Report

CS425 -MP1 - Group 9

Design Overview

A simple Client-server architecture has been used. All the machines from which log files have to be queried must have the server (called grep-server) running. A client (called grep-client) takes grep queries from user and displays back matching lines.
The processing paradigm is as follows. The client takes the grep query from user and spawns multiple threads (each talking to one server). Each thread relays the query to its server. In the server, the grep query is executed locally and the matching lines are relayed back to the client thread. The client thread then prints this matching line.

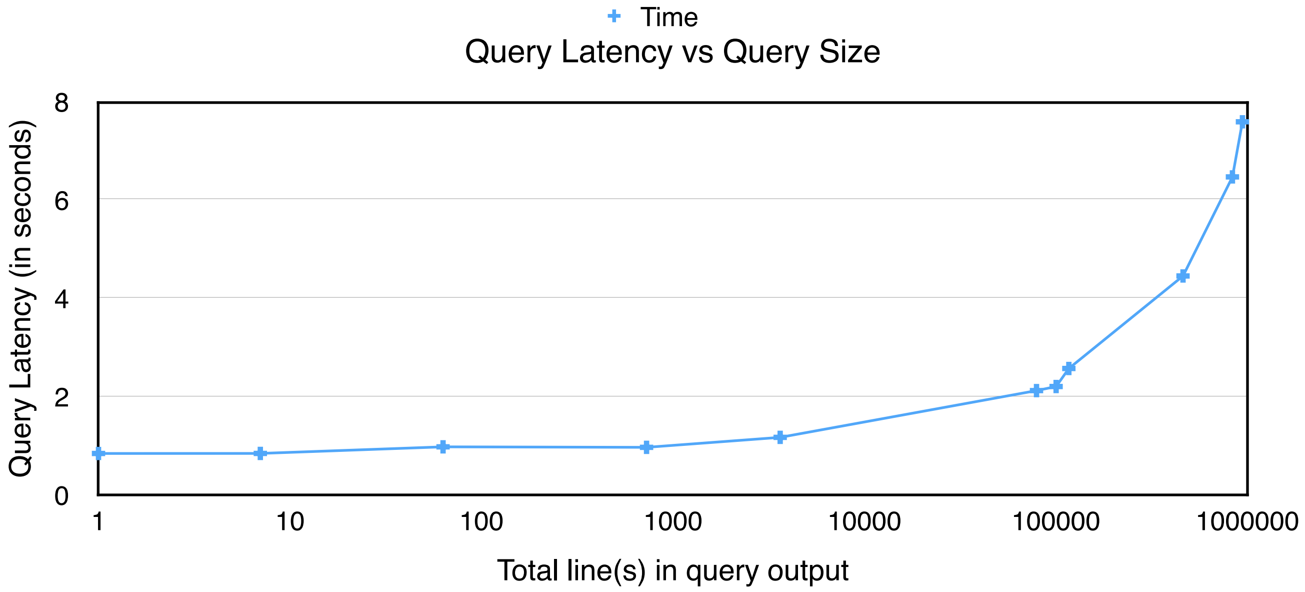
We have designed **Multithreaded client and Multithreaded server** to ensure speed, fault tolerance and scalability *(multiple distributed grep jobs can run concurrently)*. Also our application is **Fault tolerant** takes care into account scenarios when **server is not running** or **crashes in between a job.**

Unit testing

For unit testing we used JUnit-4.11 in Java. We created predefined test files by running grep on web proxy logs provided and verified the grep output with our own written distributedGrep application. We used various unit tests on comprehensive set of queries like *Simple Pattern, Regex, Advanced Regex* together with many combination of flags including *[-i Ignore Case, -n Line Numbers, -v Invert Match, -w Word Match, -x Sentence Match]*

Query Latency

Time taken by distributed grep client highly depends on the query output as the servers need to transfer query output using sockets over network. The lower the query output, lower the query latency. The graph below shows the relationship between Query Latency and Query Size. The queries used for this analysis were simple pattern search running on 4 server's with log files(vm1.log,vm2.log,vm3.log,vm4.log) with total size of 100MB approx.


+ Time
Query Latency vs Query Size
Query Latency (in seconds)
Total line(s) in query output