

# **APART and OpenMP: A Challenging Step for Automatic Performance Analysis**

Thomas Fahringer<sup>1</sup>, Michael Gerndt<sup>2</sup>, and Wolfgang E. Nagel<sup>3</sup>

<sup>1</sup>Institute for Software Technology and Parallel Systems, University of Vienna,  
A-1090 Vienna, Austria

<sup>2</sup>Central Institute for Applied Mathematics, Research Centre Juelich, D-52425 Juelich, Germany

<sup>3</sup>Center for High Performance Computing, TU Dresden, D-01062 Dresden, Germany

The development of efficient parallel programs solving scientific and industrial challenges is a cyclic process of distinct phases: writing source code, compiling it, executing it on the target platform, evaluating the performance of the program, and improving the program by manual transformation.

Quite elaborate tools are available assisting the application programmer in both steps of the performance analysis phase: program instrumentation and performance data inspection. Current techniques for program instrumentation cover: instrumented libraries, hardware performance monitors, source code instrumentation, object code instrumentation, and on-the-fly dynamic instrumentation. Performance data inspection is supported via interactive tools offering graphical and textual displays: tables of profiling information, time lines of event traces, animated diagrams, and statistical analyses. Although these tools make the analysis process more user-friendly, the learning overhead as well as the overhead in applying these interactive tools is by far too high. In summary, the current situation is dominated by an imbalance between the overhead in applying current performance analysis tools (learning overhead as well as case-based overhead) and the frequently revealed typical performance bottlenecks that seem to be easily detectable. This imbalance is one of the most important reasons why the user community does still not accept current analysis tools and, due to this reluctance, parallel computers do frequently not deliver high performance. This is even true for OpenMP programs.

To improve this situation, the EU working group APART (Automatic Performance Analysis: Resources and Tools) has been formed in January 1999. The goal of this working group is to bring together tool experts, parallel computer vendors, and software companies to discuss all issues in the automation of performance analysis for a broad range of programming environments. This covers design and implementation issues of user guidance support for existing interactive environments as well as of automatic performance analyzers detecting typical performance bottlenecks. The working group includes three European companies, six European universities, one European research center, and three American universities. The partners will try to define a common terminology, list requirements for automatic performance analysis support based on current and future machines, and summarize issues and techniques for implementing an automatic performance analyser interfacing current tools.

Even if the work has just started, there is already a large number of details which have been identified and written down. The discussion is very intensive, and also the OpenMP community is asked to participate in that process (<http://www/fz-juelich.de/apart>). The talk will present first general findings with respect to OpenMP programs and identified bottlenecks, and it will cover special situations where OpenMP programs are running on CC-NUMA machines like SGI Origin2000.