Clayton Kristiansen

10/22/2021

Programming Assignment 3 Report

Design:

The design of this program was primarily done by the professor. However, it was my responsibility to implement key features such as argument parsing, large data point collection, and file transfer. Command line parsing was done with the getopt() function using this format string: “cd:p:t:e:m:”. the : after each flag character means it requires a value after it in the command. The time to perform each operation was measured using the high_resolution_clock class as it is highly accurate. I chose not to use the gettimeofday() function because of issues with my architecture. This same method was used to determine the runtime for all client/server operations each time ./client was run.

File transfers were done simply by requesting a certain buffer size number of bytes from the server at a specific offset, writing those bits to a file, and incrementing the offset by the number of bytes read. This was repeated until the entire file was read. For the very last request, the request amount had to be made less to be exactly the amount left for the last request. This was done by getting the entire file size before the transfer, then using it as a tool to determine the size of the last request. The time for this to run was also measured.

New channels were requested by using the NEWCHAN_REQ_TYPE request and all operations after a new channel was created were run using the new channel not the old one.

All channels were closed using the QUIT_REQ_TYPE when all client/server interactions were over.

Setting a custom buffercapacity was done using the flag -m with the value being set to the buffercapacity integer object which determined the buffer capacity for the rest of the program.

There were no discernable timing differences between similar sized text files and binary files.

The charts on the next page show the timing data for different sized files, as well as timing data for different buffercapacity values.

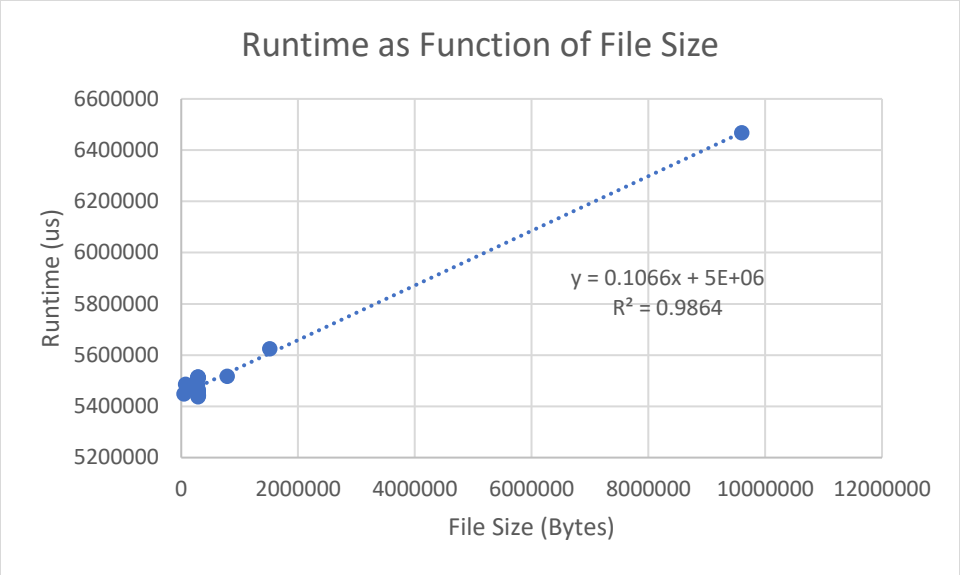

Figure 1 The buffercapacity used in this test was 256 Bytes

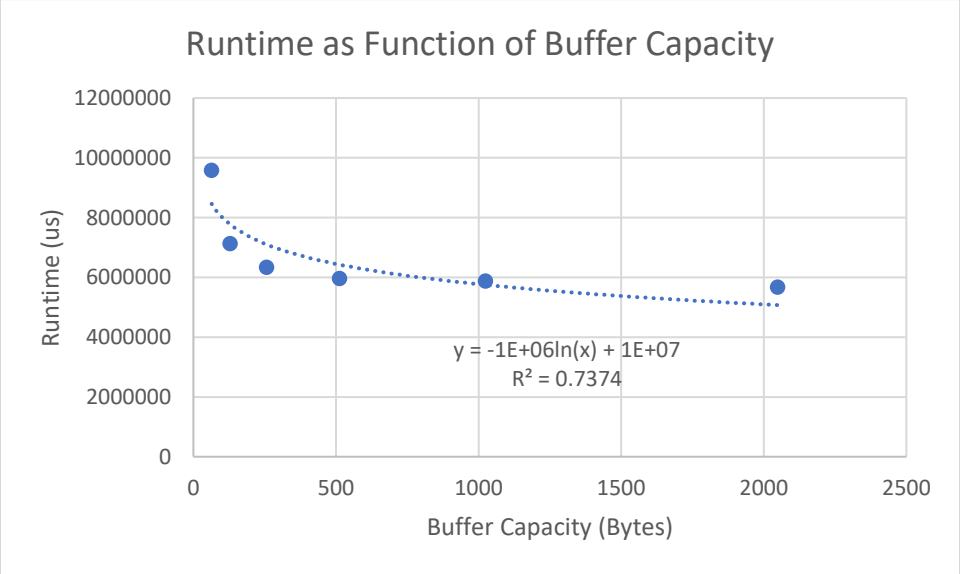

Figure 2 The file size used in this test was 9596614 Bytes