05501 Abstracts Collection
Automatic Performance Analysis
— Dagstuhl Seminar —

Hans Michael Gerndt[1], Allen Malony[2], Barton P. Miller[3] and Wolfgang Nagel[4]

[1] TU München, DE
gerndt@in.tum.de
[2] University of Oregon, US
malony@cs.uoregon.edu
[3] Univ. Wisconsin - Madison, US
bart@cs.wisc.edu
[4] TU Dresden, DE
nagel@zhr.tu-dresden.de

Abstract. From 12.12.05 to 16.12.05, the Dagstuhl Seminar 05501 "Automatic Performance Analysis" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Parallel computing, performance analysis, programming environments

05501 Summary – Automatic Performance Analysis

The Workshop on Automatic Performance Analysis (WAPA 2005, Dagstuhl Seminar 05501), held December 13-16, 2005, brought together performance researchers, developers, and practitioners with the goal of better understanding the methods, techniques, and tools that are needed for the automation of performance analysis for high performance computing.

Keywords: Automatic Performance Analysis, Parallel Computers, Performance Tuning

Joint work of: Gerndt, Michael; Malony, Allen; Miller, Barton; Nagel, Wolfgang

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/506

Zero-cost Reliability for Scalable Tree-based Overlay Network

Dorian Arnold (University of Wisconsin - Madison, USA)

With trends toward large-scale high-performance computing systems, tree-based overlay networks (TBŌNs) have emerged as an important network topology for scalable distributed tools and applications. This talk focuses on novel reliability techniques for TBŌNs that require no additional storage or computational overhead during failure-free executions.

We develop new, relaxed models of distributed consistency that allow computational states lost due to failure to be replaced by potentially non-identical computational states with little or no effect on the output of the computation. We leverage these new consistency models to develop state regeneration mechanisms that use computational state from non-failed TBŌN processes to regenerate compensatory states for failed ones.

In this talk, we present our new distributed consistency models and state regeneration mechanisms. We prove sufficient properties of distributed computations that lead to the availability of these mechanisms and show how to use these properties to derive the computation-specific recovery operations. We also show that these concepts are applicable to large classes of TBŌN computations, e.g. the data aggregation and reduction operations generally found in automated performance analysis tools.

Keywords: Reliable data reduction, Scalability, Tree-based computing

Joint work of: Arnold, Dorian; Miller, Barton

Data Profiling for Cache Optimizations on SMP Clusters

Thomas Brandes (Fraunhofer Institut - St. Augustin, D)

Profiling helps to understand the dynamic behavior of an application, to detect performance problems, and to find out where it can be made faster.

Identification and understanding of bottlenecks in the program due to cache problems is one of the most critical issues. Current profiling tools can access hardware performance counters and can give information about cache misses but this information is usually not very helpful to find out what has caused the misses and how they can be avoided. Data profiling is a mechanism that does not only count misses but also determines the data addresses having caused them.

The talk presents a tool that supports data profiling for serial and parallel Fortran programs using MPI and/or OpenMP. At runtime, data address sampling is enabled that provides information that can be used to reconstruct data addresses that have caused a cache miss. Due to a source-code instrumentation data addresses can be related back to data structures and regions of the program.

By this approach the user gets more useful information that might help him to improve his program for a better cache usage. Some examples and techniques will be presented to show the effectiveness of the approach.

Keywords: Data Profiling, Address Sampling, Performance Analysis, Parallel Programming

Integrative Concepts for Scalable Distributed Program Analysis

Holger Brunst (TU Dresden, D)

The detailed performance analysis of highly parallel software processes on the latest generation of large scale parallel computers represents a challenge with many unresolved problems. Even with modern monitoring systems which allow for data preselection and automatic filtering of some redundant data, an extraordinary amount of data needs to be handled for realistic application runs. Storage, access, preparation, and visualization of the data cannot be carried out with conventional methods anymore.

