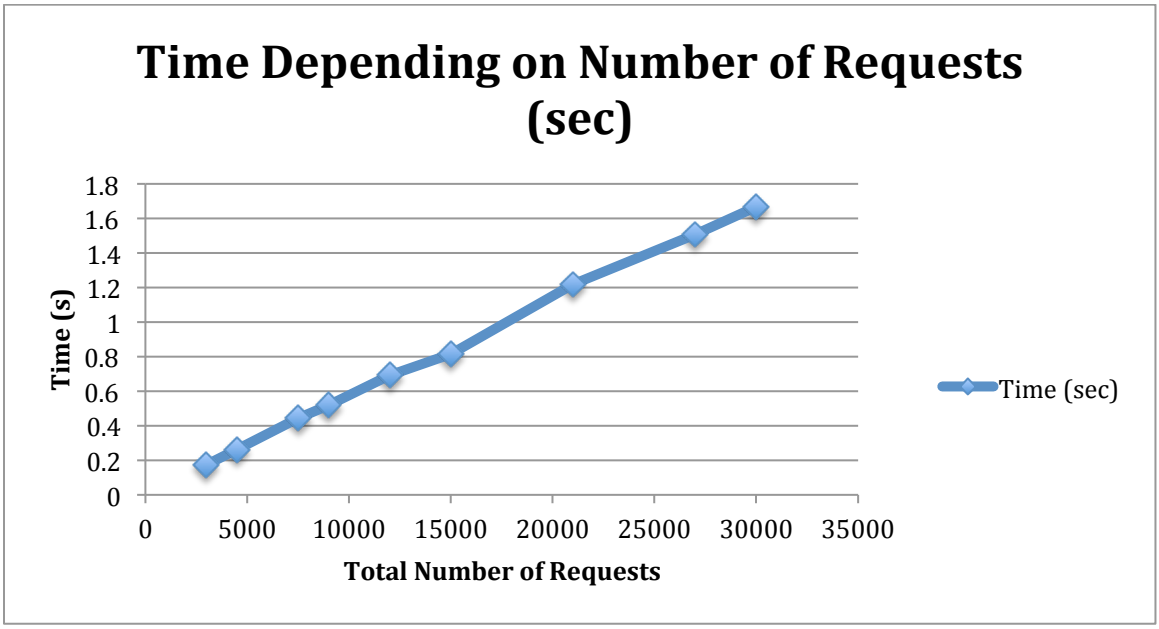
Jonathan Westerfield
Bettati
224005649

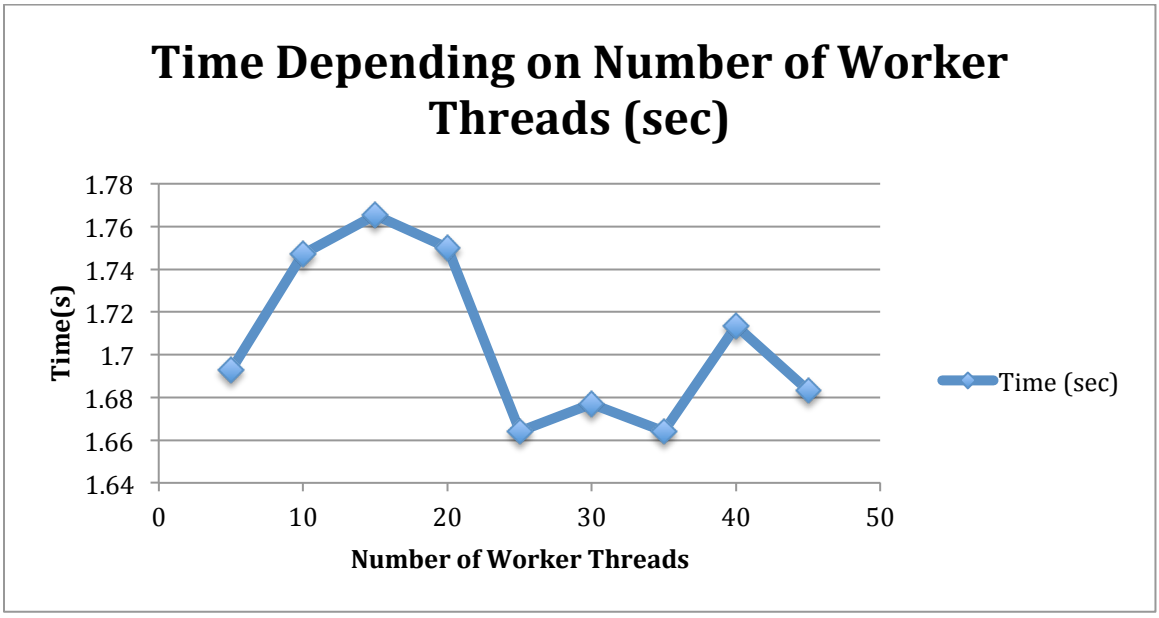
Machine Problem 3 Analysis

I ran all of my tests on my machine. When changing the number of requests, I got the results shown below. The test was performed with 15 request channels and a bounded request buffer of size 500. As you can see, as the number of requests are increased, the time it takes to complete all of the requests (provided everything else stays the same), will increase in a linear fashion.

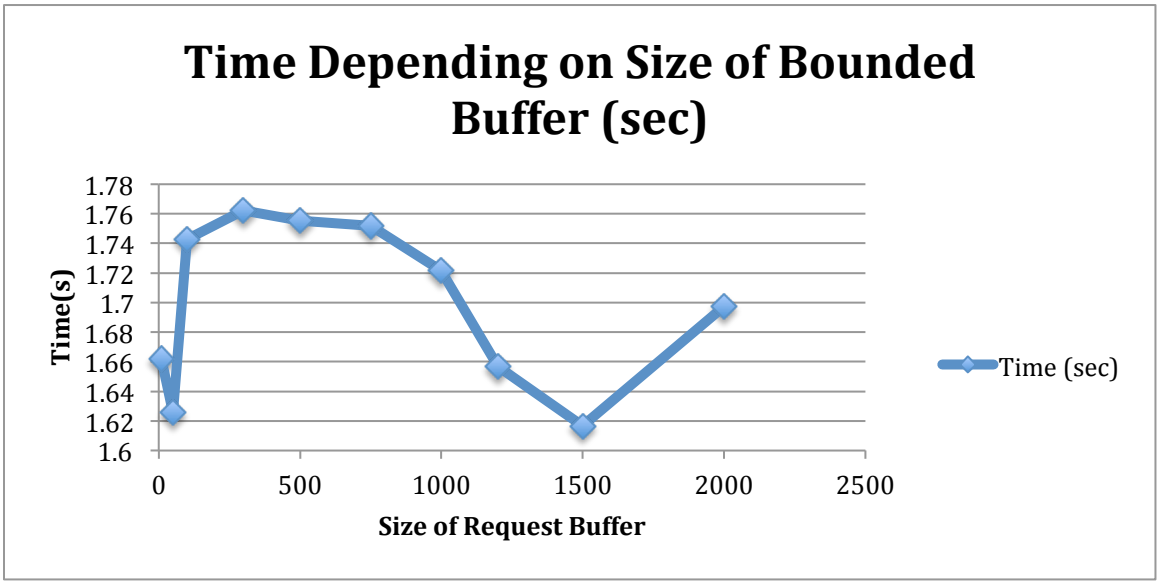


The next set of tests focused on the time changes whenever changing the number of request channels present. The tests were performed with 30000 requests (1000 per request thread) and a bounded request buffer of size 500.When changing the number of request channels. As you can see from the graph below, increasing the number of request channels did increase performance of the program to a certain extent. However, because the graph has no obvious trend, it is hard to say whether the improvements came from the increase in request channels or if there are other underlying factors at play within the operating system. Increasing the request channels to 30 did make significant improvements but more of them started to increase the runtime.

Jonathan Westerfield
Bettati
224005649



The next set of tests dealt with changing the size of the bounded buffer. The tests were performed with 30000 requests (10000 per request thread) and 15 request channels. As you can see from the graph below, increasing the size of the request buffer made improvements once the size was increased past size 1000. However, once it went to size 2000, the increased size didn't make a difference anymore and actually increased the runtime. This means that it is possible that there were other factors at play within the operating system when running multiple processes and pipes.



Jonathan Westerfield
Bettati
224005649

Comparison between MP3 and MP4:

The main difference between the 2 machine problems is the use of processes versus the use of multiple threads on the same process. From my results, using multiple processes and using pipes to communicate between them was a much faster implementation than the use of the threads. The reason for this is that there is very little over head between processes since they are completely separate. For threads, there is some operating system overhead for switching between threads and the threads all sharing the same resources.