Introduction

Felix Wolf*, Andy D. Pimentel*, Luiz DeRose*, Soonhoi Ha*,
Thilo Kielmann*, and Anna Morajko*

The spread of systems that provide parallelism either “in-the-large” (grid infrastructures, clusters) or “in-the-small” (multi-core chips) creates new opportunities for exploiting parallelism in a wider spectrum of application domains. However, the increasing complexity of parallel and distributed platforms renders the programming, the use, and the management of these systems a costly endeavor that requires advanced expertise and skills. There is therefore an increasing need for powerful support tools and environments that will help end users, application programmers, software engineers and system administrators to manage the increasing complexity of parallel and distributed platforms. This topic aims at bringing together tool designers, developers, and users in order to share novel ideas, concepts, and products covering a wide range of platforms, including homogeneous and heterogeneous multicore architectures. The Program Committee sought high-quality contributions with solid foundations and experimental validations on real systems, and encouraged the submission of new ideas on intelligent monitoring and diagnosis tools and environments which can exploit behavior knowledge to detect programming bugs or performance bottlenecks and help ensure correct and efficient parallel program execution.

This year, twenty-one papers were submitted to Topic 1 of which seven papers were accepted. The accepted papers cover a wide range of interests and contributions:

- Two papers study the performance analysis, profiling, and benchmarking of I/O. The paper “Pattern Matching and I/O Replay for POSIX I/O in Parallel Programs” describes an approach for application-oriented file system benchmarking, which allows the I/O behavior of parallel applications to be tracked and replayed. The paper “An Extensible I/O Performance Analysis Framework for Distributed Environments” presents ExPerT, an Extensible Performance Toolkit. ExPerT defines a flexible framework from which a set of benchmarking, tracing, and profiling applications can be correlated together in a unified interface.
- Two papers address the performance analysis and tuning of parallel applications. The paper “Atune-IL: An Instrumentation Language for Auto-Tuning Parallel Applications” introduces Atune-IL, an instrumentation language that uses new types of code annotations to mark tuning parameters, blocks, permutation regions, or measuring points. The paper “A Holistic Approach towards Automated Performance Analysis and Tuning” aims at improving the productivity of performance debugging of parallel applications. To this end,

---

* Topic Chairs.

it proposes a framework that facilitates the combination of expert knowledge, compiler techniques, and performance research for performance diagnosis and solution discovery.

- The paper “Assigning Blame: Mapping Performance to High Level Parallel Programming Abstractions” discusses mapping mechanisms, called variable blame, for creating mappings of high-level programming abstractions down to the low-level parallel programming constructs and using them to assist in the profiling and debugging of programs created using advanced parallel programming techniques.
- The paper “Process Mapping for Collective Communications” proposes a way to handle collective communications which transforms them into a series of point-to-point communication operations according to their implementation in communication libraries. Hereafter, existing approaches can be used to find optimized mapping schemes which are optimized for both point-to-point and collective communications.
- The paper “Grouping MPI Processes for Partial Checkpoint and co-Migration” studies a partial checkpoint and migrate facility which allows for checkpointing and migrating only a subset of the running MPI processes while the others can continue to execute and make progress.