The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite
A. Danalis C. McCurdy J. Meredith P. Roth K. Spafford V. Tipparaju J. Vetter L. Wesolowski
Future Technologies Group, Oak Ridge National Lab | http://ft.ornl.gov/doku/shoc/start

What is SHOC?

SHOC is a benchmark suite for heterogeneous systems focused on scientific computing workloads including common kernels like GEMM, FFT, and stencil computations.

It’s implemented in CUDA and OpenCL for a 1:1 comparison.

It’s distributed with MPI, so you can test multiple GPUs and GPU-accelerated clusters.

It’s open source, and version 3.0 will release on October 1. A beta version is available for download now.

Power Efficiency

We also use SHOC to evaluate the energy efficiency of GPUs. The following graphs show energy efficiency across several devices using the OpenCL version of the benchmarks.

These results were measured using NVIDIA’s GPU Computing SDK 3.0 and AMD’s Stream Computing SDK 4.0. Power measurements are the manufacturers’ thermal design points (TDP).

Resource Contention

The Stable Test

SHOC has an FFT-based stress test based on Primer55’s famous "Torture Test."

This test is designed to stress GPU hardware and identify any errors due to insufficient cooling, bad memory, or other hardware problems.

The test alternates between forward and inverse transforms, and performs the correctness check on the GPU. This in-place computation keeps the GPU hot and is very sensitive to any inaccuracy introduced by hardware errors.

GPU and API Comparison

SHOC benchmarks span a variety of computational and memory access patterns, and are useful for the common task of comparing the performance of varying GPU devices. Scan and FFT results are shown on the right, in units of GFLOPS and GFLOPS, respectively.

As all benchmarks have CUDA and OpenCL versions, you can also compare the performance of these two APIs. The chart below shows this comparison on the Tesla C2050 using NVIDIA GPU Computing SDK 3.0. SGEMM and S3D results are measured in GFLOPS, and Scan and Reduction in GB/sec.

Stability Test

There are a couple of good algorithms out there for sparse matrix-vector multiply, but there’s no clear winner—the best algorithm depends on the structure of the data.

SHOC contains implementations of SpMV based on Bell (10g), SalujaYog, and Vasquez (10g).

Run SpMV on your matrix to decide which algorithmic strategy is best for your code.

Approach:
Scalar – One thread per row
Vector – One warp per row
ELLPACK – Use an alternative data structure

Results in GFLOPS from Tesla C2050, CUDA 3.1, and OpenCL.