ELSEVIER

Future Generation Computer Systems 18 (2001) 127–130

FGCS
FUTURE GENERATION COMPUTER SYSTEMS

www.elsevier.com/locate/future

Editorial

Performance data mining
Automated diagnosis, adaption, and optimization

Keywords: Performance data mining; Parallel; Diagnosis; Adaption; Optimization

1. Introduction

The development of performance measurement and analysis techniques and tools for high-performance parallel and distributed systems has made it possible to capture a wealth of data about application and system performance behavior. These data embody the effects of interacting, performance factors found in the program, its algorithms, the architecture and hardware, and the system software, whose interdependent performance relationships grow ever more complex as the computing environment increases in sophistication. Nevertheless, the user is still, for the most part, placed in the central decision-making role in the use of the techniques/tools, the interpretation of the resulting performance information, and the guidance for program or system modification.

Recent work has sought to move human decision-making out of the performance measurement–diagnosis–optimization loop by employing "intelligent" methods based on automated performance measurement, knowledge-based diagnosis frameworks, online, adaptive performance control, and predictive performance models built from detailed empirical analysis. The term "performance data mining" is used to characterize this work.

Already back in 1997 in association with the International Conference on Supercomputing, we organized a workshop on performance data mining for high-performance parallel and distributed systems. To the best of our knowledge, this was the first scientific event dedicated to the exploitation of data mining techniques in the field of performance evaluation. The papers enclosed in this volume have their roots in this pioneering workshop. The topics discussed reflect different perspectives on the performance data mining research question and on the focus for effective strategies and techniques.

In the following sections, we first discuss our views on the concept of performance data mining. We then introduce each paper and comment on how we see the ideas presented contributing to the problem understanding and solution. We conclude with thoughts for future research.

2. Performance data mining focus

Performance evaluation for high-performance systems is commonly regarded as a *performance debugging* process of measurement, bottleneck diagnosis, and optimization. If we consider a single parallel program and the set of all possible executions of the program, as determined by choice of execution environment, program transformation, and input data set, we can regard the *performance data space* for the program as the performance characterization of every program execution instance in the set. It would be relatively easy to solve the performance debug-

0167-739X/01/$ – see front matter
PII: S0167-739X(01)00047-4

ging problem for the program if the performance data space existed — simply find the best combination of parameters that optimizes the desired performance criteria. Unfortunately, this hypothetical space is only ever sparsely populated and new performance data is almost always generated on an as needed basis. Because the performance space is large and broad, multidimensional and multivariated, and varying in resolution and accuracy, effective performance debugging must include aspects of knowledge-based experimentation, performance modeling, and predictive search.

In the field of data mining and knowledge discovery in databases (KDD), processes and techniques have been developed to work with complex data sets. Data mining considers computationally efficient algorithms that detect and enumerate patterns in data based on domain semantics. Patterns selection is optimized to fit models as a first step towards inferred knowledge. KDD uses data mining techniques to extract "interesting" patterns and "useful" knowledge from data. Patterns are considered with respect to understandability and utility — the goal of *interestingness* is desired. Knowledge is domain dependent and utility directed. With respect to the data mining/KDD process, the emphasis is on data abstraction (patterns), modeling (assigning meaning to patterns), and refinement (model evaluation and knowledge abstraction). The process also implicitly assumes data availability.

The original focus of this FGCS special issue was to consider the performance evaluation process as a data mining and knowledge discovery process. There are several reasons why this focus seemed to make sense. First, the volume and complexity of the performance data space (if it existed) requires methods to abstract the data into more understandable and meaningful forms. Secondly, finding interesting features in performance data and interpreting their importance (so-called model-based performance data analysis) is domain dependent, as in data mining/KDD. A third reason is found in the notion of refinement. Although the performance data space is not fully enumerated, the idea of model evaluation leading to new selection for data mining methods and new knowledge abstraction has its corollary in how the performance evaluation process is controlled. Incompleteness of the performance data space requires search for effective performance experiments. Performance experiments involve a tradeoff in terms of choice of evaluation alternative vs. cost and intrusion vs. value to diagnosis and tuning.

Our intent in titling this special issue *Performance Data Mining* was to emphasize the view of performance evaluation for high-performance computer systems as a process that is knowledge-based and that makes guided decisions in experimentation and in predictive hypotheses during performance debugging and tuning. We already subtitled the ICS workshop *Automated Diagnosis, Adaption, and Optimization* to highlight the new research ideas we envisioned the participants would present.

3. Contributed papers

Each of the five papers presented in this special issue takes a unique perspective on the performance data mining theme.

3.1. Performance optimization of distributed applications in an extensible, adaptive environment, A. Bakić, M. Mutka, D. Rover, A. Waheed

The paper by Bakić et al. focuses on the problem of how to build systems for performance optimization, incorporating measurement, analysis, and visualization that can address the domain specific needs of different systems and application performance scenarios. Their approach highlights tool and environment integration technologies that can work together to achieve a flexible, retargetable framework for prototyping. Bakić et al. have developed the PG^{RT} environment for performance instrumentation and visualization using a tool integration system that features support for software module development and interoperation. In addition to a detailed discussion of the PG^{RT} environment, two examples are given of its use: an adaptive, distributed multimedia server, and a simulated networked system with real-time communicating tasks. Tool integration and adaptability is very important for keeping pace with evolving parallel and distributed applications and platforms. The paper concludes with a discussion on future PG^{RT} project directions.

