MP1 Report - Group#38 - yiteng3(Zhang) & maojunx2(Xu)

Project Design

We seperated this project into to modules: server and client. We took charge of one module's designing and implementing task and then debug together.

Client Design Java's Socket and Thread have been used to design and implement. Client is the main class, it's main function controls the interaction flow and it's callServers(command, query) can establish socket connections to servers list in the property file. MySocket extends Java's Thread so it allows clients to establish socket connections to multiple servers parallelly, significantly decreasing time consuming. Result class is designed to stanardiztion response messages from servers, since results from servers can have different types veried according to grep's argument (for example, grep -c return numbers needed be sumed up, grep return strings).

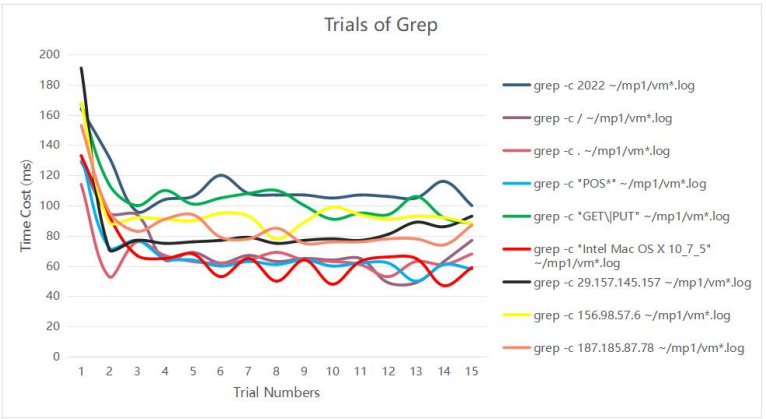
Server Design A multi-threaded scheme is used. For each request listened to, the server allocates a sub-thread to process the request. The parsed request parameters are passed to the Runtime method to invoke the system Shell command. For the result returned by the system, the server reads no more than 60000 bytes each time and writes the return result to the output stream of the socket in a uniform way.

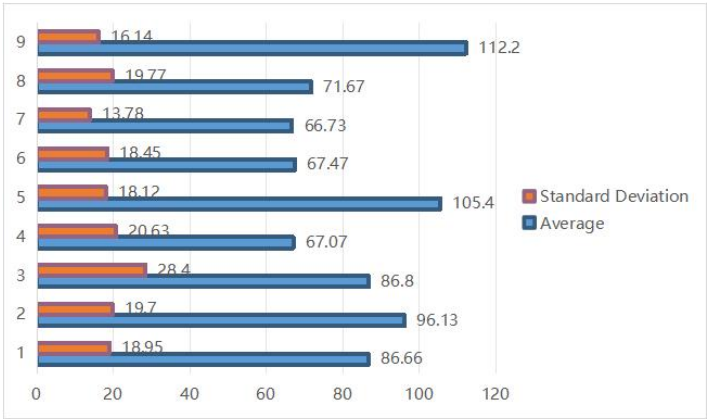
Test Design

Generate log files The format of the tested Log files is the same as Demo Data, consisting of IP, date, request header, and browser properties. Each part can be generated randomly. The default generated logs contain 100 random logs and 3 fixed logs.

Unit test First, it creates several MySocket objects to establish socket connections to servers list in properites. Then a "test" message will be sent to each server to inform them generating log files. After servers' responses being collected, which is log generated by each server, logs will be write to a local file called "logging.log". After that, for each query listed in test function, a grep message will be sent to servers. Expect value can be calculated by execute the same grep command on local "logging.log" file. If reponsed result value equals to expected value, it will be determined as PASS, otherwise FAILED.

Test and Analysis





The query latency is highest at the first request and stabilizes afterwards, due to initialization socket between client and servers.

The query samples on the right side of Figure 1 correspond from top to bottom to serial numbers 9 to 1 in Figure 2.

It is easy to find that there is no absolute relationship between query time and stability whether the sample is common or not. We speculate that the latency is related to how early the query pattern appears in each line, and the earlier it appears, the lower the latency.