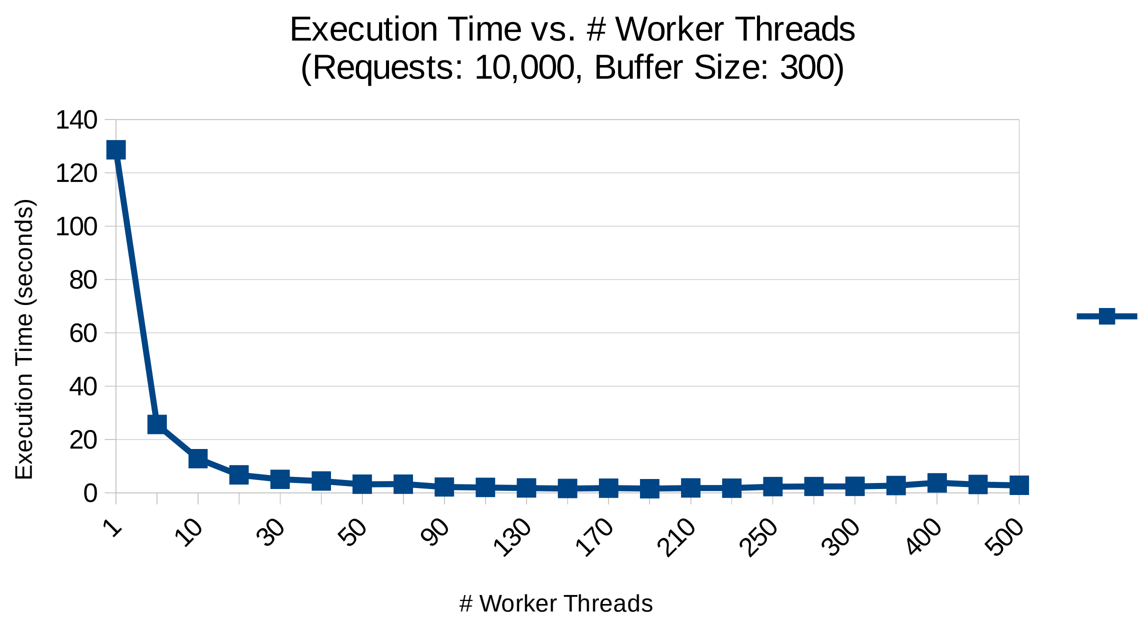
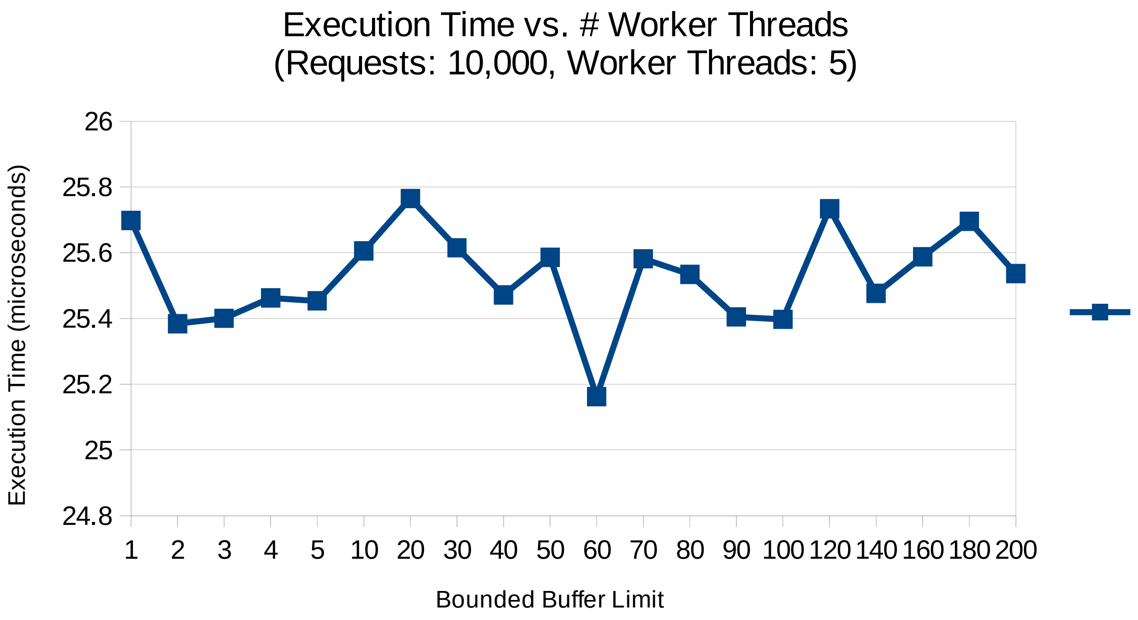
1. Compared to the last implementation of the threading in programming assignment 3, this assignment doesn’t show that much improvement in terms of the efficiency of the code. Despite the fact that all of the threads are running at the same time, there isn’t that much increase in the efficiency of the system. The bounding of the buffer creates some limitation on the speed of the system so, at some points, when the buffer size is low, the system actually works slower than the previous implementation. Despite the fact that the efficiency of the system decreased, the trade-off is necessary especially when memory is limited. If there is a limited amount of memory, the buffer will need to be bounded so that the memory doesn’t fill up and the entire system doesn’t run.

2.





As before, the initial speedup in the execution time when the number of worker threads increases is due to more threads being able to work on the buffers at once. Eventually, the overhead for running all of those threads becomes too much and the execution time slows down when the number of threads is too high. For the execution time in relation to the bounded buffer size, the execution time appears to be sporadic. This is because the execution time of the program does not increase or decrease in a significant way when the size of the bounded buffer increases. Thus, the differences in the execution time in this example is due to the differences in the system at a given time and is not due to the buffer size itself.