Part1: Write Up

The results of part1were interesting because of the performance difference between the three different mutex implementations. Based on the Mutex variance graph that I made, the built in CPP mutex’s throughput is much more efficient than the bakery mutex and the filter mutex.

Enabling 10 threads and comparing our throughput to the C++ mutex, the Bakery mutex performed admirably with about 2/3’s of the performance. Comparing our Filter mutex, it only performed about 1/5 as well. The throughput becomes even more apparent when the number of threads is changed from 10 to 320 with the C++ mutex absolutely crushing the competition. Neither the Bakery nor Filter mutex throughput is 1% of the C++ mutex.

Once the yield is implemented in both our mutex’s, our performance gets even worse. With 10 threads enabled, our Bakery throughput is cut almost in half, yet our filter lock stays roughly the same. Once we kick up the threads to 320, our results are almost as poor as if we never implemented a yield at all.

From these results, we can clearly see that the C++ mutex has a higher performance than the Bakery mutex and the Filter mutex that we were required to make. This is probably because the C++ mutex is built on the compare and swap model.