CBench: Analyzing Compute Performance for Modern NVIDIA and AMD GPUs

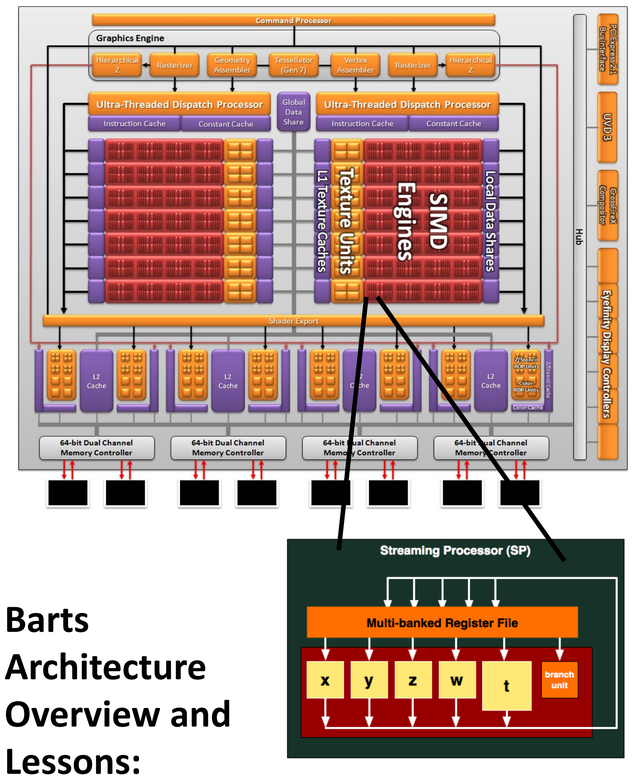
Varun Sampath

CIS 565: GPU Programming and Architecture

Abstract

General purpose GPU computation is a fast growing field with a variety of applications. For maximum performance, though, mapping high-level parallel algorithms to vendor hardware requires a solid grasp of both the algorithm's computational requirements and the microarchitectural limitations of the GPU. This work aims to explore the performance of high and low arithmetic intensity workloads on the latest NVIDIA and AMD GPU hardware, codenamed Fermi and Barts, respectively. A summed area table generator and a Black-Scholes option pricer were used as benchmarks to analyze performance for compute- and bandwidth-bound algorithms. It was found that the AMD Barts GPU provided a 50% performance boost on the Black-Scholes compute-bound workload, whereas Fermi excelled at the more memory-bound summed area table computation.

AMD Radeon HD 6870 (Barts)



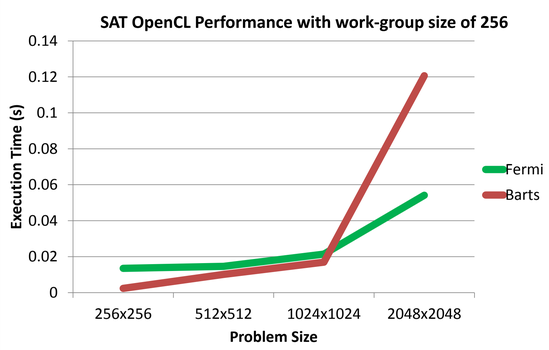
Barts Architecture Overview and Lessons:

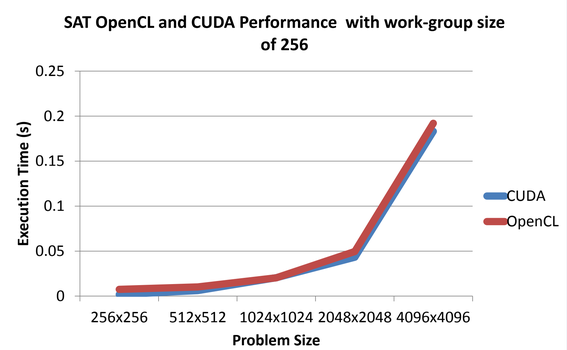
- Barts contains 14 compute units and 16 stream cores per compute unit
- Each stream core contains 5 scalar processing units
- Each stream core executes a VLIW instruction bundle targeting these processing units
  - Black-Scholes saturates 4 or 5 units for a majority of the ALU instruction bundles
  - SAT leaves most idle
- VLIW enables 2016 GFLOPS peak performance for Barts
- Groups of 64 VLIW instructions with the same clause type are executed in SIMT bundles called wavefronts
- Global memory accesses can suffer from both channel and bank conflicts
- Local/Shared memory access can suffer from bank conflicts and bandwidth issues trying to satisfy VLIW processing units

