Editorial

Gabriel P. Silva*

Department of Computer Science,
Federal University of Rio de Janeiro (UFRJ),
Rio de Janeiro, RJ, 21941-950, Brazil
Fax: +55-21-2598-9515
E-mail: gabriel.silva@ufrj.br
*Corresponding author

Denise Stringhini

Informatics and Computing Faculty,
Mackenzie Presbyterian University,
São Paulo, SP, 01302-907, Brazil
Fax: +55-11-21148286
E-mail: dstring@mackenzie.br

Biographical notes: Gabriel P. Silva received his BS in Electronic Engineering from Federal University of Rio de Janeiro, in 1983 and later his MSc and DSc in Computer Science and Systems Engineering from Federal University of Rio de Janeiro – UFRJ/COPPE, in 1991 and 2000, respectively. He used to work at Cobra – Brazilian Computers between 1984 and 1986 in the development of parallel computers. He also worked as a Researcher in the area of Computer Architecture at UFRJ/NCE until 2006. Currently, he is a Professor and the Head of the Computer Science Department of the Mathematics Institute from Federal University of Rio de Janeiro. His research interests include computer architecture, microprocessor architecture, parallel programming and high performance computing.

Denise Stringhini received her Bachelors in Informatics from Pontifícia Universidade Católica do Rio Grande do Sul in 1993, and obtained her MSc and PhD in Computer Science both from Universidade Federal do Rio Grande do Sul, in 1997 and 2002, respectively. Currently, she is a Professor at Mackenzie Presbyterian University. She has experience in the areas of computer science, with emphasis on computing systems area and the following research themes: parallel debugging, selection and visualisation of processes, parallel programming and clusters.

We are very pleased to welcome you to this edition of the *International Journal of High Performance Systems Architecture*. This is a special issue on parallel architectures and applications which comprises a set of seven papers very carefully selected from the best papers from WSCAD-SSC Conference in 2009. The authors of the selected papers submitted an extended version of their work and then all the papers went again through a new review process. We are thankful to the authors for their contributions to this special issue. Next, a brief introduction to these papers.

In the paper entitled 'A minimalist cache coherent MPSoC designed for FPGAs', authors describe the design and VHDL implementation of a cache coherent MPSoC named minimalist cache coherent MPSoC (MCCM). The system comprises 1 to 8 MIPS-I processors, coherent primary data caches, memory management units, memory controller and their interconnection. The authors present a detailed account of the implementation, focusing on the shared memory subsystem. A simple benchmark is used to assess the overall system functionality. The paper also compares the size of the proposed MCCM design to that of a LEON3-based multiprocessor and it was found that a 4-core LEON3 system needs roughly the same amount of logic/state as a 6 to 8 cores MCCM.

'Dynamic workload balancing deques for branch and bound algorithms in the message passing interface' is a paper that proposes the use of a MPI library to provide near-optimal dynamical workload balancing over branch and bound (B&B) algorithms, since B&B potentially produces huge workload unbalance during a parallel execution. The library, named RaWSDM, provides a double ended queue (deque) on which the programmer may pop, process, and, later, pull back some parallel tasks. An underlying efficient system scheduler is responsible for keeping the workload balanced by exchanging elements among all deques. Theoretical bounds are traced and practical experiments are performed with the unlimited knapsack problem. Results show a performance gain over 80% against a pure MPI implementation using round-robin scheduling, with near linear speedup and memory consumption.

Other interesting paper is 'Challenges and solutions to improve the scalability of an operational regional

meteorological forecasting model'. This work investigates the parallel scalability of BRAMS, a limited area weather forecasting production code, from O(100) cores to O(1,000) cores on large grids (20 km and 10 km resolution over South America). Initial experiments show lack of scalability at modest core count. Execution time profiling and source code examination revealed the causes of the limited scalability: sequential algorithms and extensive memory requirements at scarcely used phases of the computation. Algorithm replacement and memory reduction generate a new code version that possesses strong and weak scaling. The new version achieved a speed-up of 6 using 100 to 700 processors on the 20 km resolution grid and a speed-up of 6.9 on the same processor range on the 10 km resolution grid. This paper received the best paper award of the conference.

The paper 'Automated refactorings for high performance Fortran programmes' analyses refactoring, a software engineering technique aimed at improving the source code of software applications, without changing their external behaviour. In this work, a set of automated refactorings for Fortran based on the Photran plug-in, which is integrated with the Eclipse integrated development environment (IDE), is presented. It is also presents a set of experiments to evaluate the impact of the proposed refactorings in third-party Fortran applications. The results show that such refactorings improve the design of existing code without compromising performance and may even increase it through code restructuring.

The following paper is 'Assessing the influence of data access patterns and contention management policies on the performance of software transactional memory systems'. Transactional memory was proposed as a mean for easing the burden of traditional concurrency control mechanisms. The programmer has only to mark the code sections that are to be executed atomically, and the system takes care of the synchronisation details. As transactions are executed in parallel, some of them are likely to access resources in ways that cannot be conciliated. Conflicts among transactions are mediated by a contention manager. In this paper, the authors present a novel approach to contention management (CM), which binds different CM strategies to different data in a programme, based on the access patterns to these data.

The authors show how it can be done in a way that introduces minimum overhead and present benchmark results to evaluate their implementation, also demonstrating how the best CM strategy may vary under different levels of contention, under a varying number of threads per processing core, and under different system architectures.

The paper 'The impact of applications' I/O strategies on the performance of the Lustre parallel file system' studies and evaluates some I/O techniques to improve the performance of parallel applications on the Lustre file system. The evaluated techniques are suitable for situations like the access by different instances of a parallel application to exclusive data stored in the file system. The authors provide a guide to help developers to tune their application to extract the best performance out of Lustre. Their results show expressive gains in performance related with the choice of access pattern of the application. The authors present also considerations on operation granularity, intra-node concurrency and temporal behaviour of the application.

Finally, we present the paper '*Trebuchet*: exploring *TLP* with dataflow virtualisation'. Parallel programming has become mandatory to fully exploit the potential of multi-core CPUs. The dataflow model provides a natural way to exploit parallelism. However, specifying dependencies and control using fine-grained instructions in dataflow programmes can be complex and present unwanted overheads. This paper presents TALM: a coarse-grained dataflow execution model to be used on top of widespread architectures. TALM is implemented as the Trebuchet virtual machine for multi-cores. The programmer identifies code blocks that can run in parallel and connects them to form a dataflow graph, which allows one to have the benefits of parallel dataflow execution in a Von Neumann Machine, with small programming effort.

The guest editors would like to thank all the programme committee members, session chairs and the audience of the WSCAD-SSC 2009 for their work and participation. We also would like to thank Dr. Nadia Nedjah (*IJHPSA* Editor-in-Chief), Dr. Felipe França for the opportunity, the editorial board members and especially the journal reviewers, for their important collaboration, reviewing the final papers that allowed this special issue become a reality.