Milestone Report for Research on Oblivious Algorithms

15-400, Spring 2021

Mike Xu
https://mkpjnx.github.io/ObliviousSort/

March 1, 2021

1 Progress Update

I have done additional analysis on the security requirement and generated some figures to present to Professor Shi tomorrow during our meeting. Although parameters such as the branching factor of the algorithm (γ) and the input size (N) affect the calculations for error rate. The main parameter to be tuned is the "bucket size" (Z), which has the largest impact on the expected overflow rate. We bounded this error rate and found the required bucket size for $\epsilon < 2^{-80}$ to be $Z = 512$.

I also did some investigation on parallel programming frameworks to utilize. I will present a summary of three frameworks (OpenMP, OpenCilk, and Intel TBB) to Professor Shi during our meeting tomorrow. Since I have worked with OpenMP in the past, I have also started working on a proof a concept for the parallel version using the framework. Another reason I chose to use OpenMP is that it requires the least amount of rewriting and is the most compatible. While writing the parallel version, I realized that the lines of code needed to be rewritten was much fewer than expected.

2 Looking Forward

For our purposes, we require the bucket size parameter to be a power of 2. Relaxing this requirement could allow for better performance, though some adjustments must be made to one particular component of our algorithm.

By the next milestone, I hope to have a completed parallel version ready and tested with performance data, so that we can discuss and evaluate it. Professor Shi has also mentioned a number of potential improvements to be explored that were not in the initial milestones. These arose from implementation-specific details, and though they do not improve the asymptotic performance outlined in the paper, they are more practical optimizations for running the algorithm on actual hardware.

Another consideration on the horizon is that of cache complexity. This I anticipate to be difficult, as the theoretical representation of cache complexity outlined in the paper may not fully capture the nuances of an actual machine. Evaluating this on our algorithm will likely require some creative solutions.

3 Milestone Adjustments

We are on schedule with the parallel version implementation, but we do need to compare the performance of our algorithm with bitonic sort.

4 Resources Required

The requirements have not changed much since the last report. We may need use a better computing environment than my laptop for performance evaluation.