

## Guest Editorial: High-Level Parallel Programming and Applications

Marco Danelutto<sup>1</sup> · Susanna Pelagatti<sup>1</sup> · Massimo Torquati<sup>1</sup>

Published online: 27 June 2016  
© Springer Science+Business Media New York 2016

The arrival of the multi-/many-core systems has produced a game-changing event for the computing industry, which today, much more than few years ago, is relying on parallel processing as a means of improving application performance. Although a wide gap still exists between parallel architectures and parallel programming maturity, the only way forward to keep increasing performance and reducing power consumption is through parallelism. Any program must become a parallel program in order to exploit the capabilities of modern computers at any scale.

In the industrial practice, parallel programming is still dominated by low-level machine-centric unstructured approaches based on tools and specialized libraries that originate from high performance computing. Parallel programming at this level of abstraction is difficult, error-prone, time-consuming and, hence, economically infeasible in most application domains. Now, more than ever, it is crucial that the research community makes a significant progress toward making the development of parallel code accessible to all programmers, rather than allowing parallel programming to continue to be the domain of specialized expert programmers. Achieving a proper trade-off among performance, programmability and portability issues is becoming a must.

Parallel and distributed programming methodologies are currently dominated by low-level techniques such as send/receive message passing or data sharing coordinated by locks. These abstractions are not a good fit for reasoning about parallelism. In this evolution/revolution phase, a fundamental role is played by high-level and portable programming tools as well as application development frameworks. They may offer

---

✉ Susanna Pelagatti  
susanna.pelagatti@unipi.it

<sup>1</sup> Department of Computer Science, University of Pisa, Pisa, Italy

many advantages and have a key role to play in the scalable exploitation of ubiquitous parallelism.

Since 2001 the High-Level Parallel Programming and Applications (HLPP) series of initially workshops and later symposia has been the prime forum for researchers developing state-of-the-art concepts, tools and applications for high-level parallel programming. The general emphasis is on software quality, programming productivity and high-level performance models. HLPP solicits contributions from theory to practice, from methodologies to tools and encourages novel ideas in high-level parallel programming originating from imperative, object-oriented, functional, logic or any other programming paradigm.

The 8th Symposium on High-Level Parallel Programming and Applications has been organized by the Computer Science Department of the University of Pisa and held on July the 2nd and the 3rd, 2015. The invited talk was given by Prof. Christoph Kessler from Linköping University with the title “Composing multi-variant program building blocks: From local to global optimization”.

Twenty nine papers were submitted, out of which the international programme committee selected 11 papers for presentation at the symposium and inclusion in this journal special issue. Each paper received at least 3 reviews from experts. The journal articles have undergone an extra two rounds of rigorous peer-review according to the journal’s high standards. In the following, we provide a brief introduction to each paper selected.

Five papers address different aspects of using hardware accelerators for parallel computations. In “Autonomic Coordination of Skeleton-based Applications over CPU/GPU Multi-Core Architectures”, Mehdi Goli and Horacio González-Vélez present a methodology for automatically coordinating skeleton-based applications over heterogeneous CPU/GPU platforms. In “Using the Xeon Phi platform to run speculatively-parallelized codes”, Alvaro Estebanez, Diego R. Llanos and Arturo Gonzalez-Escribano study the performance of the Intel Xeon Phi accelerator platform for code parallelization when using thread-level speculation techniques. Functional programming languages can be used for high-level and efficient GPGPU programming as presented in “High Level Data Structures for GPGPU Programming in a Statically Typed Language” by Mathias Bourgoïn, Emmanuel Chailloux and Jean-Luc Lamotte. Another paper in this category is “Automatic CPU/GPU Generation of Multi-versioned OpenCL Kernels for C++ Scientific Applications” by Rafael Sotomayor, Luis Miguel Sanchez, Javier Garcia Blas, Javier Fernandez and J. Daniel Garcia. This paper studies the problem of automatically generate OpenCL kernel code starting from sequential application annotated with C++11 attributes. In “Data Parallel Algorithmic Skeletons with Accelerator Support”, Steffen Ernsting and Herbert Kuchen discuss the extensions of the Muesli high-level skeleton library to encompass support for GPUs and for Xeon Phi by using both C++ and Java.

