Int J Parallel Prog (2008) 36:163–165
DOI 10.1007/s10766-007-0065-y

GUEST EDITORIAL

Introduction to the Special Issue on the 18th International Symposium on Computer Architecture and High Performance Computing

Alberto F. De Souza · Rajkumar Buyya

Received: 1 November 2006 / Accepted: 2 April 2007 / Published online: 10 January 2008

This special issue of the International Journal of Parallel Programming is devoted to some of the best papers presented at the 18th edition of the International Symposium on Computer Architecture and High Performance Computing, also known as SBAC-PAD (the acronym comes from the time the symposium was in Portuguese—*Simpósio Brasileiro de Arquitetura de Computadores e Computação de Alto Desempenho*—and is kept for tradition), which was held at the city of Ouro Preto, State of Minas Gerais, Brazil, in October, 2006. Annually, SBAC-PAD gathers users, developers and researchers from academic, industrial and governmental R&D institutions to discuss progress and exchange ideas on the many aspects of computer architecture and high performance parallel and distributed computing. SBAC-PAD'2006 was sponsored by the Brazilian Computer Society (*Sociedade Brasileira de Computação*—SBC), co-sponsored by IEEE Computer Society (TCCA – TCSC), and held in cooperation with IFIP Working Group 10.3 (Concurrent Systems). The six SBAC-PAD'2006 papers selected to be included in this issue reflect the breadth of the topics covered in the symposium and the many opportunities there are today for research on computer architecture and high performance computing.

This special issue leads off with the paper selected for the SBAC-PAD'2006 Best Paper Award. In this paper, named *Dual-Thread Speculation: A Simple Approach to Uncover Thread-Level Parallelism on a Simultaneous Multithreaded Processor*, Warg and Stenstrom present a dual-thread speculation system that can be used for running

A. F. De Souza (✉)
Laboratório de Computação de Alto Desempenho, Departamento de Informática, Universidade Federal do Espírito Santo, Vitoria, ES, Brazil
e-mail: alberto@lcad.inf.ufes.br

R. Buyya
Grid Computing and Distributed Systems Laboratory, Department of Computer Science and Software Engineering, University of Melbourne, Carlton, VIC, Australia

sequential programs in multithreaded architectures. Their system automatically discovers thread-level parallelism in sequential programs and dynamically exploits multithreaded hardware for improving the performance of these programs. It extracts threads from loops and procedures/methods and uses thread-level speculation as a means to execute these threads in parallel in multithreaded hardware. In the second paper, *Dynamic Instruction Scheduling in a Trace-Based Multi-Threaded Architecture*, Rounce and De Souza present a multithreaded extension of the Dynamically Trace Scheduled VLIW (DTSVLIW) architecture, named multithreaded DTSVLIW (mDTSVLIW). The DTSVLIW architecture executes code in two modes: sequential and VLIW. The first time a trace of code is followed, the DTSVLIW executes it sequentially, dynamically schedules it into VLIW instructions, and saves these VLIW instructions in a VLIW cache. If the same trace is followed again, the DTSVLIW executes it in VLIW mode. The mDTSVLIW architecture uses the DTSVLIW scheduling logic to schedule instructions from several processes and to produce parallelized code for several threads. The code of these threads is later combined to produce a single simultaneous multithreaded instruction stream for a wide issue VLIW processor. Still in the multithread arena, in the third paper, *Analyzing the Effects of Hyper-threading on the Performance of Data Management Systems*, Hassanein, Rashid and Hammad use the hardware counters provided by the Pentium 4 Hyper-Threading processor to extensively evaluate its multithreaded micro-architecture and study the memory system behavior of each query running on a database management system (DBMS).

In the fourth paper, *The ParTriCluster Algorithm for Gene Expression Analysis*, Araujo et al. present a parallel version of the Tricluster algorithm, which is used in bioinformatics for finding 3D clusters of genes, samples and timestamps; i.e., groups of genes that show similar cellular expression across many biological samples over similar time periods. They named the parallel algorithm ParTriCluster and implemented it using the Anthill distributed run-time environment. Anthill allows exploiting the task and data parallelism available in applications by dividing the computation into multiple pipeline stages (task parallelism), which can be replicated multiple times (to handle data in parallel). In the fifth paper, *A Run-time System for Efficient Execution of Scientific Workflows on Distributed Environments*, Teodoro et al. present an Anthill extension designed to facilitate the implementation of parallel programs in this distributed run-time environment. The extended Anthill framework allows users to store and share data processing components in a repository, and provides a toolkit for generating parallel and distributed workflows based on these components.

In the sixth paper, *Modulo Path History for the Reduction of Pipeline Overheads in Path-Based Neural Branch Predictors*, Loh and Jiménez present a new neural-inspired branch predictor that is based on the path-based neural branch predictor (PBNP). In the PBNP, the path history length is equal to the history length, which is typically very long for high accuracy. In their paper, Loh and Jiménez propose to decouple the path-history length from the outcome-history length through a new technique called modulo-path history.

We would like to thank both the authors and referees for their hard work and cooperation. We also would like to express our appreciation to the members of the SBAC-PAD steering committee, organizing committee, program committee, and to the participants

of SBAC-PAD'2006 for making the conference both technically outstanding and socially enjoyable.

SBAC-PAD'2006 Program Chairs
Alberto F. De Souza
Rajkumar Buyya