This talk presents novell concepts for scalable distributed performance visualization of parallel programs which are suitable for the next generation of high performance computers with more than 100,000 processors. The applied parallel data processing model shows that innovative usage of distributed resources combined with new event data processing methods can be used to obtain insights from data volumes thus far, could not be handled.

Seamlessly integrating this architecture into production environments eliminates the highly restrictive transport of large performance data volumes between shared production resources and private desktop systems. Experiances with large scale systems like the ASC(I) machines or the Earth-Simulator are presented and discussed.

Keywords: Program Analysis, Performance Tuning, Tracing, Client/Server Architecture, Scalability

Vertical Performance Monitoring

Calin Cascaval (IBM TJ Watson Research Center, USA)

Moore's law has allowed us to build layers upon layers of software without paying too much attention to system performance. As performance advances achievable through chip fabrication technology are reaching their limits, most of the system performance must come from software. In this respect, optimizations of software layers can no longer can be done in isolation.

In this talk we will present our approach to vertical performance monitoring. We have designed and implemented a performance and environment monitoring

(PEM) infrastructure that vertically integrates performance events from various layers in the execution stack. We show how we used this framework to characterize program behavior. We also discuss how this infrastructure can be used in different optimization scenarios, for either static or dynamic compilers.

Keywords: Program optimization, performance monitoring

Development and Tuning Framework of Master/Worker Applications

Paola Caymes–Scutari (Universitat Autònoma de Barcelona, E)

Parallel/distributed programming is a complex task that requires a high degree of expertise to fulfill the expectations of high performance computation. The Master/Worker paradigm is one of the most commonly used because it is easy to understand and there is a wide range of applications that match this paradigm. However, there are certain features, such as data distribution and the number of workers that must be tuned properly to obtain adequate performance. In most cases such features cannot be tuned statically since they depend on the particular conditions of each execution. In this paper, we show a dynamic tuning environment that is based on a theoretical model of Master/Worker behavior and allows for the adaptation of such applications to the dynamic conditions of execution. The environment includes a pattern based application development framework that allows the user to concentrate on the design phase and makes it easier to overcome performance bottlenecks.

Keywords: Dynamic tuning, performance analysis, performance model

Joint work of: Caymes Scutari, Paola; Morajko, Anna; César, Eduardo; Costa, Genaro; Mesa, José; Margalef, Tomàs; Sorribes, Joan; Luque, Emilio

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/505

See also: Journal of Computer Science and Technology, vol. 5, n. 3, pp. 115-120

Active Harmony: Parallel Automated Tuning of Parallel Programs

Jeff Hollingsworth (University of Maryland - College Park, USA)

Active Harmony is an automated runtime performance tuning system. I will describe how the system can be applied to different application domains including a web E-Commerce system and several scientific applications. Using the system we have been able to achieve performance gains of up to a factor of three for real applications.

I will also describe how we have been working to create a parallel search process that allows groups nodes in a parallel computation to explore different parameters combinations at once.

Keywords: Automated tuning, parallel programming, performance tools

Comparative Performance Analysis Using PerfTrack

Karen L. Karavanic (Portland State University, USA)

PerfTrack is a data store and interface for managing performance data from large-scale parallel applications. Data collected in different locations and formats can be compared and viewed in a single performance analysis session. The underlying data store used in PerfTrack is implemented with a database management system (DBMS). PerfTrack includes interfaces to the data store and scripts for automatically collecting data describing each experiment, such as build and platform details. We have implemented a prototype of PerfTrack that can use Oracle or PostgreSQL for the data store. We demonstrate the prototype's functionality with two case studies: one is a parameter study conducted at Lawrence Livermore National Laboratory (LLNL) on two high end platforms, a 128 node cluster of IBM Power 4 processors and BlueGene/L; the second demonstrates incorporating performance data from the Paradyn Parallel Performance Tool into an existing PerfTrack data store.

