Kerry Zeng, Cesar Ortega
Section 503, 501

MP6 Report

The objective of this machine problem was to use semaphores to create a bounded buffer that's able to take requests and process them at the same time. Thanks to this, the performance appears to be better than MP5.

As *w* went up, the performance increased much the same as in MP5, where it was exponential up to a certain point. As the buffer size *b* increases along with *w*, you can see a notable performance increase as the program is able to handle more requests at the same time. However, there comes a point in which performance no longer increases because the program cannot make use of a bigger buffer size, ie. the program has reached the limits of how much buffer size it needs, and adding any more won't increase the speed. According to our data, this limit is two. It makes sense since there are only three threads filling the buffer, but there can be many emptying it. The graphs representing this are on the next page.

Overall, the performance was slightly better than in MP5. However, the lack of large performance increase may be attributed to the fact in order to accommodate the new way of handling the buffer and requests, more overhead was introduced. The program is faster until the point when the overhead is too much to take care of, at which point it mimics the MP5 drop in performance increase.

Data was gathered using the compute.cse.tamu.edu server.

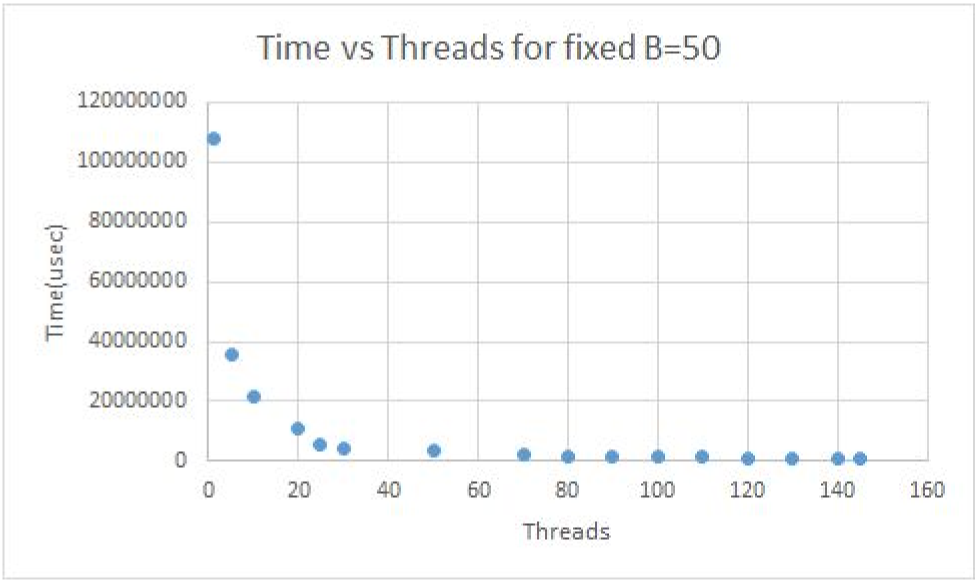

Fig1. Graph for large B with variable W.

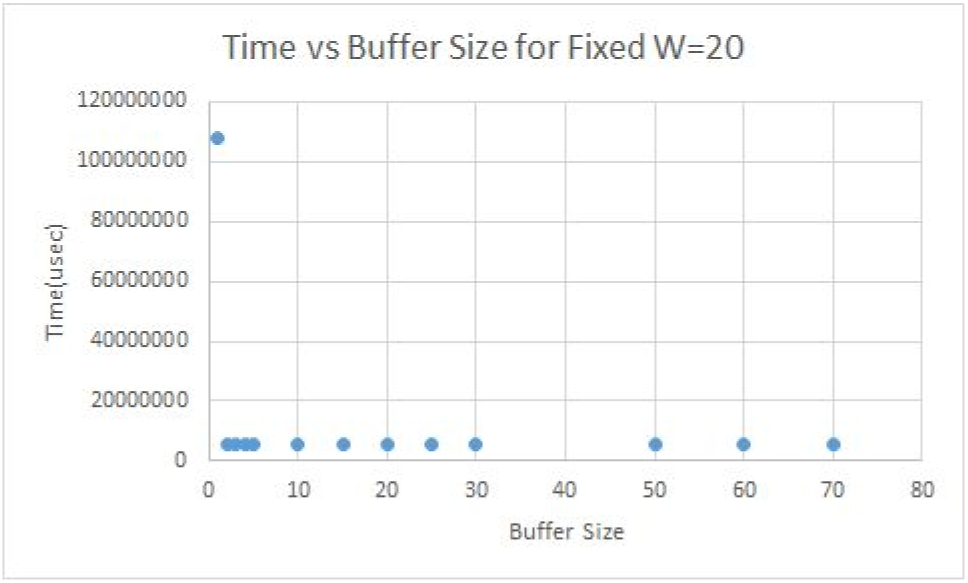

Fig2. Graph for large W with variable B.