Four papers are devoted to more theoretical aspects. In “Calculating Parallel Programs in Coq using List Homomorphisms”, Frédéric Loulergue, Wadoud Bousdira and Julien Tesson present a framework for refactoring functional programs to introduce parallelism that uses the Coq proof assistant and its code extraction capabilities, targeting OCaml and BSML for parallel execution. The design, implementation and evaluation of a domain specific language for graph processing based on structural

recursion is presented in “Towards Systematic Parallelization of Graph Transformations over Pregel” by Le-Duc Tung and Zhenjiang Hu. In “Multi-ML: Programming Multi-BSP Algorithms in ML”, Victor Allombert, Frédéric Gava and Julien Tesson present Multi-ML, a new language aiming at providing a way to program Multi-BSP algorithms. Another paper in this category is “Functional Models of Hadoop MapReduce with Application to Scan” by Kiminori Matsuzaki that presents two different functional models for Hadoop map-reduce implementations.

A novel application of the pattern/skeleton oriented methodology into the domain of Data Stream Processing has been explored in “Parallel Patterns for Window-based Stateful Operators on Data Streams: an Algorithmic Skeleton Approach” by Tiziano De Matteis and Gabriele Mencagli.

Finally, a novel framework to support the staging of large-scale HPC applications has been presented in “Tackling Complexity in High Performance Computing Applications” by J. Darlington, A.J. Field and L. Hakim.

The organizers of HLPP 2015 wish to take this opportunity to thank the members of the programme committee for their outstanding work:

- Marco Aldinucci, University of Torino, Italy
- Jost Berthold, University of Copenhagen, Denmark
- Murray Cole, University of Edinburgh, United Kingdom
- Marco Danelutto, University of Pisa, Italy (co-chair)
- Patrizio Dazzi, ISTI CNR Pisa, Italy
- Francisco De Sande, Universidad de La Laguna, Spain
- Kento Emoto, Kyushu Institute of Technology, Japan
- Jol Falcou, MetaScale/University of Paris-South, France
- Clemens Grelck, University of Amsterdam, Netherlands
- Gatan Hains, Huawei FRC Boulogne-Billancourt, France
- Kevin Hammond, University of St. Andrews, UK
- Ali Jannesari, German Research School for Simulation Sciences, Germany
- Youry Khmelevsky, University of British Columbia (Okanagan)/Okanagan College, Canada
- Peter Kilpatrick, Queen’s University Belfast, UK
- Herbert Kuchen, University of Münster, Germany
- Kiminori Matsuzaki, Kochi University of Technology, Japan
- Susanna Pelagatti, University of Pisa, Italy (co-chair)
- Frank Penczek, Intel, Germany
- Tiark Rumpf, Oracle Labs/EPF Lausanne, Switzerland
- Kostis Sagonas, Uppsala University, Sweden
- Francisco de Sande, University of La Laguna, Spain
- Sven-Bodo Scholz, Heriot-Watt University, United Kingdom
- Julien Tesson, University of Paris-East, France
- Massimo Torquati, University of Pisa, Italy (co-chair)

We extend our gratitude to the numerous additional reviewers who with their expertise substantially contributed to shape the symposium programme: Emmanuel Melin, Chong Li, Sbastien Monnet, Juan Jos Fumero Alfonso, Claudia Misale, Mau-

rizio Drocco, Steffen Ernsting, Mathias Bourgoïn, Steffen Ernsting, Maxime Vron, Emanuele Carlini, Alan Stewart, Le-Duc Tung, Chong Li and Xiaodong Meng.

Collectively, the 11 papers included in this special provide a detailed compilation of the diverse range of issues currently being investigated in the field of High-Level Parallel Programming. They will undoubtedly encourage engaging work in this exciting field. We hope you enjoy this special issue and take some inspiration from it for your own future research.

M. Danelutto, S. Pelagatti and M. Torquati

Co-editors of the IJPP special issue and co-organizers of HLPP2015

Department of Computer Science, University of Pisa