Introduction

Wolfgang Karl, Samuel Thibault, Stanimire Tomov, and Taisuke Boku

Topic chairs

The recent years have seen great research interest in exploiting GPUs and accelerators for computations, as shown by the latest TOP500 editions, whose very top entries are fully based on their use. Their potential computation power and energy consumption efficiency are appealing, but programming them however reveals to be very challenging, as not only task offloading and data transfer issues show up, but programming paradigms themselves appear to need reconsideration. Fully taping into this new kind of computation resource thus stands out as an open issue, particularly when conjointly using regular CPUs and several accelerator simultaneously. This is why we welcome this year the opening of a new "GPU and Accelerators Computing" topic along the collection of Euro-Par topics. The focus of this topic covers all areas related to accelerators: architecture, languages, compilers, libraries, runtime, debugging and profiling tools, algorithms, applications, etc.

The topic attracted numerous submissions, among which 4 papers were selected for publication. They cover various aspects, such as application-side optimizations, multiple GPU data transfer management, as well as low-level performance analysis.

In "Model-Driven Tile Size Selection for DOACROSS Loops on GPUs", Peng Di and Jingling Xue (from the University of New South Wales, Australia) propose a performance model for tiled and skewed SOR-loops on NVIDIA GPUs, and provide an evaluation of the model accuracy. The model is then used to automatically tune the tile size in a very reduced amount of time, compared to performing measurements of actual runs.

Bertil Schmidt, Hans Aribowo and Hoang-Vu Dang (from the Nanyang Technological University, Singapore); propose a new hybrid format for sparse matrices in "Iterative Sparse Matrix-Vector Multiplication for Integer Factorization on GPUs". After presenting various existing sparse formats, they analyze the shape of the matrices derived from Number Field Sieve problems, and consequently define a new format composed of several slices encoded in a few well-known formats. They present a dual gpu implementation of sparse matrix-vector multiplication (SpMV) with overlapped communications.

"Lessons Learned from Exploring the Backtracking Paradigm on the GPU", from John Jenkins, Isha Arkatkar, John D. Owens, Alok Choudhary, and Nagiza Samatova (from the North Carolina State University, the University of California, Davis, and the Northwestern University, USA), describes how very low performance can get on GPUs when implementing backtracking paradigms.

E. Jeannot, R. Namyst, and J. Roman (Eds.): Euro-Par 2011, LNCS 6853, Part II, pp. 399–400, 2011.

It provides a detailed analysis of the reasons for this and how to optimize at best. It thus provides interesting guidelines for such class of applications.

Last but not least, in “Automatic OpenCL Device Characterization: Guiding Optimized Kernel Design”, Peter Thoman (from the University of Innsbruck, Austria) presents CLbench, a suite of micro-benchmarks which aims at characterizing the performance of implementations of the OpenCL standard, including classic arithmetic and memory performance, but also branching behavior and runtime overhead. The results being provided for 3 OpenCL implementations over 7 hardware/software configuration, they can be used to guide optimizations.