Topic 1
Support Tools and Environments

Liviu Iftode, Christine Morin, Marios Dikaiakos, and Erich Focht

Topic Chairs

Despite an impressive body of research, parallel and distributed computing remains a complex task prone to subtle software bugs, which can affect both the correctness and the performance of the computation. The increasing demand to distribute computing over large-scale distributed platforms, such as grids and large clusters, overlaps with an increasing pressure to make computing more dependable. To address these challenges, the parallel and distributed computing community continuously requires better tools and environments to design, program, debug, test, tune, and monitor programs. This topic aims to bring together tool designers, developers, and users to share their concerns, ideas, solutions, and products, covering a wide range of platforms.

This year, eighteen submitted papers were reviewed in Topic 1 and eight papers were accepted. The accepted papers can be grouped in three categories, reflecting the diversity of tools needed for parallel and distributed computing.

Three papers relate to profiling tools. The paper "A Profiling Tool for Detecting Cache Critical Data Structures" describes dprof, a tool to visualize cache misses in a multithreaded program. The dprof tool is able to correlate cache miss performance with the program code, providing levels of granularity related to the whole data structures, function calls or individual variables. The paper "On Using Incremental profiling for the Performance Analysis of Shared Memory Parallel Applications" presents an improvement of the ompP profiling tool for OpenMP adding a temporal dimension to profiling data. The paper "Automatic Structure Extraction from MPI Applications Tracefiles" addresses the important issue of analysing huge trace files obtained from large message-passing parallel applications executed on hundreds or thousands of processors in supercomputers or grids. It proposes an automatic reduction technique of trace files for extracting useful patterns.

Two papers describe tools for automatic tuning of parallel applications. The paper "Automatic Generation of Dynamic Tuning Techniques" presents in the framework of the MATE environment a performance problem specification language, which can be used to automatically generate a so-called tunlet to tune the running application for better performance. The paper "Fine Tuning Algorithmic Skeletons" describes a tool for developers to identify the performance bottlenecks through monitoring execution in a skeleton-driven parallel processing runtime (Calcium).

Three papers present scheduling frameworks. The paper "A Scheduling Toolkit for Multiprocessortask Programming with Dependencies" addresses the issue of scheduling M-task applications that exploit data and task parallelism at the same time. The STK scheduling toolkit aims at closing the gap between the specification and the execution of M-task programs by automatically determining valid schedules for parallel target platforms. The paper "Building Portable Thread Schedulers for Hierarchical Multiprocessors: the BubbleSched Framework" presents the design and implementation the BubbleSched framework, whose goal is to ease the development and evaluation of customized user level thread schedulers for large shared-memory NUMA machines. The paper "Makefile::Parallel Dependency Specification Language" describes a tool to schedule and monitor the execution of parallel applications on clusters, which is based on a Makefile-like language to specify dependencies between processes. We would like to thank the panel of reviewers for their precious time and effort in the selection process.