Keywords: Parallel performance diagnosis data integration

Joint work of: Karavanic, Karen L.; May, John; Mohror, Kathryn; Miller, Brian; Knapp, Rashawn; Pugh, Brian

Observing and Controlling: The Case for Organic Computing Systems

Wolfgang Karl (Universität Karlsruhe, D)

As future high performance architectures, especially for embedded systems, are going to more and more exploit parallelism they also have to be energy-aware and reliable. Additionally, they also will be integrated in a highly dynamical environment with permanently changing situations. Thus, future system architectures have to be highly adaptable, thus having self-organizing and organic characteristics. A key challenge hereby is to permanently get information about the system status in order to be aware of the available resources, their performance and energy consumption. Thus, a new sophisticated monitoring approach is necessary. The talk presents such a flexible monitoring approach for high-performance architectures with self-organizing and organic characteristics.

Automatic Performance Analysis for Cache Architectures in SMPs

Edmond Kereku (TU München, D)

We present an automatic monitoring system consisting of a monitoring infrastructure and an automatic performance analyzer. The monitoring infrastructure supports different monitoring resources (CPU counters, simulation) and monitors the utilization of cache hierarchies in serial and OpenMP programs. A special feature of our system is the restriction of monitoring to single data structures. Our Apart Specification Language (ASL)-based automatic analyzer is able to search for predefined performance bottlenecks in code regions using a provided set of search and refinement strategies.

Keywords: Automatic Performance Analysis, AMEBA, Selective Monitoring, MRI, ASL, Instrumentation, Simulation

Program Trace Analysis based on Compressed Complete Call Graphs

Andreas Knüpfer (TU Dresden, D)

The tree-like Complete Call Graph (CCG) data structure is proposed as an alternative to the common linear storage schemes. By transparent in-memory compression CCGs are capable of exploiting redundancy as frequently found in traces and thus reduce the memory requirements notably. Evaluation algorithms can be designed to take advantage of CCGs, such that the computational effort is reduced in the same order of magnitude as the memory requirements.

Keywords: Program traces, data structures, redundancy, compression, evaluation

Remote Performance Monitor (RPM)

Chandra Krintz (Univ. California - Santa Barbara, USA)

Mobile, resource-constrained, battery-powered devices have emerged as key access points to the world's digital infrastructure. To enable our understanding of the performance of these devices, we must be able to efficiently collect accurate profile data from these devices after they are deployed in the field. Moreover, understanding the full-system power and energy behavior of these systems for real programs is vital if users are to accurately characterize, model, and develop effective techniques for extending battery life. Unfortunately, extant approaches to measuring and characterizing power and energy consumption focus on high-end

processors, do not consider the complete device, employ inaccurate (program-only) simulation, rely on inaccurate, course-grained battery level data from the device, or employ expensive power measurement tools that are difficult to share across research groups and students. To address these issues, we developed remote performance monitor (RPM).

The first component of RPM is an efficient technique for collecting accurate sample-based program profiles. The key to the efficacy of this technique is that we identify when to sample using the repeating patterns in program execution, phases. To enable fine-grained, full-system characterization of embedded computers, we couple and unify phase-aware profiling, hardware performance monitoring, and power and energy measurement within RPM. RPM consists of a tightly coupled set of components which (1) control lab equipment for power measurements and analysis, (2) configure target system characteristics at runtime (such as CPU and memory bus speed), (3) collect target system data using on-board hardware performance monitors (HPMs) and (4) provide a remote access interface.

Users of RPM can submit and configure experiments that execute programs on the RPM target device (currently a Stargate sensor platform that is very similar to an HP iPAQ) to collect very accurate power, energy, and CPU performance data with high resolution.