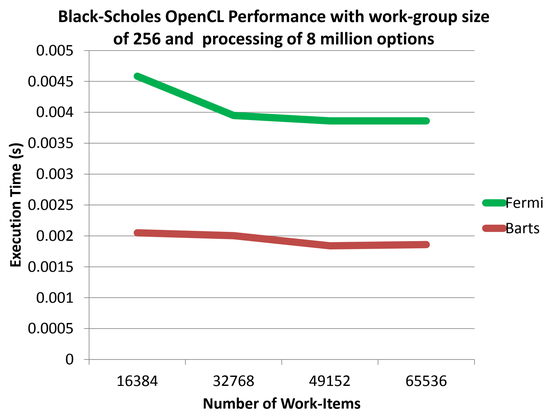
Method

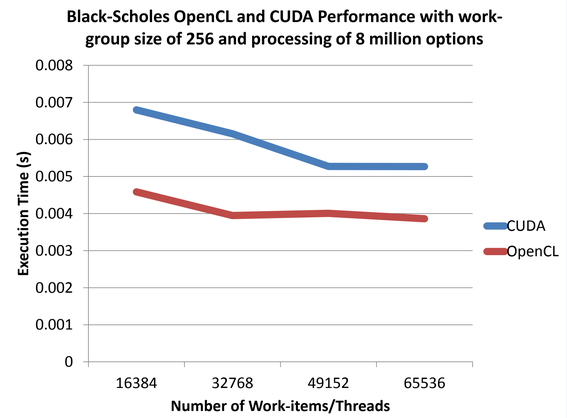
1. Develop a benchmark with low compute/memory access ratio
   - Summed Area Tables
     - inclusive scan and transpose operations on off-chip global memory
2. Develop a benchmark with high compute/memory access ratio
   - Black-Scholes option pricing
     - embarrassingly parallel streaming computation
     - Almost all single-precision floating point operations
3. Execute on both CUDA and OpenCL platforms

Results









References

ADVANCED MICRO DEVICES, INC. 2011. AMD Accelerated Parallel Processing OpenCL, Apr.

MARK HARRIS AND SHUBHABRATA SENGUPTA AND JOHN D. OWENS. Parallel Prefix Sum (Scan) with CUDA, vol. 3 of GPU Gems.

NVIDIA CORPORATION. NVIDIA's Next Generation CUDA Compute Architecture: Fermi.

NVIDIA CORPORATION. 2010. NVIDIA CUDA C Programming Guide, Oct.

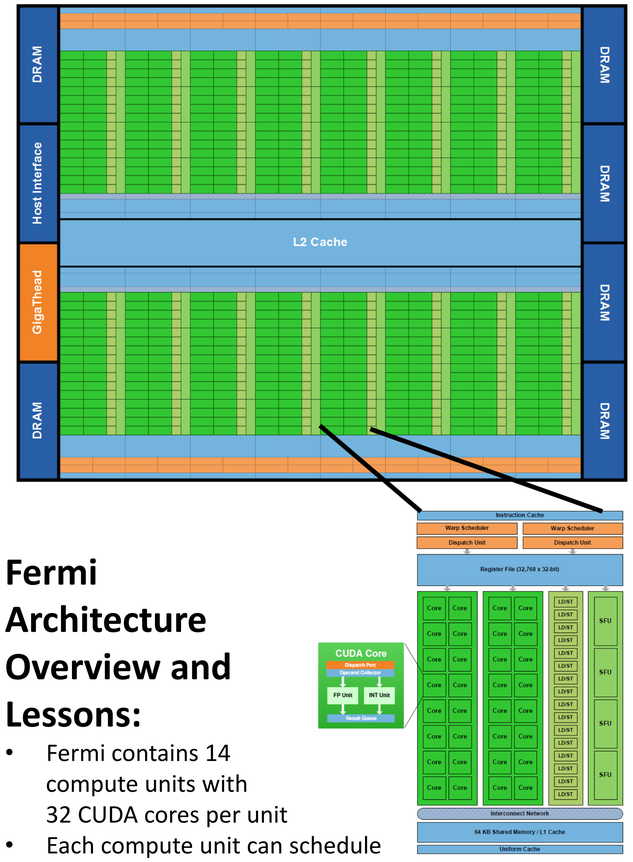
NVIDIA CORPORATION. 2010. NVIDIA Tesla Datasheet, July.

PODLOZHNYUK, V. 2007. Black-Scholes option pricing. Tech.rep., June.

RUETSCH, G., AND MICIKEVICIUS, P. 2010. Optimizing matrix transpose in CUDA. Tech. rep., June.

SMITH, R. 2010. AMD's Radeon HD 6870 & 6850: Renewing competition in the mid-range market. AnandTech (Dec.).

NVIDIA Tesla C2070 (Fermi)



Fermi Architecture Overview and Lessons:

- Fermi contains 14 compute units with 32 CUDA cores per unit
- Each compute unit can schedule two SIMT blocks ("warps") concurrently
- Fermi has a theoretical performance of 1030 GFLOPS
- Fermi has virtual memory and hashes physical addresses, a performance boon against partition camping in SAT
- The addition of an L1 and L2 cache hierarchy relaxes memory coalescing restrictions compared to previous architectures
- OpenCL and CUDA performance is very dependent on GPU compiler optimizations. Both generate PTX files that are executed by the GPU driver

Acknowledgements

Special thanks to Aleksandar Dimitrijevic for providing the AMD Radeon HD 6870 for testing, and to Patrick Cozzi and Jon McCaffrey for their advice and help.