PDFs of Monday 16 April

by vincent – Monday, 16 April 2012

http://www.streamcomputing.eu/blog/2012-04-16/pdfs-of-monday-16-april/

By exception, another PDF-Monday.

OpenCL vs. OpenMP: A Programmability Debate. The one moment OpenCL and the other moment OpenMP produces faster code. From the conclusion: “OpenMP is more productive, while OpenCL is portable for a larger class of devices. Performance-wise, we have found a large variety of ratios between the two solutions, depending on the application, dataset sizes, compilers, and architectures.”

Improving Performance of OpenCL on CPUs. Focusing on how to optimise OpenCL. From the abstract: “First, we present a static analysis and an accompanying optimization to exclude code regions from control-flow to data-flow conversion, which is the commonly used technique to leverage vector instruction sets. Second, we present a novel technique to implement barrier synchronization.”

Variants of Mersene Twister Suitable for Graphic Processors. Source-code at http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/MTGP/

Accelerating the FFDT method using SSE and GPUs. “The Finite-Difference Time-Domain (FDTD) method is a computational technique for modelling the behaviour of electromagnetic waves in 3D space”. This is a project-plan, but describes the theories pretty well.

Auto-tuning a High-Level Language Targeted to GPU Codes. Auto-tuning is (according to us) a very important field in OpenCL/GPGPU the coming years. This article describes HMPP, a language which uses pragmas, and how auto-tuning can help to maximise performance.

Intel SPMD Program Compiler: A SPMD Compiler for High-Performance CPU Programming. If you followed and liked NVidia-vs-Intel on “speed-up for free”, then you should read this.

Parboil: A Revised Benchmark Suite for Scientific and Commercial Throughput Computing. Various benchmarks pop up with GPGPU-support. This is one of them for CUDA-only, (indirectly) funded by Microsoft, Intel and NVidia, but open source. http://impact.crhc.illinois.edu/parboil.php

Dynamic Data Structures for Taskgraph Scheduling Policies with Applications in OpenCL Accelerators. This study is about “stochastic scheduling with precedencies”. Also interesting is that it is done in cooperation with ARM.

A Domain-Specific Approach To Heterogeneous Parallelism. From the abstract: “We propose leveraging domain-specific languages (DSLs) to map high-level application code to heterogeneous devices. To demonstrate the potential of this approach we present OptiML, a DSL for machine learning”.


Implementing FPGA Design with the OpenCL Standard. The official whitepaper of Altera for their upcoming product.

Towards A High-Level Approach for Programming Distributed Systems with GPUs. Introducing SkelCL, a higher level language, and dOpenCL, OpenCL for a distributed systems. No source or binaries, but the presentation describes their approach well to be able to compare with other solutions.

A Comparative Study of Parallel Algorithms for the Girth Problem. A research that compared CUDA against OpenMP with all the results at the last two pages. The girth of a graph is the length of a shortest cycle contained in the graph.

AESOP: Expressing Concurrency in High-Performance System Software. An alternative to current distributed and GPGPU programming languages. Read to see what could be improved in OpenCL according to these researchers.

Exposing Fine-grained parallelism in Algebraic Multigrid Methods. 30 pages to describe the development of ”a parallel algebraic multigrid method which exposes substantial fine-grained parallelism in both the construction of the multigrid hierarchy as well as the cycling or solve stage”.

Robust Volume Segmentation using an Abstract Distance Transform. Segmenting noisy CT and MRI data using GPGPU.

Mobile GPUs. A short, nice overview of the current state of ARM-based GPUs.

I hope you liked this overview – help the other readers by telling interesting discoveries in the PDFs in the comments. And if you have been done research in this field, please contact us.