Keywords: Profiling, hardware performance monitors, sampling, phase behavior, power, energy

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/504

Joint work of: Krintz, Chandra; Gurun, Selim

Model-Bases Parallel Performance Diagnosis

Allen Malony (University of Oregon, USA)

Scientific parallel programs often undergo significant performance tuning before meeting performance expectations. Performance tuning naturally involves a diagnosis process – locating performance bugs that make a program inefficient and explaining them in terms of high-level program design. We present a systematic approach to generating performance knowledge for automatically diagnosing parallel programs. Our approach exploits program semantics and parallelism embedded in computational models to search and explain bugs. We first identify categories of expert knowledge required for performance diagnosis and describe how to extract the knowledge from computational models. Second, we represent the knowledge in such a way that diagnosis can be carried out in an automatic manner. We also address adaptability to program implementation variations in the knowledge representation.

Finally, we demonstrate the effectiveness of our knowledge engineering approach through two case studies. Our experience diagnosing Master-Worker and Wavefront programs show that model-based performance knowledge can provide effective guidance for locating and explaining performance bugs at a high level of program abstraction.

Keywords: Parallel, performance, diagnosis, automatic, model

The PerfTrack Tool for Performance Data Management

Kathryn Mohror (Portland State University, USA)

PerfTrack is a data store and interface for managing performance data from large-scale parallel applications. Data collected in different locations and formats can be compared and viewed in a single performance analysis session. The underlying data store used in PerfTrack is implemented with a database management system (DBMS). PerfTrack includes interfaces to the data store and scripts for automatically collecting data describing each experiment, such as build and platform details. In this demonstration, we describe the current state of the PerfTrack prototype and demonstrate its use with data from a set of executions of the ASCI Purple Benchmark, UMT2K.

Joint work of: Mohror, Kathryn; Karavanic, Karen L.; May, John; Miller, Brian; Knapp, Rashawn; Pugh, Brian; Spencer, Travis; Wilson, Eric

The Automatic Performance Analysis Toolset KOJAK

Bernd Mohr (Forschungszentrum Jülich, D)

The KOJAK performance-analysis environment provides a complete tracing-based solution for automatic performance analysis of MPI, OpenMP, SHMEM, or hybrid applications running on parallel computers. KOJAK's analysis tool EXPERT describes performance problems using a high level of abstraction in terms of execution patterns that result from an inefficient use of the underlying programming model(s). The analysis is carried out along three interconnected dimensions: class of performance behavior, call tree, and thread of execution. In the result presentation tool CUBE, each dimension is arranged in a hierarchy so that the user can investigate the behavior on varying levels of detail. All three dimensions are interactively accessible using a single integrated view.

Keywords: Performance analysis, parallel computing, event tracing

Joint work of: Mohr, Bernd; Wolf, Felix; Wylie, Brian

Full Paper: http://www.fz-juelich.de/zam/kojak/

On-line Automated Performance Diagnosis on Thousands of Processes

Philip Roth (Oak Ridge National Lab., USA)

Performance diagnosis tools are critical for the effective use of large parallel computing resources, but existing tools have failed to address three problems that limit their scalability: (1) management and processing of the volume of performance data generated when monitoring a large number of processes, (2) communication between a large number of tool components, and (3) presentation of performance data and analysis results for programs with a large number of processes. In this talk, we present our approach for finding performance problems in programs with a large number of processes to address these three performance tool scalability barriers.

First, we show how to design a scalable, distributed performance diagnosis facility. We demonstrate our design with an on-line automated strategy for finding performance bottlenecks that leverages our distributed multicast and data aggregation infrastructure. Our bottleneck search strategy uses distributed, independent bottleneck search agents located in the tool agent processes that monitor application processes. Second, we present a technique called the Sub-Graph Folding Algorithm for constructing compact displays of bottleneck detection strategy results. Our visualization technique presents the results of a bottleneck search using dynamic graphs that record the refinements of a search. Graph complexity is controlled by combining sub-graphs representing processes with similar qualitative behavior into a single composite sub-graph.

