European Conference on Parallel Computing

Topic 9: Parallel Programming: Models, Methods and Languages

Description

This topic provides a forum for the presentation of the latest research results and practical experience in the development of parallel programs. Advances in algorithmic and programming models, design methods, languages, and interfaces are needed to produce correct, portable parallel software with predictable performance on different parallel and distributed architectures.
The topic emphasizes results that improve the process of developing high-performance programs, including high-integrity programs that are scalable with both problem size and complexity. Of particular interest are novel techniques by which parallel software can be assembled from reusable parallel components without compromising efficiency. Related to this is the need for parallel software to adapt, both to available resources and to the problem being solved.
Where appropriate, contributions should demonstrate quantitative performance results in support of their claims, and address applications not adequately handled by well-established approaches.

Focus

- Languages, libraries and interfaces for different parallel programming models (e.g. data-parallelism, task-parallelism, communicating processes, functional, object-oriented, logic, component-based, etc.).
- Implementation and optimization techniques for innovative parallel languages and programming models (e.g. threads, dataflow, tiling, skeletons, declarative languages, and generalised data-parallel approaches, etc.).
- Performance models and their integration into the design of efficient parallel algorithms and programs (e.g. BSP, LogP, CGM, N-half and their alternatives, cost calculi and static performance prediction, profile-driven approaches).
- Parallel programming paradigms and tools, their comparison and integration (e.g. data-parallel vs. task-parallel, coordination programming, performance analysis and debugging).
- Methodological aspects of developing, optimizing and validating parallel programs (formalisms, semantics, specification, design, transformations, verification, etc.).
- Software engineering for parallel and distributed systems (design patterns, portability, robustness, standardization, etc.).
- Systematic approaches and programming models to support effective program development in grid environments.
- Domain-specific parallel libraries and languages (e.g. for simulation, irregular and unstructured meshes, computational geometry, etc.).

Global Chair
Prof. Dr. José C. Cunha
New University of Lisbon
Faculty of Sciences and Technology
Department of Informatics
Monte de Caparica, Portugal
jcc@di.fct.unl.pt

Local Chair
Prof. Dr. Sergei Gorlatch
TU Berlin
Fakultät Elektrotechnik und Informatik
Berlin, Germany
gorlatch@fmi.uni-passau.de

Vice Chair
Dr. Daniel Quinlan
Lawrence Livermore National Laboratory
Center for Applied Scientific Computing
Livermore, CA, USA
dquinlan@llnl.gov

Vice Chair
Prof. Dr. Peter H. Welch
University of Kent
Computing Laboratory
Kent, UK
P.H.Welch@kent.ac.uk