Parallelizing Protein Docking Code to Accelerate Drug Discovery

Brad Burkman[1], Michal Brylinski[2,3], Wei Feinstein[2,3]

[1]Louisiana School for Math, Science, and the Arts, [2]Center for Computation & Technology, Louisiana State University,
[3]Department of Biological Sciences, Louisiana State University

LA-SiGMA
Louisiana Alliance for Simulation-Guided Materials Applications

Abstract

The majority of drugs work by binding to specific regions of pharmacologically relevant proteins (enzymes, receptors, regulatory proteins, etc.) to attenuate or even impair their molecular functions. Because protein-protein interaction sites are attractive targets for therapeutics, the accurate modeling of protein assemblies at the atomic level is critical in modern computer-aided drug discovery.

This work took an academic version of ZDOCK, one of the most widely used protein docking algorithms, and attempted to speed up the code through parallelization on multiple CPU cores, as well as using MIC and GPU accelerators. In anticipation of LSU's new SuperMIC cluster becoming operational, this study used the Stampede cluster, an XSEDE resource at the Texas Advanced Computing Center.

Thus far, we have achieved a ten-fold speedup using just the sixteen cores on the CPU and a four-fold speedup by offloading Fast Fourier Transforms to the GPU. Our work with the MIC continues. By combining and refining these techniques, we reasonably hope for a twenty-fold speedup in this key step in drug discovery.

Glossary

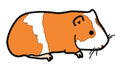
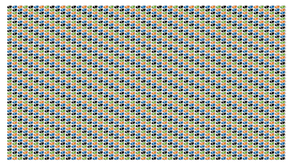
CPU	Central Processing Unit ("host")
MIC	Intel Many Integrated Core coprocessor ("target")
GPU	Graphics Processing Unit ("device")
Serial	Running one process at a time
Parallel	Running multiple processes simultaneously
FFT	Fast Fourier Transform

Serial Code Profile

At the heart of the code is a loop of thousands of independent processes, each requiring several Fast Fourier Transforms.

Section of Code	Runtime (Seconds)	Runtime (Proportion)
Before Loop	1.94	0.57 %
Loop	340.38	**99.32** %
After Loop	0.39	0.11 %

Parallelizing for the CPU, MIC, and GPU

CPU	MIC	GPU
		
Intel Xeon E5 16 cores 21.6 GFLOPS/core 32 GB	Intel Xeon Phi 60 cores 16.2 GFLOPS/core 8 GB	NVIDIA K20 2496 cores 1.4 GFLOPS/core 5 GB
		

Results

Protein Size	Serial Code (Seconds)	Parallel on CPU (Seconds)	Speedup on CPU	Parallel on GPU* (Seconds)	Speedup on GPU*
Small	319	33	**9.5x**	110	**2.9x**
Medium	1410	112	**12.5x**	342	**4.1x**
Large	14000	1230	**11.4x**	2800	**5.0x**

*Preliminary results. GPU code runs but does not yet give correct results.

Parallelization Strategies

CPU: OpenMP on big loop

GPU
Copy protein to device
Repeat $\approx 25,000$ times:
- **On Host**
  - Rotate ligand
  - Copy ligand to device
    - **On Device**
      - Forward FFT
      - Complex Multiplication
      - Inverse FFT
  - Copy ligand to host
  - Calculate scoring on host

Parallelization Challenges

GPU
- Finding helpful, current documentation
- `cufft` and `fftw` have diffferent syntax and options
- CUDA code gives different results (still debugging)

MIC
- Automatic offloading of Fast Fourier Transforms does not yet exist, and may not be possible
- Passing `fftw` structs to the target

Future Work

Easy
- In GPU code, do the scoring on the device instead of copying the ligand to the host
- Flatten `fftw` structs to send to MIC to accelerate FFT computation
- Anticipate developments in Intel FFT libraries for the MIC

Hard
- Fully utilize the host and device simultaneously
- Use multiple nodes

Acknowledgements

This material is based upon work in the LA-SiGMA Research Experiences for Teachers program, supported by the National Science Foundation under the NSF EPSCoR Cooperative Agreement No. EPS-1003897 with additional support from the Louisiana Board of Regents.

This work used the computing resources of the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575

Contact

Brad Burkman `bburkman@lsmsa.edu`
Lecturer in Mathematics,
Student Research Advisor
XSEDE Campus Champion
Louisiana School for Math, Sci., and the Arts

Louisiana School for Math, Science, and the Arts

LSU Center for Computation & Technology

NSF XSEDE
Extreme Science and Engineering Discovery Environment