Blasting Through Lattice Calculations using CUDA
Kipton Barros, Ronal Babich, Richard Brower, Michael A. Clark, and Claudio Rebbi
Boston University

ABSTRACT
Modern graphics hardware is well suited to high parallelism and provides high performance benefits. Graphics hardware vendors are now making available development tools to support high performance computing. NVIDIA’s CUDA platform, in particular, offers direct access to graphics hardware, and is available in CUDA C, which uses an extension of the C programming language. Using the CUDA platform we have implemented a Dirac-Wilson operator which runs at an efficiency 60 GigaFlops on the Tesla C850 GPU. The recently released GTX 280 GPU runs this same code at 90 GigaFlops and we expect improvement pending code optimization.

1 Introduction
Some’s lore, the doubling of computer power every year and a half, can be traced to exponential increases in the density and performance of processor-based multiprocessors. Multiprocessor– and/or multiple cores–with very high memory bandwidth is a key limit to achieving outstanding performance on fine-grained massively parallel applications. Multiple cores and very high memory bandwidth are the keys to the path to exascale computing.

2 The NVIDIA HPC Graphics Cards
• NVIDIA’s Tesla line of graphics cards
  - NVIDIA’s Tesla line of graphics cards is designed specifically for high performance computing using the Compute Unified Device Architecture (CUDA) software architecture for issuing and managing computations on the GPU. CUDA uses the same C programming language as a higher level code generator [5].

3 The CUDA Programming Model
• CUDA (Compute Unified Device Architecture) is a new hardware and software architecture for running and managing computations on the GPU. It is a data-parallel computing device from NVIDIA’s CUDA Programming Platform.

4 Wilson Inverter: Main features
• The power of GPUs has already been harnessed to speed up Lattice Calculations (LCT) [4], but the task is made much easier with CUDA, which also helps to write more efficient code.

5 Wilson Inverter: Example Code

6 Performance
• Both the en-vogue preconditioned Dirac-Wilson and the FFTW inverter have been benchmarked for a variety of different volumes. The CG1 label in the plot signifies that the gauge-fixing trick was used, in this case the GigaFlop/s number reported in the efficiency figures is only weakly volume dependent. For all but the smallest volumes the performance remains above 60 GigaFlops on the C870 and around 90 GigaFlops on the GTX 280. The gauge-fixing trick employed to reduce the bandwidth results in over 50 GigaFlops and 80GigaFlops on the C870 and GTX 280 respectively. In addition the scaling with increasing volume is relatively constant, compared to GPU implementations which drastically fall in performance as the local volume is increased out of the cache.

References
[1] See for example http://www.nvidia.com/object/tesla.html
[2] "In CUDA, CUDA Programming Guide", http://developer.download.nvidia.com/compute/cuda/3.2/NVIDIA_CUDA_Programming_Guide.pdf
[3] Cliff Jones, "NVIDIA CUDA Tutorial CUDA", http://people.maths.ox.ac.uk/cjones/pubs/NVIDIA_NVIDIA_CUDA_Tutorial_NVIDIA.pdf
[4] "Lattice QCD as a game", Comput. Phys. Commun. 171, 631 (2007) http://arxiv.org/abs/0704.3822.0704.3822
[5] "The CUDA Programming Language", http://www.nvidia.com/object/developer_nvidia.html

Achievements
This work was supported in part by US DOE grants DE-EE02-06ER46458 and NSF grants 0223682, 0704554, and OCE 1037263.