Computing (2014) 96:1113–1114
DOI 10.1007/s00607-013-0379-0

EDITORIAL

Introduction to the computing special issue: performance portability and tuning for multi-core and many-core computing systems

Sabri Pllana · Jacob Barhen

Published online: 13 December 2013

The pervasiveness of homogeneous and heterogeneous multi-core and many-core processors, in a large spectrum of systems from embedded and general-purpose to high-end computing systems, poses major challenges to software industry. In general, there is no guarantee that software developed for a particular architecture will be executable (that is functional) on another architecture. Furthermore, ensuring that the software preserves some aspects of performance behavior (such as temporal or energy efficiency) across different such architectures is an open research issue. Therefore, this special issue focuses on novel solutions for functional and performance portability as well as automatic tuning across different architectures.

This special issue includes six papers. Authors of these papers have presented their preliminary research results at MuCoCoS workshop that was organized in conjunction with the 2012 Supercomputing Conference (SC12), Salt Lake City, Utah, November 16, 2012.

"*A Low Level Component Model Easing Performance Portability of HPC Applications*"—by Julien Bigot, Zhengxiong Hou, Christian Perez and Vincent Pichon—addresses the issue of performance portability. The proposed component model supports C++, Fortran, MPI, and Corba. Its usefulness is evaluated with experiments on Grid5000 cluster and on Curie supercomputer.

"*Efficient and Highly Portable Deterministic Multithreading (DetLock)"*—by Hamid Mushtaq, Zaid Al-Ars, Koen Bertels—presents an approach for deterministic

S. Pllana (✉)
Department of Computer Science, Linnaeus University, SE-351 95 Växjö, Sweden
e-mail: sabri.pllana@lnu.se

J. Barhen
Oak Ridge National Laboratory, PO BOX 2008 MS6015, Oak Ridge, TN 6015, USA
e-mail: barhenj@ornl.gov

execution of multi-threaded programs. The LLVM compiler is used to insert the code for updating logical clocks.

"*Experiences with OpenMP, PGI, HMPP and OpenACC Directives on ISO/TTI Kernels*"—by Sayan Ghosh, Terrence Liao, Henri Calandra, Barbara M. Chapman—reviews the performance and programmability of various directive-based approaches for programming GPU-accelerated systems. Authors use Tilted Transversely Isotropic kernels for evaluation of directive-based approaches in this context.

"*Energy-centric DVFS Controlling Method for Multi-core Platforms"*—by Shin-gyu Kim, Hyeonsang Eom, Heon Y. Yeom, Sang Lyul Min—presents a prediction and controlling approach for dynamic voltage and frequency scaling. Authors evaluated their approach using the SPEC CPU2006 benchmark.

"*Performance and Energy Impact of Parallelization and Vectorization Techniques in Modern Microprocessors*"—by Juan M. Cebrian, Lasse Natvig, Jan Christian Meyer—using the PARSEC benchmark suite, studies how parallelization and vectorization affects the energy efficiency for Intel(R) Core(TM) i5/i7 and ARM(R) Cortex(TM) A9/A15 processors.

"*The PEPPHER Composition Tool: Performance-Aware Composition for GPU-based Systems*"—by Usman Dastgeer, Lu Li, Christoph Kessler—presents a compo nent-based approach for automatic composition of performance-oriented applications for GPU-accelerated computing systems. Authors evaluate their approach using Rodinia benchmark suite.

We are grateful to all authors who contributed papers to this special issue. Furthermore, we would like to acknowledge the encouragement and support of Schahram Dustdar (Editor-in-Chief of Computing), Priya Balamurugan (JEO Assistant Computing), Visveswaran Devaraj (Production Editor), and Christine Kamper (Assistant to the Editor-in-Chief).

Sabri Pllana and Jacob Barhen
Special Issue Editors