Introduction

Pierre Manneback, Thierry Gautier, Gudula Rnger, and Manuel Prieto Matias

Topic chairs

Developing parallel or distributed applications is a hard task and it requires advanced algorithms, realistic modeling, efficient design tools, high-level programming abstractions, high-performance implementations, and experimental evaluation. Ongoing research in this field emphasizes the design and development of correct, high-performance, portable, and scalable parallel programs. Related to these central needs, important work addresses methods for reusability, performance prediction, large-scale deployment, self-adaptivity, and fault-tolerance. Given the rich history in this field, practical applicability of proposed methods, models, algorithms, or techniques is a key requirement for timely research. This topic is focusing on parallel and distributed programming in general, except for work specifically targeting multicore and manycore architectures, which has matured to becoming a Euro-Par topic of its own.

Each submission was reviewed by at least four reviewers and, finally, we were able to select five regular papers, spanning the topics scope, ranging from low-level issues like failure detectors, all the way up to parallelization of a parser.

In particular, Greve et al. in A Failure Detector for Wireless Networks with Unknown Membership propose a protocol for a new class of detector which tolerates mobility and message losses. In Correlated Set Coordination in Fault Tolerant Message Logging Protocols, Bouteiller et al. describe a hierarchical partitioning of a set of processes that takes benefit of a coordinated protocol on each many-core nodes, as well as a message logging protocol for scalability between nodes. Liu et al. contributed "Towards Systematic Parallel Programming over MapReduce", a framework based on list homomorphisms to derive MapReduce programs from sequential specification. In "HOMPI: A Hybrid Programming Framework for Expressing and Deploying Task-Based Parallelism", Dimakopoulos et al. present a framework to exploit cluster of multicores on task-based parallel programs. Last but not least, Cameron et al. present a original parallelization of the XML parser in their paper "Parallel Scanning with Bitstream Addition: An XML Case Study".

We are proud of the scientific program that we managed to assemble. Of course, this was only possible by combining the efforts of many. We would like to take the opportunity to thank the authors who submitted their contributions, and the external referees who have made the scientific selection process possible in the first place.

E. Jeannot, R. Namyst, and J. Roman (Eds.): Euro-Par 2011, LNCS 6853, Part II, p. 1, 2011.