Scientific Computing with GPUs

Graphics processing units (GPUs) aren’t just for graphics anymore. These high-performance, many-core processors are routinely used to accelerate a wide range of science and engineering applications, in many cases delivering dramatically improved performance. GPU technology originated from the need to improve the performance of computer games and graphics applications, and for a long time it wasn’t widely accessible by the scientific computing community. However, with the introduction of the Compute Unified Device Architecture (CUDA) developed by Nvidia and the C for CUDA language extensions and compiler, the power of GPUs was unlocked for general-purpose use. AMD also quickly followed suit with the introduction of its line of GPUs and support for Open Computing Language (OpenCL). As a result, today we have two viable and competitive product lines, Nvidia and Advanced Micro Devices (AMD) GPUs, with support for a wide range of programming languages.

Computer architects also use GPUs to build the world’s largest supercomputers. According to the November 2011 release of the TOP500 list of supercomputer sites (see www.top500.org), three out of the five top supercomputers are GPU-based:

- Tianhe-1A, deployed at the National Supercomputing Center in Tianjin with 4.7 petaflops (Pflops) peak/2.5 Pflops sustained performance;
- Nebulae-Dawning, deployed at the National Supercomputing Centre in Shenzhen with 2.9 Pflops peak/1.2 Pflops sustained performance; and
- TSUBAME 2.0, deployed at the Tokyo Institute of Technology with 2.2 Pflops peak/1.2 Pflops sustained performance.
In the US, two major GPU-based deployments are underway: the 20-Pflops Titan at the Oak Ridge National Laboratory and 10-Pflops Blue Waters at the National Center for Supercomputing Applications. A 10-Pflops GPU-based system is also planned to be deployed at Moscow State University in Russia.

However, using GPUs in scientific computing comes with an added risk: application-porting efforts could be substantial, and not every application benefits from GPU acceleration equally well. While early successes in porting many computational kernels to GPUs demonstrated the potential of this technology for scientific computing, only now such efforts are starting to deliver production-grade scientific codes that truly benefit from GPUs.

**In this Issue**

This special issue attests to the widespread use of GPUs in the scientific computing community. The five selected articles cover a wide range of scientific applications, from DNA docking to earthquake simulations.

In “Computational Fluid Dynamics Simulations Using Many Graphics Processors,” Ali Khajeh-Saeed and J. Blair Perot show how unsteady, incompressible computational fluid dynamics (CFD) simulations of turbulence are performed using up to 64 GPUs. The researchers optimized the CFD-GPU algorithm by using the GPU’s shared memory and overlapping communication with computation. As a result, GPU-based calculations became more efficient and the operations related to data exchange between the compute nodes began causing poor performance (a scaling bottleneck) on all but the largest problem sizes.

The next article, “A GPU-Based Approach to Accelerate Computational Protein-DNA Docking,” by Jiadong Wu and his colleagues, investigates strategies to accelerate conformational search algorithms for the protein–DNA docking problem using GPUs. The authors integrate algorithmic techniques to optimize the computation on a GPU-based cluster. The newly developed GPU-accelerated docking algorithm achieves 10.4 Tflops of sustained performance using 128 Tesla M2070 GPUs, which represents a $3.6 \times$ speedup over a traditional cluster of 1,000 CPU cores.

Rio Yokota and Lorena Barba consider a GPU implementation of algorithms designed to solve a classical many-body problem in their article, “Hierarchical N-Body Simulations with Auto-tuning for Heterogeneous Systems.” The authors propose a new hybrid treecode and fast multipole method with the capability to autotune the kernels on heterogenous architectures. They demonstrate the method’s performance on a GPU architecture using a Laplace potentials kernel that’s at the core of many astrophysics and molecular dynamics simulations.

In the article “Three Applications of GPU Computing in Neuroscience,” Javier Baladron and his colleagues demonstrate the use of GPUs in the domain of theoretical neuroscience. They describe the GPU-accelerated continuous model of the primary visual area, the simulation of a stochastic neural network, and the computation of the probability distribution on the possible states of a network while also demonstrating an improved performance of these three applications.

Last but not least, in “Accelerating a 3D Finite-Difference Earthquake Simulation with a C-to-CUDA Translator,” Didem Unat and his colleagues describe their experience in porting a real-world earthquake code to GPUs. Their approach is based on using an annotation-based programming model and source-to-source translation for automatically generating GPU code. This approach produces GPU code that performs comparably to handwritten code, yet at a fraction of the development time.

GPU technology is revolutionizing the way we think about parallel computing and the way we write parallel applications. With numerous scientific and engineering codes being ported to GPUs, the scientific computing community is awaiting the introduction of Nvidia’s next-generation graphics processor architecture, code-named Kepler. Kepler is expected to bring considerable performance improvements and flexibility in programming, which should speed up the development of even more applications.

**Acknowledgments**

I would like to thank the reviewers, whose thoughtful input was essential in selecting these articles. Their feedback made the authors’ contributions even better.

Volodymyr Kindratenko is a senior research scientist at the National Center for Supercomputing Applications and a lecturer in the Department of Electrical and Computer Engineering at the University of Illinois in Champaign-Urbana. His research interests include high-performance computing and special-purpose computing architectures. Kindratenko received a DSc in analytical chemistry from the University of Antwerp. He’s a senior member of IEEE and the ACM. Contact him at kindr@ncsa.illinois.edu.