**Socket-Based Network File Sharing**

**Cloud Server Project**

CNT3004.03

Joseph Romeo

https://github.com/jromeo3415/Cloud-Server-Project.git

11/3/24

**Introduction**

For this project we are going to implement a server application that will store files and allow clients to perform specific operations to files within its directory. This server will create a directory if one with the desired name does not exist and will use this directory to store all the files that the client stores/uploads or wishes to download. Clients can perform numerous operations, like uploading or deleting files for example. Clients interact with their side of the application through command prompt and will need python installed on the respective device. To connect to this server, clients must know the servers’ IP address and port number. If this application is being ran on a Google Cloud compute engine, ensure the firewall will allow traffic on the desired port, as it is typically blocked by default for most ports.

**Objectives**

* Implement connect, upload, download, delete, dir, and subfolder commands. These functions allow the client to interact with the files in the servers’ directory. These functions should work with most file types and update the client on their status or issues.
* Authenticate Users. Users can add their desired username and password to the dictionary contained in the server .py file. Clients must sign in with a recognized login to interact with the server, which increases system and file security.
* Utilize multithreading. Multithreading will allow the server to handle multiple clients at the same time. Increasing the amount of concurrent users will improve usability and client experience.
* Create functions to analyze system performance. The application will automatically calculate and log performance metrics like upload/download speed, and server response time. These metrics will later be used to analyze the performance of the system.
* Employ error handling for all functions and operations. Every function should handle errors that may be encountered during the programs use.
* Host the server on a Google Cloud compute engine. Making the server cloud based allows users to store and upload files practically anywhere they can connect to the internet.

System Architecture and Design

Implementation details

Experimental Results

Problems

Lessons

Individual Contributions

Conclusions