

Performance Analysis of Parboil Benchmark using RenderScript

Abdul Dakkak Cuong Manh Pham Prakash Srivastava

University of Illinois at Urbana-Champaign
{dakkak, pham9, psrivas2}@illinois.edu

1. Methodology

1.1 Performance Analysis

Parboil is unique in that it offers optimized benchmark implementations for different targets. Benchmarks have been ported for multicore configurations using C/OpenMP, GPU implementations using CUDA and OpenCL exist as well. There is currently no way to run CUDA on Android devices, but OpenCL 2.0 [3] does have support for *Android Installable Client Driver Extension*. If OpenCL is supported by Android, then we will do performance measurements using it as a base line. If not, then we will use a native client multithreaded C implementation as our base line.

1.2 Power Analysis

Aside from performance, power is an important factor for mobile applications. Based on a quick survey, there does seem to be power profiling tools for android [2]. We will be looking at how the change in language impacts the power usage of the device.

1.3 Programability

The final (objective) metric that we will look at is programability. Performance programming using CUDA or OpenCL exposes low level details, the memory hierarchy being the most noticeable, for hardware. This makes programs written in these languages sensitive to these low level optimizations as they are run on different hardware. Renderscript hides low level details and while promising to be performance portable. In the process of porting the Parboil benchmarks, we will examine these claims.

1.4 Hardware

While development will be against the Android emulator, our performance metrics will be measured on real hardware. Currently we have access to a Nexus 7 Android tablet, but we have contacted the Android development team to find if there is a way to get access to more devices to run our benchmarks.

Another approach to consider is using existing services, such as Apkudo [1], that allow one to test Android applications across devices. The Apkudo service allows one to upload an application and then run it on over 260 android devices.

2. Schedule

References

- [1] Apkudo. Apkudo for developers, Feb. 2014. URL <https://www.apkudo.com/developers.html>.
- [2] Google. Power profiles for android, Feb. 2014. URL <http://source.android.com/devices/tech/power.html>.
- [3] Khronos. Khronos opencl 2.0 specification, Feb. 2014. URL <https://www.khronos.org/opencl/>.