**Homework1 Write-Up**

Screenshot of Server.cpp

A screenshot of a cell phone

Description automatically generated

Screenshot of Client.cpp

A screenshot of a social media post

Description automatically generated

**Performance Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Repetition** | **Number Buffer** | **Buffer Size** | **Sending Time** |
| 1 | 20000 | 15 | 100 | 931184 |
| 1 | 20000 | 30 | 50 | 1748676 |
| 1 | 20000 | 100 | 15 | 5741513 |
| 2 | 20000 | 15 | 100 | 387337 |
| 2 | 20000 | 30 | 50 | 166487 |
| 2 | 20000 | 100 | 15 | 130114 |
| 3 | 20000 | 15 | 100 | 78418 |
| 3 | 20000 | 30 | 50 | 76973 |
| 3 | 20000 | 100 | 15 | 80008 |

**Type-1 Transfer: Multiple Write**

The type 1 transfer had a very large difference between the 15 & 100 buff and the 100 & 15 buff. Changing the variables for the buffer number and the buffer size proved to increase the time by a very large marginal amount.

A screenshot of a cell phone

Description automatically generated

**Type-2 Transfer: Writev**

For type 2 transfer, the writev was very interesting as it proved to react the complete opposite to type 1 where buff 15 & 100 turned out to take the longest amount of time compared to 100 & 15 which took the least amount of time. However, the middle buff variable which was buffer number 30 and buffer size 50 seemed to not have a large change between the two different write types.

A screenshot of a cell phone

Description automatically generated

**Type-3 Transfer: Single Write**

For the type 3 results, I was the most surprised at these resulting timers since this type of write took the least amount of time out of all the trials that had been done before for any other type of write no matter the type. The speed on the type 3 write had almost more than double the speed of the fastest times for type 1 and type 2.

A screenshot of a cell phone

Description automatically generated

**Discussion**

If this same test was done on and run on a slower network, for example a 1 mbps upload speed, then the amount of time for each type to run its test would increase at the same amount the mbps were increased. Lets say I had a 12 mbps upload speed so then multiply the results I got by 12 and so you would have speeds that were 12 times as long as my test speeds for each type.

You would want to use a thread to serve the connection rather than serving it in the main function because it allows the computer to service more requests at the same time. Currently when computers send requests to servers for connection its usually never just one request its usually a multitude of requests. So by having a thread to serve the connection, you can serve multiple connections at once which speeds up the process like multi-tasking so you can complete more requests in a shorter amount of time.