Final Project Proposal

Due: 2022/03/09

1 Introduction

In this course's assignments, we have discovered how computationally expensive nanoscale modelling can be. Exactly simulating the interaction of subatomic particles in various configurations requires immense processing power, often outside of the realm of current technological capabilities. Despite the technological limitations, however, there has been decades of research and development in this field. This is largely attributed to clever optimizations and simplifications to various quantum models which have allowed for the highly accurate simulation of nanoscale systems. This project hopes to explore how the current generation of computational resources can be optimized to solve these problems.

Naively programming a scientific workload can lead to numerous inefficiencies if factors such as locality or parallelism are not taken into account. By effectively using a computational resource through programming that takes advantage of proper scheduling and utilization of computing resources, big performance gains can be produced. For example, graphics processing units (GPUs), which were originally designed for real-time rasterized 3D graphics processing possess an architecture that is suited for a variety of other high-throughput arithmetic operations. These devices are found in the majority of high-performance computers but are not utilized unless the software has been explicitly written to make use of them. This software also needs to be written in such a way that the workload is effectively partitioned to stream through the architecture of the GPU in an effective manner. GPUs are now used in a variety of scientific workloads with many nanoscale modelling packages accelerating calculations through them [1].

There also exists the possibility of designing dedicated high-performance hardware to perform singular tasks with high throughput. For example, using field-programmable gate arrays (FPGAs), it is possible to implement in hardware computational blocks designed to accelerate scientific workloads such as those implemented by nanoscale simulations. These same hardware systems could also be designed into custom silicon and act as modelling co-processors. For example, there are several [2][3][5] areas of resarch that explore the implementation of Hartree-Fock and its constituent parts in hardware, which may provide some good insight for this project. FPGAs have the benefit of being more accessible for custom hardware applications at the expense of being more expensive in terms of cost and power. Custom silicon, on the other hand, have very high upfront costs, but are cheaper in mass quantities and can be expected to be more power efficient.

2 Methodology

The abundance of GPUs in today's everyday computing platforms makes them an attractive prospect for this project, so they are chosen as the computational resource to leverage and optimize for. A simulation or modelling algorithm, likely one that was covered in this class, will be identified as the subject for acceleration. Either a baseline simulation or modelling program will be written from scratch, or an existing one that is not using GPU acceleration will be chosen as the project's subject. A common workload will be chosen and benchmarked to be able to compare the performance improvements before and after GPU acceleration. To implement the GPU acceleration, the widely-used CUDA [4] platform and programming model will be used. It is well documented and there is an NVIDIA GPU available for use in this project. Once the performance optimizations have been applied and the data has been gathered, a summary report will be written to summarize the results of the project.

3 Work Plan

The following rough workplan has been laid out to help estimate the amount of work required to get the system up and running producing results. In brackets are the estimated number of weeks that will be required to finish that portion of the task.

3.1 Milestones [Estimated Weeks]

- [0.75] Literature review
- [0.25] Workload acceleration identification
- [0.50] Baseline metrics obtained for comparison
- [1.50] Acceleration implementation
- [0.50] Analysis and report

References

[1] R. C. Walker, A. W. Goetz, and W. O. Library, *Electronic structure calculations on graphics processing units: from quantum chemistry to condensed matter physics.* Chichester, West Sussex, United Kingdom: John Wiley & Sons Inc, 2015;2016;.

[2] W. K. U. Klaassen, “Analysis of high performance scientific programming workflows,” Ph.D. dissertation, 2018.

[3] M. Wielgosz, E. Jamro, and K. Wiatr, *Hardware Implementation of the Exponent Based Computational Core for an Exchange-Correlation Potential Matrix Generation*, ser. Parallel Processing and Applied Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 115–124.

[4] [Online]. Available: https://developer.nvidia.com/cuda-zone

[5] T. Ramdas, G. Egan, D. Abramson, and K. Baldridge, “Towards a special-purpose computer for hartree–fock computations: What’s on the table, and how do we take it?” *Theoretical chemistry accounts*, vol. 120, no. 1-3, pp. 133–153, 2007;2008;.