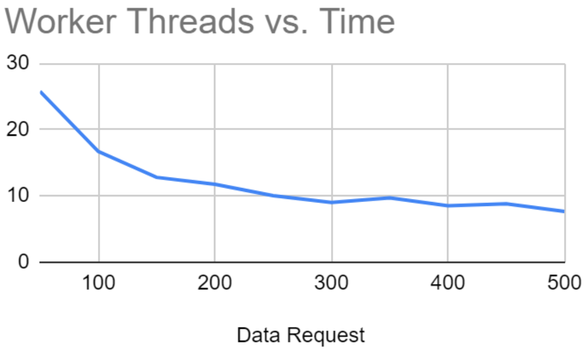
Nathaniel Trujillo

File Transfer Tool Report

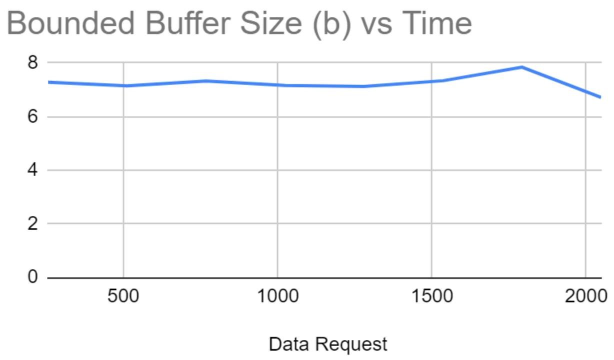
Design

The overall design of the program is identical to a previous project assignment with the only change being how data is transferred from server to client. In this project, data is transferred over the network. In order to achieve this, the FIFOreqchannel class was overhauled to a TCPreqchannel class structure. The functionalities of TCP allow the client to create a socket and connect the socket to a desired hostname and port. It also allows the server to set up a socket binded to a port on the machine which also listens for any possible connections from the client. The server accepts all incoming connections from the client and sets up a new file descriptor to read and write between client and server. This accepted file descriptor is how data is transferred between the client and the server.

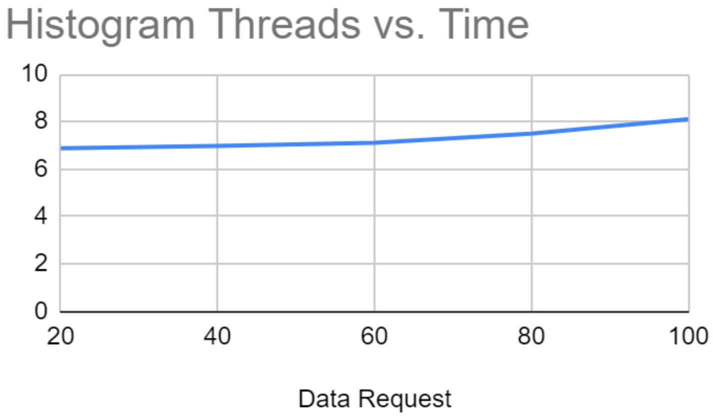
Data Requests



When analyzing the worker threads vs time graph, it is seen that as worker threads increase time dramatically decreases and performance increases. This is likely due to the fact there are more threads handling the actual data transfer at the same time. This is an expected behavior.

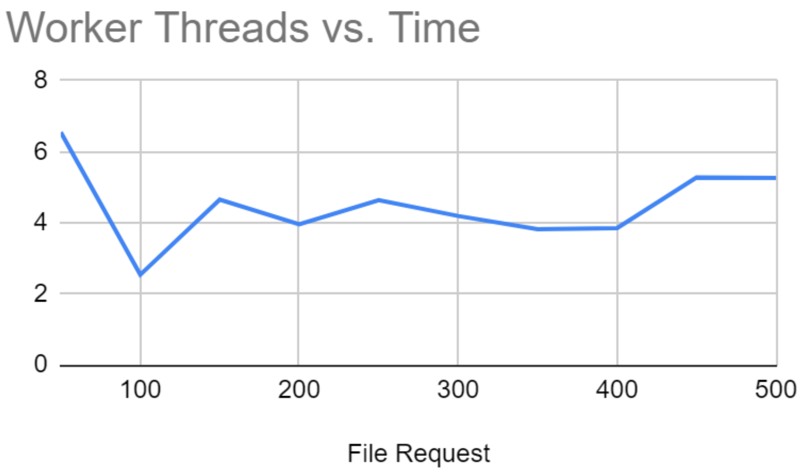


When analyzing the bounded buffer size vs time graph performance remains constant. This is likely because the bounded buffer never reaches a point where it gets full. This experiment was done with 500 worker threads meaning there was plenty of time for the buffer to be cleared out in time.

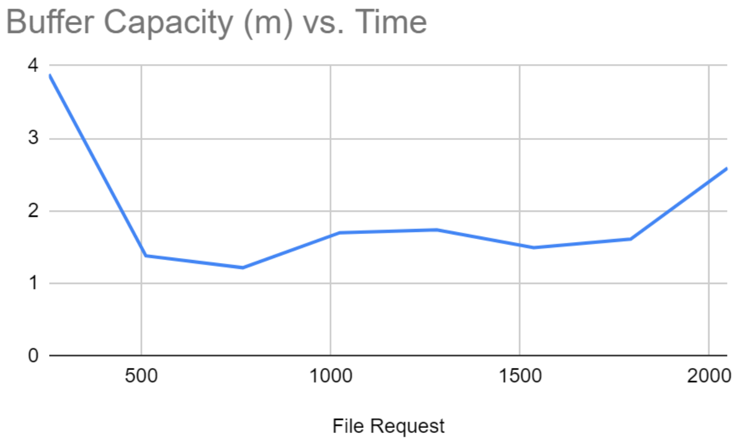


When analyzing the histogram threads vs time graph, it is seen that performance remains relatively constant. There is a slight decrease in performance, but this is likely due to other factors such as network connection. The constant performance is likely due to histogram threads receiving data slow enough where more threads would affect the performance.

File Requests



When analyzing file requests and their relationship with worker threads it becomes rather difficult to determine what the effect is. The graph seems volatile and hard to decipher what the behavior is. This likely has a large relation with network performance. In the beginning it is seen that performance increases with more worker threads but there is likely a limit to how well performance can increase over the given network connection. This is probably why it's not possible to see an increase in performance with more worker threads since the network connection was the limiting factor.



When analyzing the buffer capacity vs time there is a performance increase when increasing the buffer capacity. However, as buffer capacity got bigger the server and client began running into strange errors of connection. This likely has to do with the network of communication, an external factor outside of control.

Insight

Much of the findings of this project was very interesting. Performance wise, network connection has a large impact on the programs data transfer, and it can be hard to monitor since network connection can be volatile depending on loads from other machines. However, this project was a lot of fun. During my personal experiments with this server and client I attempted to transfer some of my own files such as some pictures and word documents and found that they were successfully transferred without corruption. This was a great moment of success since this is a very practical tool in our everyday lives. I'm grateful for this project and will likely implement it in future personal projects.

Demo Video

https://youtu.be/tsI7Snq5aJ0

GitHub

https://github.com/n-trujillo/Client-Server-TCP-Network-Communication