Client-Server Storage

Thomas Michael O’Brien

CMSI 402: Senior Thesis

Loyola Marymount University

*2.0: Preliminary Proposal Document*

The goal of the Client-Server Storage application is to create two separate applications, a client and a server, that can work in unison to send files from one machine to another. This application will require extensive knowledge in computer network architecture and the TCP/IP stack. A project like this will allow me to exercise my understanding of computer networks fundamentals and at the same time, enhance my programming skills in the language Go. Continuing to develop these technical skills will greatly prepare me for the career path I wish to go down.

3.0: *Proposal Document*

**Verbal Description**

At its core, the purpose of this thesis is to create a client-server application built from first principles. The purpose of this application is to allow a user to spawn up an instance of the application as a server. The user must compile the server application into a bash command called server. Then, the user will give the server command 2 two arguments: an available port number and a directory. The directory is used as the location where the server will store arbitrary files it receives. The size of the directory will initially be limited to 50 megabytes. If the directory were to go past the initial limit size, the server will dynamically allocate double of the current memory limit.

In order for a user to send an arbitrary file over to a listening server, the user will be required to compile the client application into a bash command called client. The user will then give the client command 3 arguments: a hostname/IP, port number, and file name. If any of the arguments given are not valid, then the client command will immediately return an error. Assuming the arguments are valid, the client command will attempt to make a TCP connection with the server application at the appropriate hostname and then following that, the proper port number. To do this, the TCP request will go through a handshake process and return whether or not it is successful. Once a connection is established, the client will turn the file into packets, send them over to the server through the established connection, and put the packets back together in the directory specified by the server.

This application will be written in Go, a system programming language written by Google. Go boasts its fast compile and run times due to not compiling to another language. These fast run times will be critical for the speed of this application. Not only that, but these applications will be Dockerized in order to condense its size and memory usage.

**Justification**

Creating a client-server storage application will require a mastery of the courses offered at Loyola Marymount University with a focus in the following: Data Structures, Systems Programming, Programming Languages, and Operating Systems. These courses have given me the fundamentals in how to write optimal code and an understand of how machines function on a lower level. By using this knowledge, this gives me the tools to expand my knowledge in the field of computer networks.

The main tool required to create this application is simply my Macbook Pro and free software such as Terminal and Atom. A Raspberry Pi will also be used for testing purposes, however is not necessary for the success of the thesis.

While this application might be on the more difficult side given the amount of self-teaching and research required, this is a field of computer science that I want to pursue in my future. Given my previous experience, I aspire to be either a Site Reliability Engineer or a Junior DevOps Engineer right out of college. Developing a senior thesis so heavily involved with network architecture would be an excellent way to broaden my knowledge of this field.