Using an approach that combines these two synergistic parts, we performed bottleneck searches on programs with up to 1024 processes with no sign of tool resource saturation. Also, with 1024 application processes the Sub-Graph Folding Algorithm reduced a bottleneck search results graph with over 30,000 nodes to a 44-node graph showing the same qualitative performance information as the original graph.

Keywords: Scalability, performance diagnosis, tools, automation, visualization, Paradyn

Joint work of: Roth, Philip; Miller, Barton; Arnold, Dorian

Perspectives on Performance Analysis and Prediction for Next-Generation Architectures

Philip Roth (Oak Ridge National Lab., USA)

The Future Technologies Group at Oak Ridge National Laboratory explores performance analysis, evaluation, and prediction of architectures and algorithms in support of scientific computation. We seek to identify core technologies that will

satisfy the computation, communication, memory, and storage demands of future petascale applications. Toward this end, we maintain the Experimental Computing Laboratory (ExCL) for evaluating technologies such as Field Programmable Gate Arrays (FPGAs), array processors, optical processors, homogeneous and heterogeneous multi-core processors, and non-traditional programming models.

In this talk, we present our group's perspectives on modeling and predicting the performance of systems that incorporate emerging technologies. We perceive the challenges for performance prediction and analysis with emerging technologies are increasing in scale, architectural complexity, and application complexity. As a case study, we describe our work modeling the performance of the AMBER molecular dynamics code, overcoming barriers that limited scalability to 128 processors, and accelerating a portion of the AMBER code using FPGAs. To combat the challenges of performance prediction on emerging architectures, we also introduce Modeling Assertions, a technique that embeds an analytic performance model into the programs source code using symbolic expressions. With run-time support, these symbolic expressions can be verified and composed into an overall application model as the application runs.

Keywords: Performance modeling, performance analysis, performance prediction, emerging architectures

Joint work of: Vetter, Jeffrey; Roth, Philip; Alam, Sadaf

Open|SpeedShop - Open Source Performance Analysis for MPI Clusters (Demonstration)

Martin Schulz (LLNL - Livermore, USA)

Open|SpeedShop is a scalable open-source performance framework for Linux clusters. It is based on dynamic instrumentation using DynInst/DPCL to insert performance probes into binary applications (both sequential and parallel) and to retrieve performance data. Users can control this process and view the resulting data using a comprehensive graphical user interface, a batch command language, and a Python Interface.

Keywords: Parallel performance analysis, dynamic instrumentation, open source tools, Linux cluster

Automatic model generation for performance prediction

Martin Schulz (LLNL - Livermore, USA)

Accurately modeling and predicting performance for large-scale applications becomes increasingly difficult as system complexity scales dramatically. Analytic predictive models are useful, but are difficult to construct, usually limited in scope, and often fail to capture subtle interactions between architecture and software. In contrast, we employ multilayer neural networks trained on input data from executions on the target platform.

This approach is useful for predicting many aspects of performance, and it captures full system complexity.

Keywords: Performance modeling, neural networks, parameter studies, automatic model generation

Joint work of: Schulz, Martin; Ipek, Engin; de Supinski, Bronis; Singh, Karan; McKee, Sally A.; Caruana, Rich

Performance Analysis of Grid Workflows in K-WfGrid and ASKALON

Hong–Linh Truong (Universität Innsbruck, A)

The complexity and quantity of Grid workflows are so overwhelming that new performance analysis techniques are required to instrument multilingual workflow-based applications and to select, measure and analyze various performance metrics at different levels of abstraction. Moreover, Grid workflow performance analysis tools have to support Grids based on service-oriented architecture (SOA) and to address well-known issues of the Grid computing such as the diversity, dynamics and scalability.

