PA3: A Client Program Speaking to a Server

Christopher McGregor – 726009537

I. Design

Implementation for this PA was very minimal, but I will briefly describe the function's implemented for each task.

A. Requesting Data Points

The function creates a data request, writes it to the channel via cwrite() and receives the server's response via cread(). We output this for the user's convenience

B. Requesting Files

We first send a request with parameters (0,0) to let the server know we want the file size. We use the size to request pieces of the file in increments of the buffer size.

C. Requesting a New Channel

We create a request with new channel type and write it to the control channel. The response of the server is the new channels name, which we then use to create the channel on the client side.

D. Closing Channels

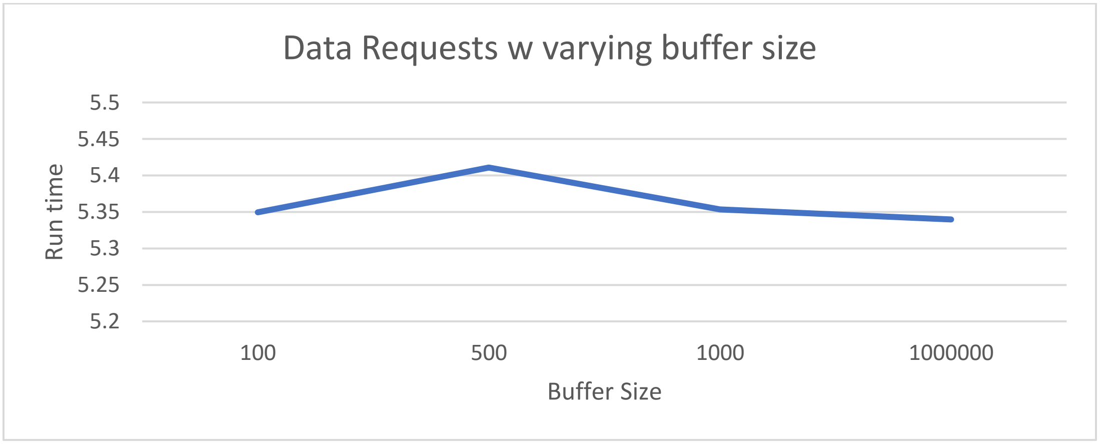
We create a request with quit type and write it to each channel we have created. It is worth noting here that we must explicitly quit the created 'minion' channels as well as the control.

II. Timing Data

Timing data is collected for sending 2000 data request with varying buffer size, requesting various file sizes with a constant buffer size, and lastly requesting one file size with varying buffer sizes.

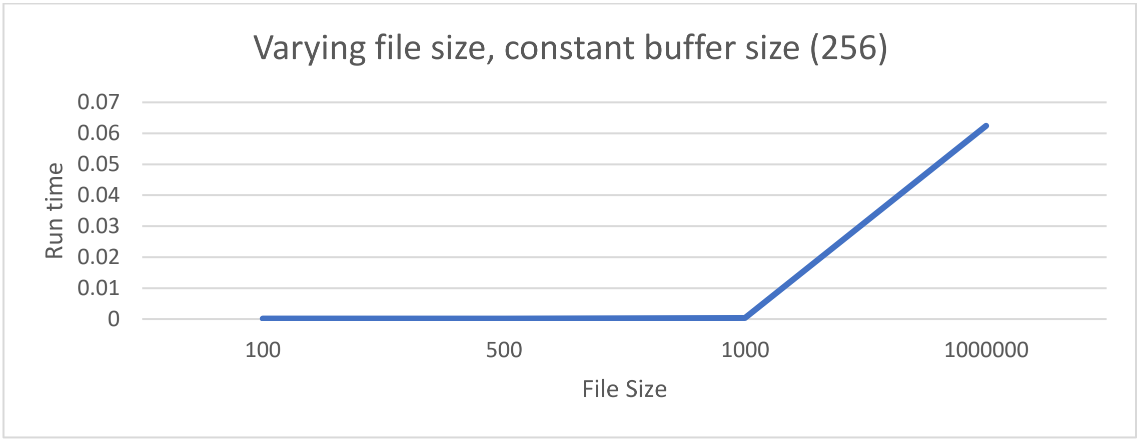
A. Data Requests

We notice that buffer size has zero if not a potentially negative impact on the data request time. This is because the data request is not broken up into pieces, so a larger buffer does not allow more data to be sent over a single piece. We could see a potentially worse run time with larger buffers since we must read the length of the entire length of the buffer regardless of it being filled with nothing.



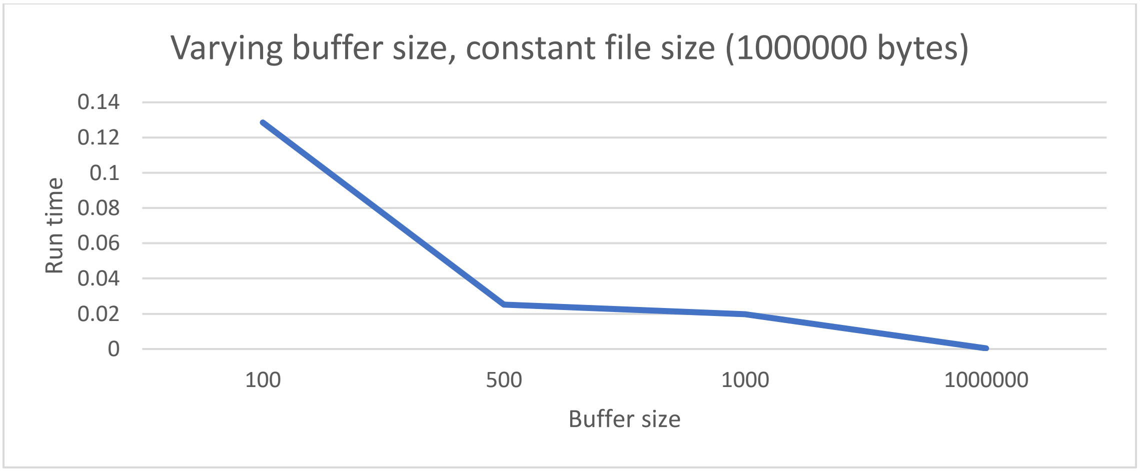
B. File Requests – vary file size

We notice that with increasing file sizes, we see an increased run time. This is because we must send more requests to get the entire content of the file. For instance, if we have a file of size 100 and buffer of size 10, we send 10 requests. Now if the file is 1000 with the same buffer of size 10, we now need 100 requests.



C. File Requests – vary buffer size

We notice that with increased buffer size, we see a decreased run time. This is because we need fewer total requests since the increased buffer size allows more returned data. Using the previous example, a file with size 100 and buffer 10 needs 10 requests. However, if we have the same files size 100 and now a buffer of size 100, we only need 1 request.


Varying buffer size, constant file size (1000000 bytes)
Run time
Buffer size