Topic 02
Performance Evaluation and Prediction

Thomas Fahringer and Wolfgang E. Nagel

Topic Chairmen

Even today, when microprocessor vendors announce breakthroughs every other week, performance evaluation is still one of the key issues in parallel computing. One of the observations is that on a single PE, many, if not most applications relevant to the technical field do not benefit adequately from clock rate improvement. The reason for this is memory access: most data read and write operations access memory which is relatively slow compared to processor speed. With several levels of caches we now have complex system architectures which, in principal, provide plenty of options to keep the data as close as possible to the processor. Nevertheless, compiler development proceeds in slow progression, and the single PE still dominates the results achieved on large parallel machines.

In a couple of months, almost every vendor will offer very powerful SMP nodes with impressive system peak performance numbers, based on multiplication of single PE peak performance numbers. These SMP nodes will be coupled to larger systems with even more impressive peak numbers. In contrast, the sustained performance for real applications is far from satisfactory, and a large number of application programmers are struggling with many complex features of modern computer architectures.

This topic aims at bringing together people working in the different fields of performance modeling, evaluation, prediction, measurement, benchmarking, and visualization for parallel and distributed applications and architectures. It covers aspects of techniques, implementations, tools, standardization efforts, and characterization and performance-oriented development of distributed and parallel applications. Especially welcome have been contributions devoted to performance evaluation and prediction of object-oriented and/or multi-threaded programs, novel and heterogeneous architectures, as well as web-based systems and applications. 27 papers were submitted to this workshop, 8 were selected as regular papers and 6 as short papers.

Performance Diagnosis Tools

More intelligent tools which sometimes even work automatically will have a strong impact on the acceptance and use of parallel computers in future. This session is devoted to that research field. The first paper introduces a new technique for automated performance diagnosis using the program's call graph. The implementation is based on a new search strategy and a new dynamic instrumentation to resolve pointer-based dynamic call sites at run-time. The second

A. Bode et al. (Eds.): Euro-Par 2000, LNCS 1900, pp. 105–107, 2000.

paper presents a class library for detecting typical performance problems in event traces of MPI applications. The library is implemented using the powerful high-level trace analysis language EARL and is embedded in the extensible tool component called EXPERT. The third contribution in this session describes Pajé, an interactive visualization tool for displaying the execution of parallel applications where a (potentially) large number of communicating threads of various life-times execute on each node of a distributed memory parallel system.

Performance Prediction and Analysis

The efficient usage of resources – either memory or just processors – should be a prerequisite for all kinds of parallel programming. If multiple users have different requests over time, the scheduling and allocation of resources to jobs becomes a critical issue. This session summarizes contributions in that field. The first paper presents a hybrid version of two previous models that perform analysis of memory hierarchies. It combines the positive features of both models by interleaving the analysis methods. Furthermore, it links the models to provide a more focused method for analyzing performance contributions due to latency hiding techniques such as outstanding misses and speculative execution. The second paper describes the tool set PACE that provides detailed predictive performance information throughout the implementation and execution stages of an application. Because of the relatively fast analysis times, the techniques presented can also be used at run-time to assist in application steering and the efficient management of the available system resources. The last paper addresses the problem of estimating the total execution time of a parallel program-based domain decomposition.

Performance Prediction and Simulation

The third part of the workshop covers a couple of important aspects ranging from performance prediction to cache simulation. The first paper in this session describes a technique for deriving performance models from design patterns expressed in the Unified Modeling Language (UML) notation. The second paper describes the use of an automatic performance analysis tool for describing the behaviour of a parallel application. The third paper proposes a new cost-effective approach for on-the-fly micro-architecture simulations using long running applications. The fourth paper introduces a methodology for a comprehensive examination of workstation cluster performance and proposes a tailored benchmark evaluation tool for clusters. The fifth paper investigates performance prediction for a discrete-event simulation program. The performance analyzer tries to predict what the speedups will be, if a conservative, “super-step” (synchronous) simulation protocol is used. The last paper in this session focuses on cache memories, cache miss equations, and sampling.

Performance Modeling of Distributed Systems

Performance analysis and optimization is even more difficult if the environment is physically distributed and possibly heterogeneous. The first paper studies the influence of process mapping on message passing performance on Cray T3E and the Origin 2000. First, the authors have designed an experiment where processes are paired off in a random manner and messages of different sizes are exchanged between them. Thereafter, they have developed a mapping algorithm for the T3E, suited to n-dimensional cartesian topologies. The second paper focuses on heterogeneity in Networks of Workstations (NoW). The authors have developed a performance prediction tool called ChronosMix, which can predict the execution time of a distributed algorithm on parallel or distributed architecture.