J. Parallel Distrib. Comput. 73 (2013) 1523–1524

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Editorial

Heterogeneity in parallel and distributed computing

Heterogeneity is one of the most profound and challenging features of today's parallel and distributed computing systems. From the macro level, where networks of distributed computers, composed by diverse node architectures, are interconnected with potentially heterogeneous networks, to the micro level, where deeper memory hierarchies, heterogeneous multicores, and various accelerator architectures are increasingly common, the impact of heterogeneity on all computing tasks is increasing rapidly.

This special issue on heterogeneity in parallel and distributed computing is inspired by the great success of the 21st International Heterogeneity in Computing Workshop (HCW 2012, held in Shanghai, China, in May 2012, in conjunction with IPDPS), which attracted 41 high-quality submissions, 17 of which were accepted for presentation and publication in the workshop proceedings. All submissions to this special issue, both extended versions of some HCW 2012 papers and many external original manuscripts, were rigorously reviewed by top-quality experts in the field. The result of the collective efforts of the reviewers and all the authors that submitted their work is this issue, featuring 15 accepted papers. They cover a wide range of topics from multicore processor architecture to scheduling algorithms. A brief outline of the contents of each paper is given in the following paragraphs.

The first two papers evaluate two recent on-chip multiprocessor architectures. In the paper "Design space exploration of on-chip ring interconnection for a CPU–GPU heterogeneous architecture", Jaekyu Lee et al. study a perspective heterogeneous chip multiprocessor architecture integrating on the same chip both CPU and GPU cores. In this architecture, the on-chip interconnection network is used to control the access to the resources shared by the CPU and GPU cores. The study focuses on the impact of this interconnection network on the overall performance of the architecture when CPU and GPGPU applications run simultaneously, and suggests an optimal ring interconnection network.

In the paper "Sparse matrix–vector multiplication on the single-chip Cloud computer many-core processor", Juan C. Pichel and Francisco F. Rivera study the performance potential and power efficiency of an experimental 48-core Intel processor, Single-Chip Cloud Computer (SCC), for execution of such an irregular application as sparse matrix–vector multiplication.

The next paper, "Energy-efficient multithreading for a hierarchical heterogeneous multicore through locality–cognizant thread", by Patrick Anthony La Fratta and Peter M Kogge, deals with a novel heterogeneous multicore processor architecture, passive/active multicore, proposed by the authors in their previous publications. In this paper, the authors present energy-efficient multithreading techniques for this architecture.

A compute node consisting of a multicore CPU and a GPU accelerator as well as HPC systems built from such compute nodes are getting more and more common. The next three papers study optimization of three different classes of applications for efficient execution on these platforms. In the paper "Combining multicore and GPU computing for solving combinatorial optimization problems", Imen Chakroun et al. propose and study optimization techniques for the implementation of branch-and-bound algorithms on a node combining a multicore CPU and a GPU.

The paper "Efficient heterogeneous execution on large multicore and accelerator platforms: case study using a block tridiagonal solver", by Alfred J. Park and Kalyan S. Perumalla, presents a case study applying a number of optimizations to a block tridiagonal solver, which allowed the authors to achieve a 10-fold speedup on a high-end platform consisting of a large number of multicore CPU + GPU nodes, compared with a multicore-only implementation.

The paper "Exploiting hierarchy parallelism for molecular dynamics on a petascale heterogeneous system", by Qiang Wu et al., presents a parallelization scheme for molecular dynamics simulations on high-end systems consisting of multicore CPU + GPU compute nodes and its evaluation on one such a system, a petascale supercomputer TH-1A.

Like the three previous papers, the next paper, "Distributed and hardware accelerated computing for clinical medical imaging using proton computed tomography (pCT)", by Caesar Estrada Ordonez et al., is also application driven, and it presents a parallel implementation of image reconstruction in proton computed tomography on a cluster of 60 compute nodes, each consisting of a multicore CPU and two GPUs.

Efficient implementation of applications on GPU-accelerated platforms requires highly optimized GPU or multi-GPU computational kernels. Development of such kernels is a challenging scientific and engineering problem. Sometimes algorithms and methods considered marginal and inefficient for traditional HPC platforms become superior for GPU-based platforms. In the paper "A block-asynchronous relaxation method for graphics processing units", Hartwig Anzt et al. study the implementation of linear algebra kernels, namely, iterative methods for the solution of linear systems, on GPUs and multi-GPUs. They demonstrate that despite their lower convergence rate the overall performance of asynchronous relaxation methods is better than that of the synchronous ones.

The next three papers address the problem of programmability of heterogeneous parallel platforms. Moises Viñas et al., in their paper "Exploiting heterogeneous parallelism with the heterogeneous programming library", propose and evaluate an alternative to OpenCL, the Heterogeneous Programming Library, for programming heterogeneous compute nodes.

In the paper "dOpenCL: toward uniform programming of distributed heterogeneous multi-/many-core systems", Philipp Kegel et al. present dOpenCL, the extension of OpenCL for clusters of heterogeneous nodes.

0743-7315/$ – see front matter
http://dx.doi.org/10.1016/j.jpdc.2013.08.010

HMPP (Hybrid Multicore Parallel Programming) is a directive-based programming model designed to simplify programming for a single GPU-accelerated compute node. The paper "Generating data transfers for distributed GPU parallel programs", by Frédérique Silber-Chaussumier et al., describes a programming tool that automatically transforms an HMPP program into an HMPP + MPI program for execution on a cluster of GPU-accelerated compute nodes.

The four papers concluding this issue deal with scheduling and resource allocation in heterogeneous environments. In the paper "TLA: temporal look-ahead processor allocation method for heterogeneous multi-cluster systems", Po-Chi Shih et al. propose a new method for dynamic allocation of jobs to processors of a heterogeneous multi-cluster that uses an allocation simulation process.

The paper "Stochastic DAG scheduling using a Monte Carlo approach", by Wei Zheng and Rizos Sakellariou, proposes a new approach to scheduling a DAG of tasks with the uncertainty in individual task execution times based on a Monte Carlo method.

In the paper "On minimizing the resource consumption of cloud applications using process migrations", Nikos Tziritas et al. propose a fully distributed algorithm for dynamic selection of an assignment scheme between processes and a virtual machine and the computing nodes of the cloud system aimed at minimization of the resource consumption.

The paper "Robust static resource allocation of DAGs in a heterogeneous multicore system", by Luis Diego Briceno Guerrero et al., presents a study motivated by an application involving processing a large amount of satellite data on a heterogeneous cluster of multicore compute nodes under a deadline constraint. The authors propose and analyze five allocation heuristics aimed at assignment of the tasks to the cluster in a way that all tasks complete before a common deadline, and their completion times are robust against uncertainties in execution times.

Alexey Lastovetsky
School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland
E-mail address: Alexey.Lastovetsky@ucd.ie.

15 August 2013
Available online 29 August 2013