Introduction

Sabri Pllana, Jean-François Méhaut, Eduard Ayguade,
Herbert Cornelius, and Jacob Barhen

Topic chairs

Modern multicore and manycore systems offer impressive performance for various applications. However, achieving this performance is a challenging task. While multicore and manycore processors alleviate several problems that are related to single-core processors – known as memory wall, power wall, or instruction-level parallelism wall – they raise the issue of the programmability wall. The multicore and manycore programmability wall calls for new parallel programming methods and tools. Therefore, this topic focuses on novel solutions for efficient programming of multicore and manycore processors in the context of general-purpose and embedded systems.

The quality of submissions was very high. Papers have been selected based on the recommendations of at least four reviewers. The nine accepted papers address a representative set of issues related to the multicore and manycore programming.

The paper – "Hardware and Software Tradeoffs for Task Synchronization on Manycore Architectures" by Yonghong Yan, Sanjay Chatterjee, Daniel Orozco, Elkin Garcia, Zoran Budimlic, Jun Shirako, Robert Pavel, Guang R. Gao, and Vivek Sarkar – describes an implementation of the "phasers" synchronization construct on the IBM Cyclops64 manycore processor.

In the paper – "OpenMPspy: Leveraging Quality Assurance for Parallel Software" by Victor Pankratius, Fabian Knittel, Leonard Masing, and Martin Walser – authors describe OpenMPspy. This tool may be used for detecting mistakes that occur while the code is typed in Eclipse and for collecting statistics on the use of OpenMP language constructs.

The paper – "A Generic Parallel Collection Framework" by Aleksandar Prokopec, Phil Bagwell, Tiark Rompf, and Martin Odersky – describes an approach for development of parallel containers such as parallel arrays or hash maps.

In the paper – "Progress Guarantees when Composing Lock-free Objects" by Nhan Nguyen Dang and Philippas Tsigas – authors describe a novel synchronization mechanism for composing lock-free data objects that guarantees lock-free progress.

The paper – "Engineering a multicore Radix Sort" by Jan Wassenberg and Peter Sanders – describes a novel variant of radix sorting algorithm that is based on a micro-architecture-aware variant of counting sort.

E. Jeannot, R. Namyst, and J. Roman (Eds.): Euro-Par 2011, LNCS 6853, Part II, pp. 110–111, 2011.

In the paper – "Accelerating code on multicores with FastFlow" by Marco Aldinucci, Marco Danelutto, Peter Kilpatrick, Massimiliano Meneghin, and Massimo Torquati – authors describe an approach for parallelization of sequential codes via thread-offloading. Basically, a thread uses other threads as software accelerators.

The paper – "A Novel Shared-Memory Thread-Pool Implementation for Hybrid Parallel CFD Solvers" by Jens Jgerskpper and Christian Simmendinger – describes an approach for shared-memory parallelization of grid-based CFD solvers.

In the paper – "A Fully Empirical Autotuned Dense QR Factorization for Multicore Architectures" by Emmanuel Agullo, Jack Dongarra, Rajib Nath, and Stanimire Tomov – authors describe an empirical approach for tuning dense linear algebra libraries on multicore architectures.

The paper – "Parallelizing a Real-Time Physics Engine Using Transactional Memory" by Jaswanth Sreeram and Santosh Pande – describes experiences of authors in parallelizing the ODE physics engine that is used in computer games.

We are grateful to all authors for submitting their high-quality papers to this topic and to reviewers for their efforts to evaluate submitted papers. Furthermore, we would like to acknowledge the encouragement and support of conference chairs Emmanuel Jeannot, Raymond Namyst, and Jean Roman.