COURSE PROJECT: TUNING A GPU-IMPLEMENTATION THROUGH PROFILING AND NEW MEMORY ACCESS SCHEMES

JING LIU, CARL NETTELBLAD

Background

This project has its origin in the field of structural biology, where the diffraction patterns arising from X-ray illumination of macroscopic crystals of biological compounds have been used as a method to determine their structure (3D coordinates for each atom) for decades. Newly developed and constructed X-ray free-electron lasers, like the 1 kilometer long Linac Coherent Light Source (LCLS) facility at Stanford University, can provide extremely powerful and extremely short bursts of X-rays. In the duration of a single pulse, the beam focused to a micron-sized spot has the same power density as all the sunlight hitting the Earth, focused to

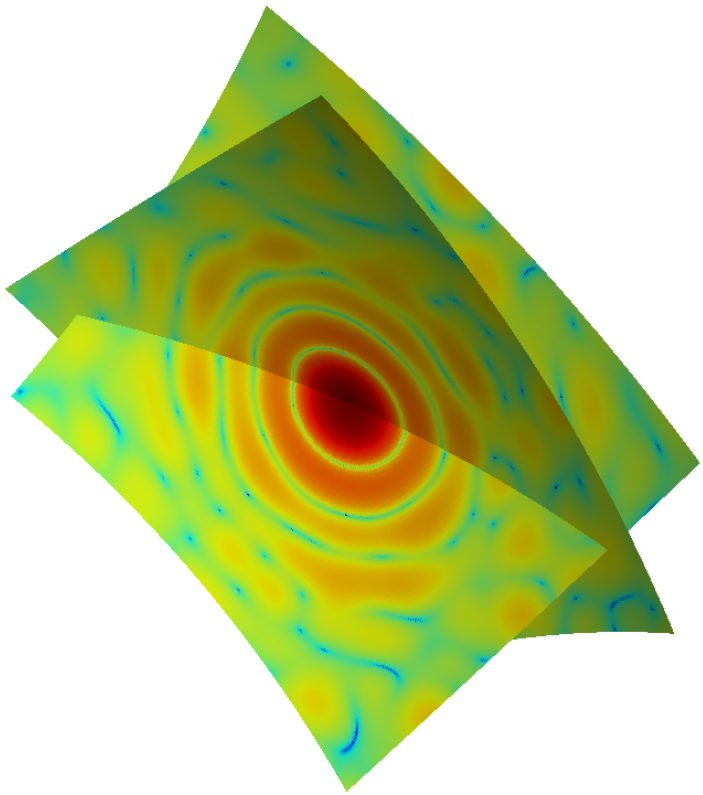

FIGURE 1. Alignment of multiple curved diffraction patterns in a 3D volume, the central problem of the EMC algorithm.

a millimeter square. With an ultra-short and extremely bright coherent X-ray pulse, a single diffraction pattern may be recorded from a *single* protein molecule, a virus particle, or a cell before the sample explodes and turns into a plasma. A femtosecond X-ray pulse can outrun slower damage processes in the sample (there is simply no time for the particle to collapse). Hence, the diffraction pattern comes from a practically undamaged object.

Description

Sample particles are injected individually into the X-ray beam and are intercepted randomly, in unknown orientations, by the femtosecond X-ray pulses. Each diffraction pattern represents a Fourier transform of the density of the particle in a specific orientation. This turns out to correspond to a curved section of a total 3D volume of diffraction data. If multiple particles are identical up to this orientation difference, the total 3D volume can be filled in by aligning all such 2D slices.

Further details

The alignment of 2D diffraction slices can be approached with several methods. We have a GPU implementation of the Expansion expectation Maximization and Contraction algorithm [1]. When analyzing the behavior of this implementation, it is clear that the total computational throughput is only a few percent of what is possible on these cards.

This projects aims to improve upon this. For this purpose, the behavior of the algorithm will be studied using Nvidia profiling tools. In addition, we already have tentative suggestions for what to test. These are outlined below. Depending on the time available, the project group could study some or all of these:

- The Nvidia compiler is able to use additional decoration of pointers such as `const restrict` to make further optimizations and allow asynchronous caching of memory content. Proper decoration should be included throughout the relevant computational kernels and performance compared.
- The "expansion" step in EMC is basically analogous to rasterizing a 3D texture into a 2D map based on a curvature (one per tested orientation for approximately 100,000 orientations). CUDA GPU programming has builtin support for using the texture units in the GPU. This allows better memory access patterns even on architectures where `const restrict` is not used. Furthermore, interpolation operations can be done "for free", rather than being explicitly implemented, which is the case today.
- The point of the "expansion" step is to make further comparisons of patterns against data cheaper and more sequential. However, it also increases load on the memory subsystem dramatically. The "expansion" and "maximization" steps of the algorithm could be merged. This increases the number of floating point operations needed quite significantly, but it might reduce memory bandwidth load enough to be preferable, since the original 3D volume is miniscule compared to the expanded separate 2D orientations.
- Similarly, the contraction/compression step in the EMC algorithm is also done as a two-step process, first creating a new model for each extracted orientation, and then merging those into a consistent 3D model. Since these operations require memory

[1] NTD Loh, V Elser. Reconstruction algorithm for single-particle diffraction imaging experiments. *Phys Rev E* **80**(026705), 2009.

writes (not only reads), coalescing these two steps into one is slightly more tricky, but the same general argument about reducing the count of write operations might hold.

Overall, the goal is to recognize that many current computational problems are bound by memory bandwidth. In addition to general schemes for reducing the total load, blocking algorithms for improving memory locality could be investigated. Each improvement in the list above should be tested on simulated and real datasets of different sizes, and corroborated against different hardware counters and diagnostics available in the profiler.

Aims

By testing the different improvements outlined above, we hope to achieve a performance increase of a factor between 1.5 and 10x. This would still only be equivalent to using up to 20% of the computational power possible for "useful" calculations (total FLOP load will increase more, since improvements are gained by performing some calculations multiple times rather than storing them in memory), but such seemingly low efficiency ratings are quite common in realistic GPU applications, and would still amount to approximately a 100x improvement compared to a regular CPU-based implementation.

Contact

We are looking for motivated and interested students to take on this challenge. Experience in C/C++ programming using CUDA is highly beneficial, so is low-level profiling in CUDA or on CPU architectures. The actual implementation uses mainly plain CUDA with some Thrust.

We are continuously looking for new ways to attack the computational challenges in this nascent area. Further possibilities for e.g. master thesis and PhD student projects might exist, although those would probably include more of a balance between purely computational and data analysis aspects of the problem.

(J. Liu) Division of Scientific Computing, Department of Information Technology, Uppsala University, Sweden.
E-mail address: jing.liu@it.uu.se

(C. Nettelblad) Laboratory of Molecular Biophysics, Department of Cell and Molecular Biology, Uppsala University, Sweden.
E-mail address: carl.nettelblad@icm.uu.se