log, creates an output file and tags the file.
6) The tagged output files from all the peers are collected, merged on S and the merged output file is then sent to the client.

Query and file transfers are achieved through TCP using the sockets library. The commands are executed as system calls to UNIX.

Unit tests

This unit test checks to see if the return of the grep query is correct. The algorithm is,

1) Unit test runs on a single test machine (T)
2) If using static log files, jump to step 5
3) Each log file is created by writing random lines, inserting a few known lines and inserting random lines again.
4) The log files are then sent to the relevant machines.
5) T spawns a client and runs a particular query for which it knows the output.
6) T receives the processed output and compares it with the expected output
7) Success if no differences found! Failure otherwise.

The average latency (time difference between query fire and receipt of the output file) for 4 machines each with 100MB logs is 4.2s for a frequent pattern (50% of the log) and 1 second for a rare pattern (10% of the log).

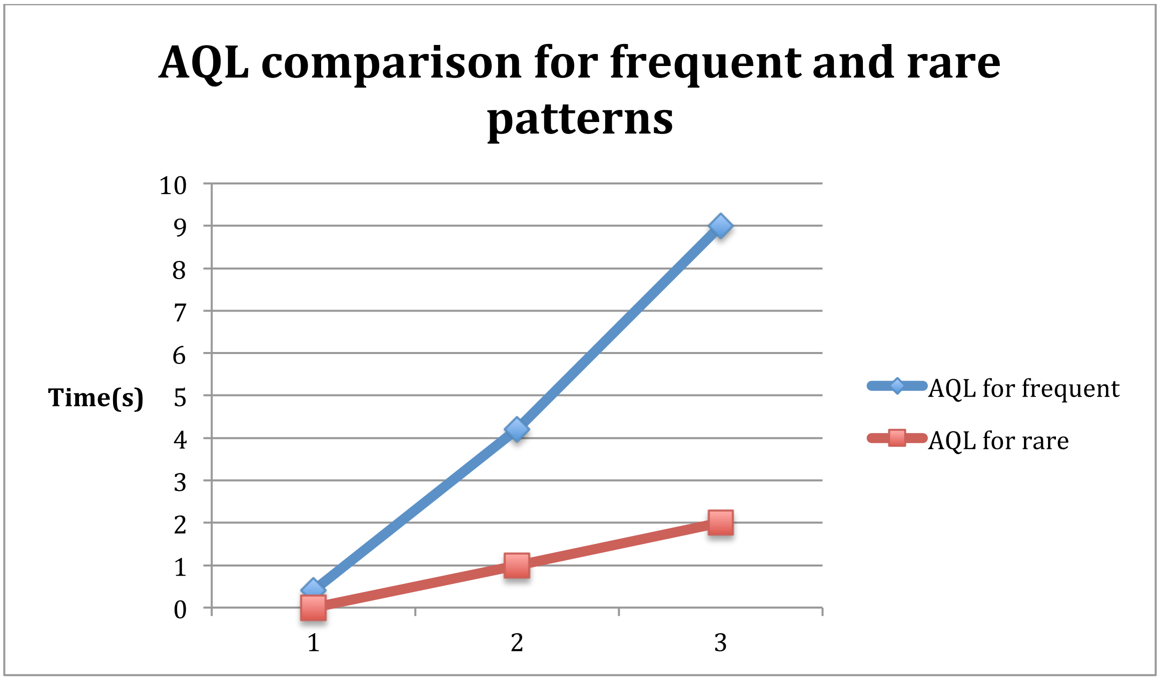
Extra Credit Report

Raw Data

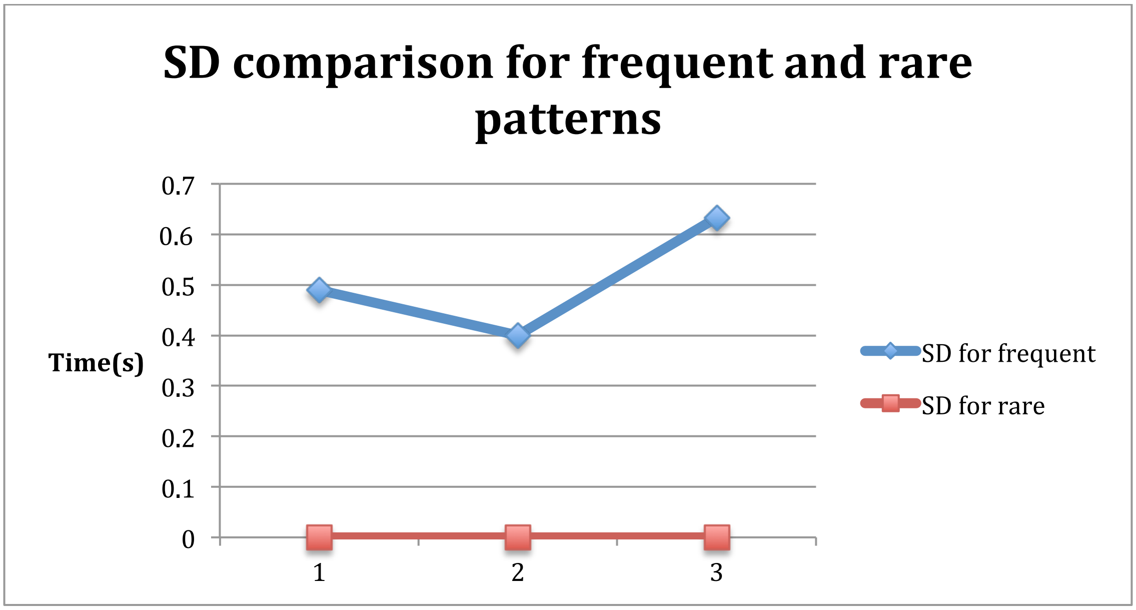
- All time readings in seconds
- Freq – Frequent pattern which makes up 50% of the log
- Rare – Rare pattern which makes up 10% of the log

Size ->	11 M		103 M		256 M	
Sl.no/Freq	**Freq**	**Rare**	**Freq**	**Rare**	**Freq**	**Rare**
1	1	0	4	1	9	2
2	1	0	4	1	9	2
3	0	0	5	1	8	2
4	0	0	4	1	10	2
5	0	0	4	1	9	2
Average	0.4	0.0	4.2	1	9	2
S.D	0.4899	0	0.4	0	0.63246	0

Graphs of average query latencies and standard deviations



AQL – Average Query Latency



SD – Standard Deviation

Comments on the AQL graph

- Most striking observation is the fact that the AQL increases linearly with log size.
- Relevance to our design – The threaded nature of the design makes sure that any latency is mostly due to the grep operation and the network transfer time, and not any processing overhead.

Comments on the SD graph

- The 256 MB file shows a much higher deviation from the mean. This could be due to network latency and jitter that can have a bigger impact on larger files.

Overall observations

The occurrence of the frequent pattern is 5 times that of the rare pattern. The AQL also shows a similar behavior, which means that the network transfer time is the dominating factor in our system (since the grep is on the same file both times)

Related observations

A second set of readings was taken at a time when network traffic was higher. A grep on the frequent pattern for 103 MB log files yielded an AQL of 19.6s! This further asserts the dominance of network transfer time in the overall query latency.