In this talk, we present methods and techniques that are being applied to the performance monitoring and analysis of Grid workflows in the K-WfGrid project and the ASKALON toolkit. Ontology is used for describing performance data of Grid workflows and performance data is associated with multiple levels of abstraction. Various instrumentation techniques are utilized to support multilingual and service-based Grid workflows. A novel overhead classification for Grid workflows is introduced and the performance analysis is conducted in a distributed fashion. Performance monitoring and analysis components are implemented as P2P-based Grid services. All performance data, monitoring and analysis requests are described in XML, thus simplifying the interaction and integration between various instrumentation, monitoring and analysis services and increasing the interoperability of the performance tools.

Keywords: Grid performance monitoring and analysis, Grid workflows, performance ontology

Joint work of: Truong, Hong–Linh; Fahringer, Thomas

Perspectives on Performance Analysis and Prediction for Next-Generation Architectures

Jeffrey Vetter (Oak Ridge National Lab., USA)

The Future Technologies Group at Oak Ridge National Laboratory explores performance analysis, evaluation, and prediction of architectures and algorithms in support of scientific computation.

We seek to identify core technologies that will satisfy the computation, communication, memory, and storage demands of future petascale applications. Toward this end, we maintain the Experimental Computing Laboratory (ExCL) for evaluating technologies such as Field Programmable Gate Arrays (FPGAs), array processors, optical processors, homogeneous and heterogeneous multi-core processors, and non-traditional programming models.

In this talk, we present our group's perspectives on modeling and predicting the performance of systems that incorporate emerging technologies. We perceive the challenges for performance prediction and analysis with emerging technologies are increasing in scale, architectural complexity, and application complexity. As a case study, we describe our work modeling the performance of the AMBER molecular dynamics code, overcoming barriers that limited scalability to 128 processors, and accelerating a portion of the AMBER code using FPGAs. To combat the challenges of performance prediction on emerging architectures, we also introduce Modeling Assertions, a technique that embeds an analytic performance model into the programs source code using symbolic expressions. With run-time support, these symbolic expressions can be verified and composed into an overall application model as the application runs.

Keywords: Performance modeling, performance analysis, performance prediction, emerging architectures

Joint work of: Vetter, Jeffrey; Roth, Philip; Alam, Sadaf

Scalable Automatic Trace Analysis

Felix Wolf (Forschungszentrum Jülich, D)

A powerful and widely-used method for analyzing the performance behavior of parallel programs is event tracing. While event tracing enables the identification of performance problems at a high level of detail, escalating trace-file size often constrains its scalability on large-scale systems. Automatic trace analysis based on pattern recognition, as implemented in KOJAK, has addressed this problem to some extent, in that it relieves the user from the burden of searching large amounts of data manually. With the goal of extending KOJAK's approach to present and future architectures with very large processor numbers, this talk presents a new preliminary parallel design of KOJAK's trace analyzer that utilizes the entire allocation of processors reserved to execute the target application.

Keywords: Parallel computing, performance analysis, event tracing, scalability

Holistic hardware counter performance analysis of parallel programs

Brian J. N. Wylie (Forschungszentrum Jülich, D)

The KOJAK toolkit has been augmented with refined hardware performance counter support, including more convenient measurement specification, additional metric derivations and hierarchical structuring, and an extended algebra for integrating multiple experiments.

Comprehensive automated analysis of a hybrid OpenMP/MPI parallel program, the ASC Purple sPPM benchmark, is demonstrated with performance experiments on equisized POWER4-II-based IBM Regatta p690+ cluster, Opteron-based Cray XD1 cluster and UltraSPARC-IV-based Sun Fire E25000 systems. Automatically assessed communication and synchronisation performance properties, combined with a rich set of measured and derived counter metrics, provide a holistic analysis context and facilitate multi-platform comparison.

Keywords: Parallel program execution tracing, hardware performance counters, automated structured analysis, performance comparison

Joint work of: Wylie, Brian J. N.; Mohr, Bernd; Wolf, Felix

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/507

Full Paper: http://www.fz-juelich.de/zam/kojak/pubs/ib-2005-14.pdf