Machine Programming 1 – Distributed Log Querier Report

Design

For the log querier system, we designed a centralized system by using one client and multiple servers to run this system. For the connections between client and server, we utilize socket and TCP to build up all the connections. The first step is to create a TCP socket, bind it to port 7000, and listen for incoming connections. Second, we use select() to make the server can handle multiple clients. Once it gets a client request, it would call the grep_log() function to calculate the result and send it back to the client. For the client, we create a TCP socket as well, and once it connects success to the server, it sends the user's grep command to the server and prints it out after receiving it from the server. Besides, we also make this system fault-tolerant, if a server crashes, the client point would print out the failure message for the crash server and other servers would not be affected.

Unit Tests and Analyze

-Same frequency of query with different thread comparing

We compare the multi-thread and single-thread performance. For single-thread, it sends a grep command for one server each time. By contrast, multi-thread send a grep command to all the server at once. The comparing result shown in Figure 1, the multi-thread has better performance in all the frequency queries. Most of the multi-thread average querier latency is under 0.1 seconds, the best one is 0.042 seconds, and all the standard deviation is very small, which is around 0.001.

-Comparing regular expressions and common strings by using multi-thread

Besides comparing different thread log queries' performance, we also compare the average querier latency of common strings and regular expressions. As Figure 2 shows, the common string grep command has better performance than regular expression in all frequency patterns. We thought the reason is regular expression needs to be parsed, which causes it to be slower than string operations.

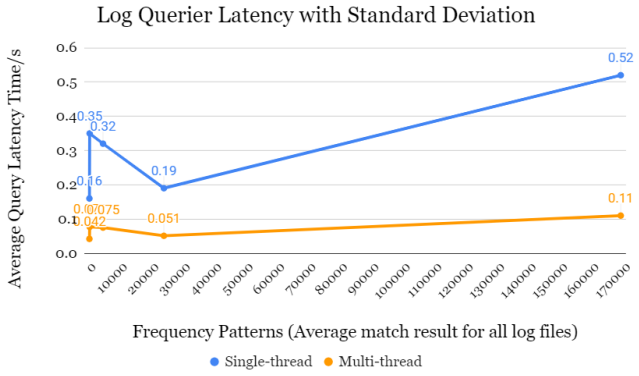

Figure 1. Log Querier Latency with single-thread and multi-thread

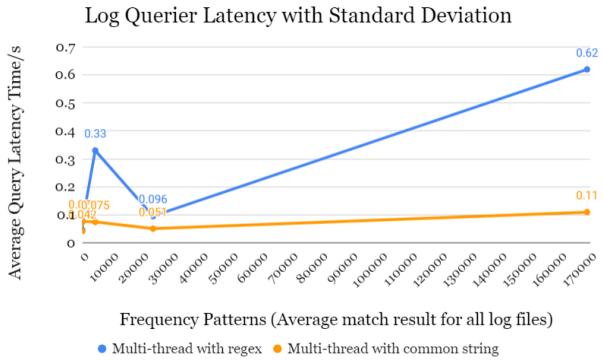

Figure 2. Log Querier Latency with regular expression and common strings

Pattern	Frequency	Linecat	Mean(sec)	StandardDeviation
/paul.info/posts/	rare	0	0.042	0.0033
sawyer.com	somehow rare	80	0.077	0.0054
156	medium	4330	0.075	0.0054
Aug	popular medium	23683	0.051	0.0025
.com	frequent	168999	0.11	0.0074

Table 1. The standard deviation and mean for each frequency pattern (multi-thread)

Besides the above analysis, we noticed some interesting things. When we run the log querier system for the first time, it would spend more time to return the results, and we thought the reason is the server would store some information in cache, and for the following request that can utilize previous calculate results. Moreover, the frequency patterns do not affect log querier latency performance, we thought the reason is whatever the frequency pattern in log files is frequent or rare, it still needs to scan all the text in the log files.