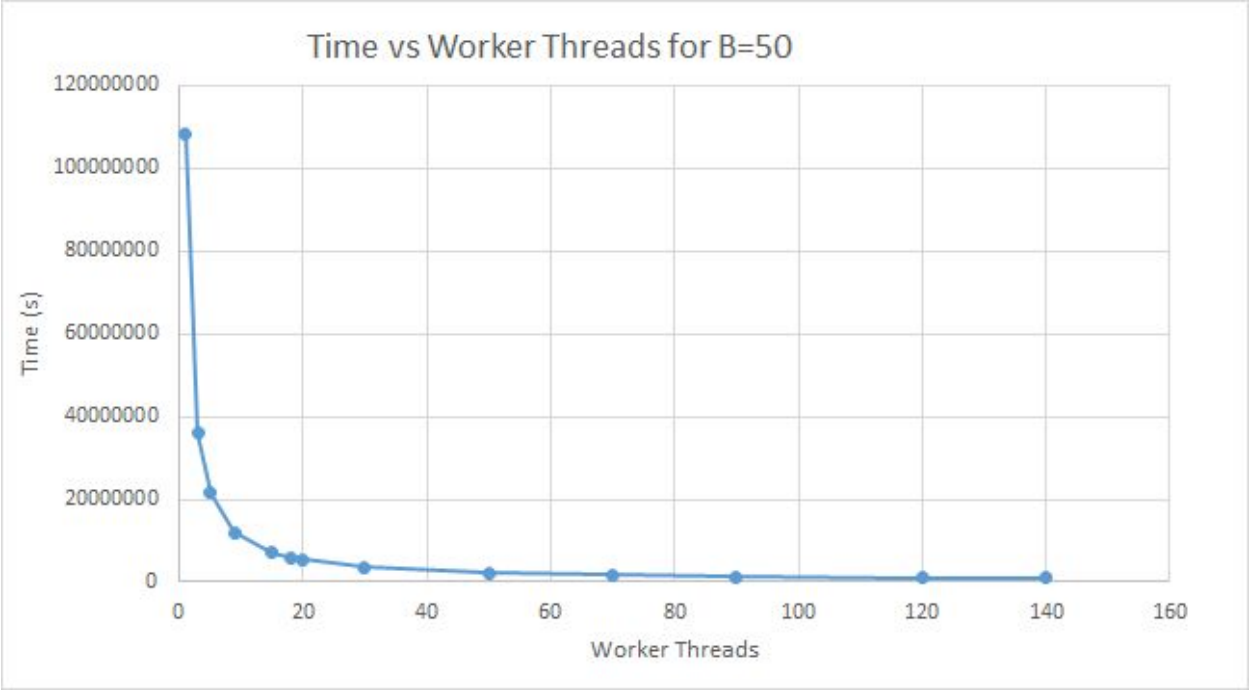
Cesar Ortega, Kerry Zeng
11/21/2016

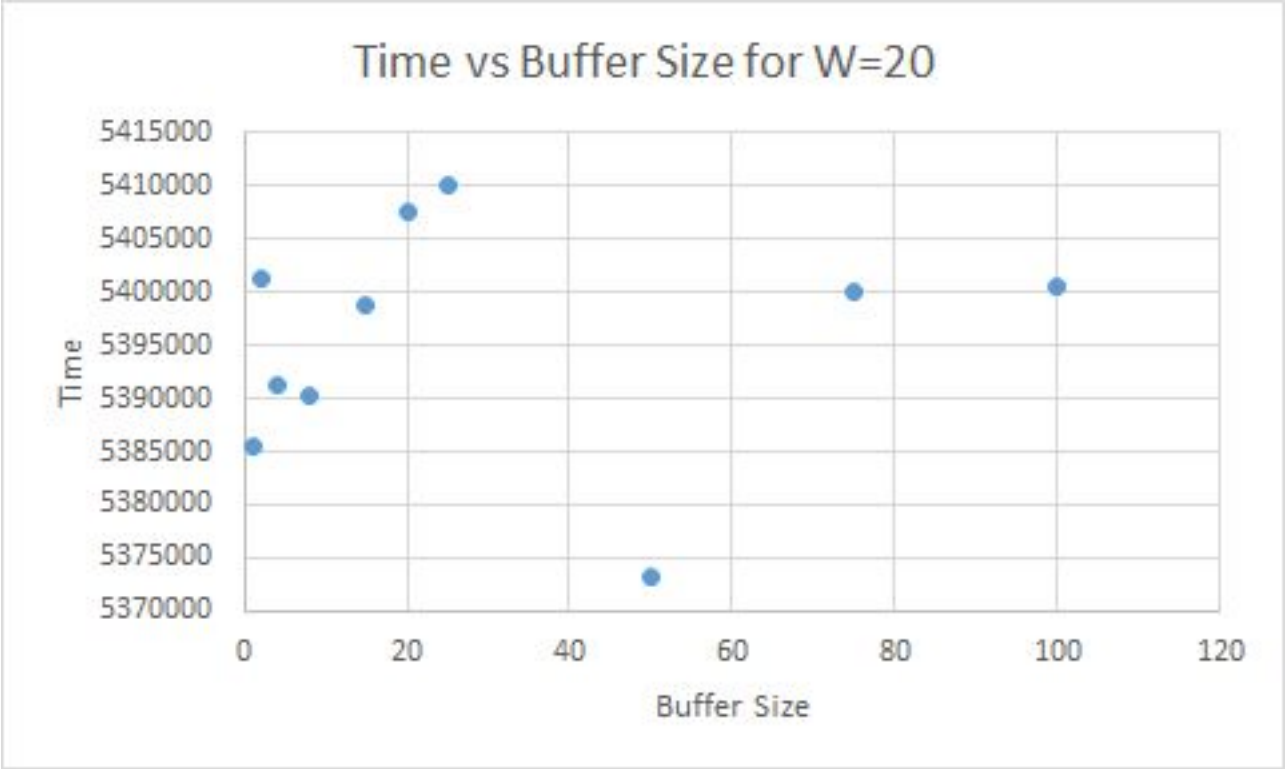
Machine Problem 7

The purpose of this machine problem was to further optimize the client by reducing the amount of threads while keeping the same level of parallelism. To do this, we create *w* channels, all handled through an event handler thread, rather than create *w* threads.

The overall performance of the client for b=50 and varying worker channels was overall better. While not noticeable for the lower number of *w* channel threads, the performance increase can easily be observed at higher *w* values. However the overall shape of the graph is still the same, suffering from diminishing returns after w=30 or so.



For the Time vs Buffer Size graph for W=20, the performance was a little bit better than the previous Machine Problem, but the performance was the same for any buffer size. In the previous Machine Problem, the performance was really bad for buffer size of one but dropped drastically after that.



The likely reason that performance is overall better for for MP7 than MP6 is that the *send_request()* function is generally slow, so handling the channels through the handler using *cwrite()* and *cread()* causes responses and requests to be processed much faster. At the same time, because there is less concurrency overhead (the event handler is only 1 thread accessing data rather than *w* threads), then data can be handled much faster without hang-ups. Both of these things combined let MP7 process requests faster than MP6, leading to a faster overall runtime.