TCSS 558 Applied Distributed Computing

**Execute Summary**

Yaqun Yu, Xiao Liang

**Assignment Overview:**

The purpose of this project is to use two protocols TCP and UDP to perform three basic client/server functions: PUT, GET and DELETE. PUT function contains two augments KEY and VALUE, while GET function and DELETE function only need argument KEY. The server should always running so it can receive requests from the client at any time, and it will display the requests it received and the responses by itself. The client should be robust to server failure by using a timeout mechanism. The client is able to read commands from a script containing a command set, which will be discussed later. The communication between server and client, to be more specific, the client’s request and server’s reply and all kinds of exceptions, are all recorded into both server log and client log. Based on the assignment, we need also create a makefile for users to build both the server and client code bases into executable binaries, .class format. For the test set, we need to implement 50 distinct GET/PUT/DELETE transactions on 10 key-value pairs, which contains at least five of each operation: 5 PUTs, 5 GETs, 5 DELETEs. During the communication, since the client and server are single-thread, we can only execute one command at a time. The TCP protocol can guarantee package delivery, while the UDP protocol is more flexible and the sender will not wait to make sure that the recipient will be receiving the package.

**Technical Impression:**

During the implementation we created a hashmap class to define our storage and three functions easy for future recall. Then we generated a log class to record all the information generated by our client and server. By defining different file path, we can create different log files for server and client separately. Next, for TCP server, we have an argument gives the port number. After creating the log file for server, for each request we received, if there is errors during communication, record this into our log, if successfully operated, reply to client with the correct information and then output the key, value or value and also record into our log. For TCP client side, it will send the request and get the output or error from the server side. During the execution of our 50 test set, since TCP need to wait for server reply, it should be slower than UDP protocol in theory. But in our practice, we can rarely tell the speed difference. For our UDP protocol, the implementation is pretty similar, the delivery of our package is not guaranteed but the speed is improved in theory. For our test data we tried a shell method but resulted in confusion causing. So we switch to reading command directly from text file. We documented the usage of the test set in our document.

During the implementation we were a little bit confused about the test set at first, but it had been clarified by the professor from last class. Doing this project helps us understanding how TCP and UDP protocol works in real examples, as well as the main differences between them.