

Guest Editorial: High-Level Parallel Programming with Algorithmic Skeletons

Sergei Gorlatch¹ · Herbert Kuchen²

Published online: 24 May 2017

© Springer Science+Business Media New York 2017

The architectures of high-performance computers become increasingly complex. A typical machine consists of several computing nodes connected by some high-speed network. Each node provides multiple processing cores and possibly several accelerators such as graphics processing units (GPUs). The complexity of such an architecture leads to the complexity of the software running on it. Typically, message passing based on MPI is employed for communicating over the high-speed network. Within a computing node, approaches such as OpenMP allow to coordinate and properly synchronize several threads running on the multiple cores. Additionally, accelerators with an own complicated memory and thread hierarchy may be used on such computing nodes. Low-level approaches such as OpenCL and CUDA are mostly used to program them. A major challenge here is the efficient organization of a complex thread hierarchy, as well as the memory transfer between main memory of the CPU and device storage. Overall, programming contemporary high-performance machines requires the sophisticated combination of several parallel-computing frameworks such as MPI, OpenMP, OpenCL, and CUDA. This often overburdens application programmers, which would prefer to just focus on their application domain. But even for experts in parallel programming, this style of programming is tedious and error-prone.

Current research attempts to replace the low-level frameworks by a more high-level programming approach which hides technical details as much as possible. Algorithmic skeletons are such an approach, which has attracted significant attention. Algorithmic

✉ Herbert Kuchen
kuchen@uni-muenster.de

Sergei Gorlatch
gorlatch@uni-muenster.de

¹ Institut für Informatik, Universität Münster, Münster, Germany

² ERCIS, Universität Münster, Münster, Germany

skeletons are typical parallel-programming patterns which are efficiently implemented on the available hardware and offered to the user with an easy-to-use application programming interface (API). Problem-specific details are typically passed as skeletons' parameters. This significantly simplifies and speeds-up the parallel programming process. Communication and synchronization problems such as deadlocks and race conditions are avoided by design. Developing parallel software with algorithmic skeletons essentially boils down to composing a few skeletons and providing them with application-specific parameters.

This special issue of the International Journal of Parallel Programming is devoted to high-level parallel programming with algorithmic skeletons. It comprises eight papers, listed in the sequel. In their contribution "The Missing Link! A New Skeleton for Evolutionary Multi-Agent Systems in Erlang", Christopher Brown et al. present a new skeleton for evolutionary multi-agent systems. The paper "High-Level Programming for Many-Cores using C++14 and the STL" by Michael Haidl and Sergei Gorlatch presents an approach for programming CPU and accelerators in a unified way using exclusively C++ and STL. Fabian Wrede and Steffen Ernsting investigate in their article "Simultaneous CPU-GPU Execution of Data Parallel Algorithmic Skeletons", how far a combination of CPU and GPU usage can speedup the execution of skeletons. August Ernstsson, Lu Li, and Christoph Kessler have contributed a paper on "Flexible and type-safe skeleton programming for heterogeneous parallel systems". They present the skeleton framework SkePU 2 for heterogeneous parallel systems. The paper "Analysing Multiple QoS Attributes in Parallel Design Patterns-based Applications" by Antonio Brogi et al. presents a probabilistic approach to select the best combination of skeletons w.r.t. considered quality of service attributes. Ari Rasch and Sergei Gorlatch extend the notion of homomorphism towards multi-dimensional arrays, introduce a new skeleton for such homomorphisms, and develop an efficient OpenCL implementation schema for this skeleton. Their paper is entitled "Multi-Dimensional Homomorphisms and Their Implementation in OpenCL". Mehdi Goli and Horacio Gonzalez-Velez have written a contribution on the "Formalised Composition and Interaction for Heterogeneous Structured Parallelism". They propose a grammar to build block components to execute computational functions in heterogeneous multi-core architectures. Finally, Venkatesh Kannan and Geoff W. Hamilton present in their article "Functional Program Transformation for Parallelisation using Skeletons" a program transformation which eliminates overhead due to intermediate data structures in a sequence of skeleton calls.

The mentioned articles are extended versions of selected contributions presented at the 9th Symposium on high-level parallel programming (HLPP), which took place on July 4–5, 2016, in Münster, Germany. HLPP is series of international symposia, which started in 2001 and has since then been a forum for researchers developing state-of-the-art concepts, tools, and applications for high-level parallel programming.

The papers contained in this special issue provide an overview of the aspects which are currently being investigated in the field of high-level parallel programming in general and algorithmic skeletons in particular. We hope you will enjoy this special issue and take some inspiration from it.

We thank the following programme-committee members and reviewers for their careful work on reviewing the selected articles: Marco Aldinucci, Rob Bisseling,

Murray Cole, Marco Danelutto, Maurizio Drocco, Francisco de Sande, Clemens Grelck, Bastian Hagedorn, Joel Falcou, Gaétan Hains, Kevin Hammond, Zhenjiang Hu, Christoph Kessler, Peter Kilpatrick, Kiminori Matsuzaki, Susanna Pelagatti, Aleksandar Prokopec, Ari Rasch, Kostis Sagonas, Michel Steuwer, Massimo Torquati, and Fabian Wrede.

We are also grateful to the editorial team of Springer Verlag for enabling this special issue and for their continuous support.

Sergei Gorlatch and Herbert Kuchen

Guest editors of the IJPP special issue on
“High-Level Parallel Programming with Algorithmic Skeletons”
Münster, Germany, May 5, 2017