3.2. Performance analysis of an HPF-like compiler, M. Calzarossa, L. Massari, D. Tessera

The exploitation of data parallelism like in HPF + codes is the intriguing performance analysis challenge of preprocessors and parallelizing HPF compilers. The communication overheads induced by data and work distribution policies imposed by the compilers as a consequence to source code HPF directives provided by programmers can be overwhelming and very complex to analyze. Calzarossa et al. use systematic mining techniques to analyze the cost of parallelization strategies in the VFC (Vienna Fortran Compilation System) and relate the observed performance effects back to HPF source code. Their experiments with HPF + kernels for finite element solvers on a massively parallel execution platform suggest the integration of run-time and post-mortem performance analysis tools with compile-time analyzers to successfully support the development of high performance data parallel codes.

3.3. Distributed simulation performance data mining, A. Ferscha, J. Johnson, S. Turner

Distributed simulation is well known as a particularly challenging performance analysis domain. The distributed synchronization protocols employed to preserve global causality constraints among events that occur locally exhibit performance behavior that is influenced by a prohibitive manifold of factors. This paper demonstrates how — by a 2^k full factorial design as a performance data mining setup — distributed simulation processes can be predicted in their performance behavior. With this approach, besides a performance ranking among the different distributed simulation protocols (conservative and optimistic strategies), also a quantification of factor influence becomes possible. Counterintuitive performance phenomena are discovered for the interaction of factors and help to prevent from inadequate performance engineering decisions made in the early phases of the implementation process of distributed simulation software. The performance data mining strategy proposed here represents a very general method for the performance oriented development of parallel codes or distributed programs.

3.4. The autopilot performance-directed adaptive control system, R. Ribler, D. Reed

In this paper, Ribler and Reed put forth the view of a dynamic, adaptive performance infrastructure that can measure and analyze performance data on the fly, and identify performance behaviors that could help guide resource management and optimization decisions in real time. In response to an increase in next-generation applications that are irregular, have data dependent execution behavior, and generate time varying resource demands, Ribler and Reed discuss the development of the Autopilot toolkit for real-time adaptive control of distributed and parallel computations. The architectural design and components are described, as is the implementation issues that governed the first generation prototype. The implementation of Autopilot's decision control procedures using fuzzy logic is also discussed and offers some interesting extensions of the performance data mining theme to consider. Ribler and Reed present a parallel input/output example as demonstration of the veracity of their approach. Their paper concludes with prospective plans for future work.

3.5. A theory and architecture for automating performance diagnosis, A. Malony, R. Helm

Performance diagnosis is the process of finding and explaining sources of performance problems in programs. Malony and Helm look at performance diagnosis in the context of parallel programs. Their work suggests that difficulties encountered in prior research in this area stem for a lack of a formal theory of how expert programmers do performance diagnosis. Capturing expert knowledge about how to find and correct parallel performance bugs is a challenge. Malony and Helm apply general models of diagnostic problem-solving to help in classifying and representing this knowledge to improve the performance diagnosis process. In addition, this formulation aids in defining how performance tools for measurement, analysis, and presentation should be used in support of evolving diagnosis requirements. The goal is to support a performance diagnosis system that is both adaptable and automated. The Poirot architecture is proposed by Malony and Helm as a framework for building next-generation performance

diagnosis systems. Poirot combine a problem-solver that decides on diagnostic methods to pursue based on currently known performance evidence and hypothesis, and an environment interface that interacts with available performance tools. The paper concludes with a discussion of how well the goal of adaptability and automation can be achieved by performance diagnosis systems in practice.

4. Future work

Several of the ideas presented in the papers and discussed here have been actively pursued in the field in the last years since the workshop was held. This is particularly true with respect to adaptive performance analysis for heterogeneous high-performance computing systems, also known as metacomputing or grid-computing systems. The inherent performance variability (and instability) in these systems due to multiple users competing for resources and dynamic system behavior requires some support for online mining of performance data generated during execution. Not only does the performance feedback control mechanisms need to be provided, but it requires the integration of models for performance behavior of an application in different system contexts and the automation of runtime diagnosis and tuning response.

Indeed, as new high-performance computing platforms are employed, the performance data space continues to change and become more complex, demanding new flexibility in performance analysis tools and more automation in performance diagnosis. It is clear that user-centric analysis in the traditional execute–measure–modify program development process is limited by the user's ability to both interpret the data and to make informed decisions about performance problems and profitable tuning. In recognition of this, an Esprit Working Group on Automatic Performance Analysis: Resources and Tools (APART) has been established to try to identify requirements for automatic performance analysis support, to develop knowledge about typical performance bottlenecks, and to explore promising base implementation technologies. Efforts such as these to improve automated diagnosis, adaptation, and optimization in complex performance data spaces will have an important impact in high-performance computing practice in the future.

Alois Ferscha
Institut für Praktische Informatik, Universität Linz
Altenberger Strasse 69, 4040 Linz, Austria
Corresponding author. Tel.: +43-732-2468-8555
fax: +43-732-2468-8426
E-mail address: ferscha@soft.uni-linz.ac.at
(A. Ferscha)

Allen D. Malony
Department of Computer and Information Science
University of Oregon, Eugene, OR 97403, USA
E-mail address: malony@cs.uoregon.edu
(A.D. Malony)