DOI 10.12694/scpe.v18i1.1229
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 18, Number 1, pp. iii–iv. http://www.scpe.org

INTRODUCTION TO THE SPECIAL ISSUE ON PRACTICAL ASPECTS OF HIGH-LEVEL PARALLEL PROGRAMMING

From smartphones to supercomputers, parallel architectures are nowadays pervasive. However parallel programming is still reserved to experienced and trained programmers. The trend is towards the increase of cores in processors and the number of processors in multiprocessor machines: The need for scalable computing is everywhere. But parallel and distributed programming is still dominated by low-level techniques such as send/receive message passing and POSIX threads. Thus high-level approaches should play a key role in the shift to scalable computing in every computer and device.

Algorithmic skeletons (Google's MapReduce being the most well-known skeletal parallelism approach), parallel extensions of functional languages such as Haskell and ML, parallel logic and constraint programming, parallel execution of declarative programs such as SQL queries, meta-programming in object-oriented languages, etc. have produced methods and tools that improve the price/performance ratio of parallel software, and broaden the range of target applications. Also, high level languages offer a high degree of abstraction which ease the development of complex systems. Moreover, being based on formal semantics, it is possible to certify the correctness of critical parts of the applications. The aim of all these languages and tools is to improve and ease the development of applications.

The paper presented in this special issue are extended versions of papers presented at the Practical Aspects of High-Level Parallel Programming (PAPP) workshop that was affiliated to the International Conference on Computational Science (ICCS) from 2004 to 2012, and the Practical Aspects of High-Level Parallel Programming (PAPP) track of the ACM Symposium on Applied Computing (SAC). It also features a paper written specifically for this special issue.

While a lot of work in high-level approaches focus on regular data structures such as arrays and multi-dimensional arrays, the papers of this issue consider more irregular data structures, in particular trees and graphs.

Kaheki, Matsuzaki and Emoto present a new approach for parallel reduction on trees in a distributed memory setting. Their algorithm is based on a suitable serialization of trees and its efficiency is shown for the Bulk Synchronous Parallel (BSP) model. Matzusaki proposes a set of efficient skeletons on distributed trees: their efficient implementation relies on the segmentation of trees based on the idea of m-bridges that ensures high locality, and allows to represent local segments as serialized arrays for high sequential performance.

Dazzi, Carlini, Lulli and Ricci propose Telos, a high-level approach for large graphs processing based on overlay networks, and an implementation on top of Apache Spark.

Hajibaba, Gorgin, and Sharifi present a sequence similarity parallelization technique in the context of another Apache project: Storm a stream processing framework.

I wish to thank Anne Benoît and Fré dé ric Gava, co-chairs of some the previous PAPP workshops and the program committee members of the PAPP workshops and the PAPP ACM SAC track who reviewed the initial papers and the revised versions of this special issue.

Frédéric Loulergue
Northern Arizona University
School of Informatics, Computing and Cyber Systems
Flagstaff, Arizona, USA