CS 575 Project 7b

Jeremy Dao

June 2020

1. The following results were gathered using a machine with a Intel i7-7700k CPU (4 cores, 8 threads) running at 4.20GHz and a Nvidia Quadro 4000 GPU.

2. The following plot show the autocorrelation scatterplot for the provided noise signal.

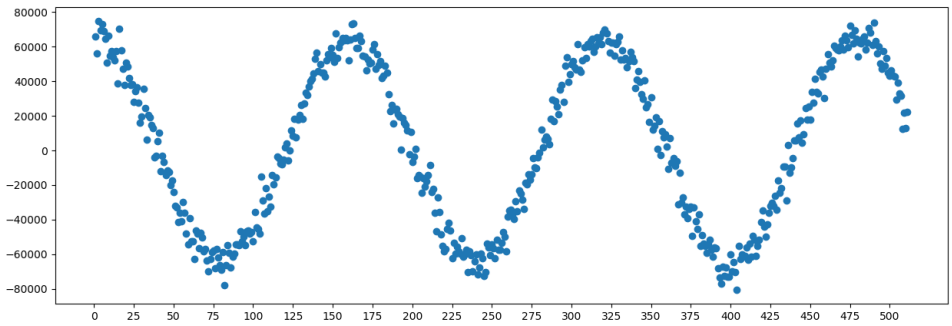

Figure 1: Autocorrelation scatterplot of the noise signal

3. Judging from the autocorrelation scatterplot, it appears that period of the hidden sine-wave is about 160. Every time a shift of multiple 160 occurs, a maximum in the computed autocorrelation occurs.

4. Figure 2 is a bar graph comparing the performance using four different methods: single threaded (computing the autocorrelation sums normally), parallelization across 8 threads using OpenMP, parallelization with SIMD, and parallelization with a GPU using OpenCL.

   Note that the y axis is on a log scale. The most apparent feature of the comparison is that GPU parallelization massively outperforms the other methods, achieving a performance at least an order of magnitude higher. After OpenCL, 8-thread OpenMP performs the next best, followed by SIMD, and finally the baseline of 1-thread non-parallelized performing the slowest.

5. The massive performance gap between OpenCL and all other methods can be explained by the huge difference in GPU and CPU parallelization. GPUs have significantly more compute units that can work in parallel, thus resulting in a huge performance boost over almost any CPU parallelization method (thread parallelism and SIMD included). For example, the Nvidia Quadro 4000 GPU used in these experiments has 256 parllel processing cores, far more separate units than the 8 threads of the CPU that the other methods were run on. For a similar reason, we see 8-thread OpenMP outperform SIMD. Since for SIMD we are using the SSE intrinsics, it runs only 4 things in parallel at the same time, as opposed to 8-threads OpenMP running 8 things in parallel. Obviously, 1-thread OpenMP performed the slowest because it didn't utilize any parallelization.

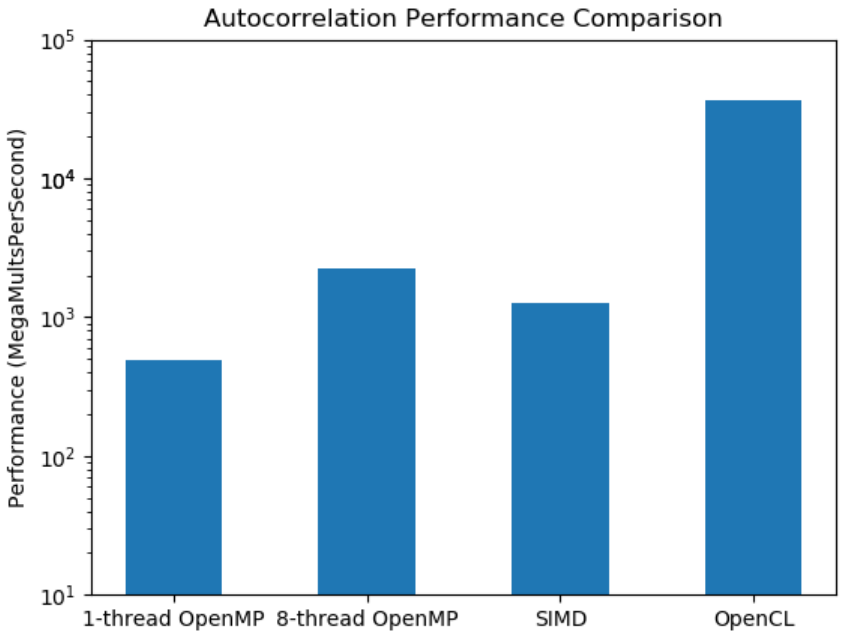

Figure 2: Bar graph comparing the performance across different parallelization methods