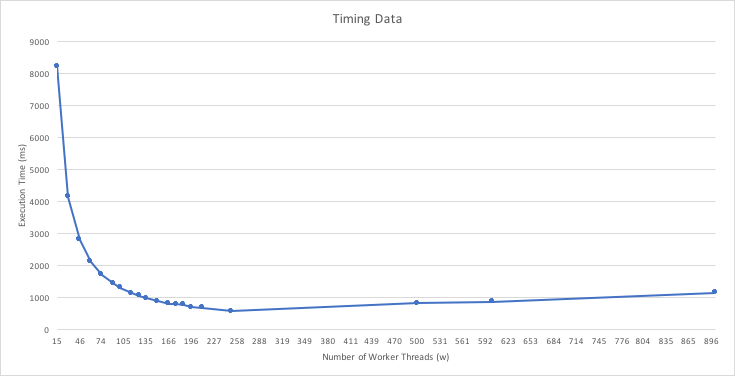
Pa3 report

1. The initial code was using only single thread, which the process waits until all the task for a one thread completing their task and do the next task. Therefore, it’s running time grows very fast as the number of task for each increased, so my task for this assignment was to change this into multithreading which does not wait until the other thread complete their tasks. To accomplish this, I had to put all the request for each thread into SafeBuffer using request\_thread\_functionfunction first not directly request into server and created working threads as much as the user give input for w and using worker\_thread\_function. These functions have to be locked first so that same thread cannot access this function again while it process it,do the tasks, and unlock to solve race condtion. In the worker\_thread\_function it pop the safebuffer and check if the return value is quit it delete the channel else it do the job using cwrite and cread and update histogram based on the request and response.



1. Info Device that it run

MacBook Pro (Retina, 13-inch, Early 2015)

Processor 2.7 GHz Intel Core i5

Memory 8 GM 1867 MHz DDR3

What is the maximum number of threads that the host machine will allow your client program to create before reporting an error? What error is reported?

The initial maximum number of threads that this machine allow is 125. However I can set maximum number of threads using ulimit -n 2000 command. When user give input more than max number of thread, it handle error by printing too many file is opened