Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Journal of Systems Architecture 49 (2003) 421–439
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

Automatic performance analysis of hybrid MPI/OpenMP applications

Felix Wolf [a,*,1], Bernd Mohr [b]

[a] Innovative Computing Laboratory, University of Tennessee, 1122 Volunteer Boulevard, Suite 413, Knoxville, TN 37996-3450, USA
[b] Zentralinstitut für Angewandte Mathematik, Forschungszentrum Jülich, 52425 Jülich, Germany

Abstract

The EXPERT performance-analysis environment provides a complete tracing-based solution for automatic performance analysis of MPI, OpenMP, or hybrid applications running on parallel computers with SMP nodes. 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 set of predefined problems can be extended to meet application-specific needs. The analysis is carried out along three interconnected dimensions: class of performance behavior, call tree, and thread of execution. 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; User interface

1. Introduction

Coupling SMP systems combines the packaging efficiencies of shared-memory multiprocessors with the scaling advantages of distributed-memory architectures. The result is a computer architecture that can scale more cost-effectively in size. Unfortunately, these systems come at the price of a more complex programming environment to deal with the different modes of parallel execution: shared-memory multithreading vs. distributed-memory message passing. As a consequence, performance optimization becomes more difficult and creates a need for advanced performance tools that are custom made for this class of computing environments. While performance tools exist for shared-memory systems and for distributed-memory systems, solving performance problems on parallel computers with SMP nodes is not as simple as combining two tools. When dealing with hybrid (MPI/OpenMP) parallel executions, performance problems arise where an integrated view is required. Current state-of-the-art tools such as VGV [1] can provide such an integrated view including the necessary monitoring capabilities, but suffer from performance-information overload, unable to abstract performance problems from detailed performance data in an integrated hybrid framework.

* Corresponding author. Tel.: +1-865-974-8992; fax: +1-865-974-8296.
E-mail addresses: fwolf@cs.utk.edu (F. Wolf), b.mohr@fz-juelich.de (B. Mohr).
[1] This work was done while Felix Wolf was a Ph.D. student at Forschungszentrum Jülich.

1383-7621/$ - see front matter
doi:10.1016/S1383-7621(03)00102-4

The EXPERT performance-analysis environment [2] is able to automatically detect performance problems in event traces of MPI [5], OpenMP [6], or hybrid applications running on parallel computers with SMP nodes as well as on more traditional non-SMP or single SMP systems. Performance problems are represented as execution patterns that correspond to situations of inefficient behavior. These patterns are specified as compound events which are input for an automatic analysis process that recognizes and quantifies the inefficient behavior in event traces. Mechanisms that hide the complex relationships within compound-event specifications allow a simple description of complex inefficient behavior on a high level of abstraction. In addition, the set of predefined performance problems can bed extended to meet individual (e.g., application-specific) needs.

Like Paradyn [7], which searches for performance problems along different program-resource hierarchies including the call graph [8], EXPERT takes advantage of decomposing the search space into multiple hierarchical dimensions. The analysis process of EXPERT automatically transforms the event traces into a three-dimensional representation of performance behavior. The first dimension is the kind of behavior. The second dimension is the call tree and describes the behavior's source-code location and the execution phase during which it occurs. Finally, the third dimension gives information on the distribution of performance losses across different processes or threads. The hierarchical organization of each dimension enables the investigation of performance behavior on varying levels of granularity. Each point of the representation is uniformly mapped onto the corresponding fraction of execution time, allowing the convenient correlation of different behavior using only a single view. The user can interactively access all the hierarchies constituting a dimension of performance behavior using standard tree browsers.

[2] The work on EXPERT is carried out as a part of the KOJAK project [2,3] and is embedded in the IST working group APART [4].

The remainder of this article is organized as follows: First, we consider related work in Section 2. Then, we describe the overall architecture of our analysis environment in Section 3. In Section 4, we present the abstraction mechanisms used to simplify the specification of complex situations representing inefficient performance behavior. After that, we introduce the actual analysis component and how it can be extended to deal with application-specific requirements in Section 5. While Section 6 lists limitations of the current implementation, Section 7 proves our concept by applying it to four realistic codes. Finally, we conclude the paper in Section 8.

This work evolved from a Ph.D. thesis project at Forschungszentrum Jülich. A more detailed and comprehensive description of this article's contents can be found in the thesis document [9].

2. Related work

The multidimensional hierarchical decomposition of the search space for performance problems has a long tradition. Miller et al. [7] developed the W^3 search model as the basis of the online performance-analysis performed by Paradyn. The W^3 model describes performance behavior along the dimensions performance problem, program resources including the call graph [8], and time. Performance problems are expressed in terms of a threshold and one or more metrics such as CPU time, blocking time, message rates, I/O rates, or number of active processors. The different metrics can be specified in a flexible manner using the MDL metric-description language [10]. The main accomplishments of EXPERT in contrast to Paradyn is the description of performance problems in terms of complex event patterns that go beyond counter-based metrics. Also, the uniform mapping of arbitrary performance behavior onto execution time allows the correlation of different behavior in a single view.

Espinosa [11] implemented an automatic trace-analysis tool KAPPA-PI for evaluating the performance behavior of MPI and PVM message-passing programs. Here, behavior classification is carried out in two steps. At first, a list of idle times