Int J Parallel Prog (2009) 37:247–249
DOI 10.1007/s10766-009-0095-8

Guest Editors' Introduction

Rudolf Eigenmann · Eduard Ayguadé

Published online: 7 April 2009

OpenMP is a portable, scalable programming model that provides a simple and flexible interface for developing shared-memory parallel applications in Fortran, C, and C++. Since its introduction in 1997, as a result of the work by a group of major computer hardware and software vendors, OpenMP has gained support from the majority of high-performance compiler and hardware vendors. Under the direction of the OpenMP Architecture Review Board (ARB), the OpenMP standard has continued to evolve. Version 3.0 was released last year, adding several new features to the OpenMP specification, including: tasking (move beyond loops with generalized tasks and support complex and dynamic control flows), loop collapse (combine nested loops automatically to expose more concurrency), enhanced loop schedules (support aggressive compiler optimizations of loop schedules and AUTO schedule), better definition of and control over nested parallel regions (including new API routines to determine nesting structure).

The community of OpenMP researchers and developers in academia and industry is organized under cOMPunity, a forum for the dissemination and exchange of information about OpenMP. It also serves as a forum for discussing the experiences with this programming API and for debating ideas that might improve it. IWOMP is the annual series of international workshops dedicated to the promotion and advancement of all aspects focusing on parallel programming with OpenMP. The event was a premier opportunity to learn more about parallel programming with OpenMP and other programming models, and to interact with OpenMP users, developers and people working

R. Eigenmann
Purdue University, West Lafayette, IN, USA

E. Ayguadé (✉)
Barcelona Supercomputing Center and Technical University of Catalunya, Catalunya, Spain
e-mail: eduard.ayguade@bsc.es

on the next release of the standard. The workshop also serves as a forum to present the latest research ideas and results related to OpenMP.

The fourth International Workshop on OpenMP (IWOMP-2008) was celebrated at Purdue University (West Lafayette, IN, USA), 12–14 May 2008. The program included 16 papers, 2 keynote/invited presentations and a panel. Over 80 researchers attended the workshop. This special issue of the International Journal of Parallel Programming is devoted to extend the best papers that were presented at IWOMP-2008.

The first three papers present new tools and methodologies to gain insight into OpenMP applications behavior and performance. The first paper, "CLOMP: Accurately characterizing OpenMP Application Overheads" by Greg Bronevetsky, John Gyllenhaal, and Bronis R. de Supinski, introduces a new benchmark to characterize the cost of initiating OpenMP regions. CLOMP complements existing benchmark suites to provide simple, easy to understand measurements of OpenMP overheads in the context of application usage scenarios. CLOMP is able to identify the amount of work required to compensate for the overheads introduced by the implementation of the model.

The second paper, "Capturing and Analyzing the Execution Control Flow of OpenMP Applications" by Karl Fuerlinger and Shirley Moore, describes an approach to capture and visualize the execution control flow of OpenMP applications in a compact way. The approach presented in the paper does not require a full trace of program execution events but is instead based on a straightforward extension to the summary data already collected by a profiling tool. Although the proposed approach is able to represent independent flows of control for each thread in a parallel application, it also offers a simplified representation of the execution control flow for threads with similar behavior, making simpler the understanding of the application.

The third paper, "MPI Correctness Checking for OpenMP/MPI Applications" by Tobias Hilbrich, Matthias S. Muller, and Bettina Krammer, presents a novel approach to correctness checking in hybrid MPI/OpenMP applications. In order to support developers of these hybrid MPI applications, the paper introduces new extensions to an existing tool, Marmot, that make it aware of OpenMP multi-threading, while further ones add new correctness checks. Multiple real and synthetic applications are used in the paper to show the applicability of the approach and the overheads incurred by the data race analysis.

The fourth paper is about an extension to the current specification of OpenMP. "A Proposal to Extend the OpenMP Tasking Model with Dependent Tasks" by Alejandro Duran, Roger Ferrer, Eduard Ayguadé, Rosa M Badia and Jesús Labarta, proposes new clauses that extend the recently introduced tasking model. The proposed extension allows runtime detection of dependencies between generated tasks, enlarging the range of applications that can benefit from tasking or improving the performance when load balancing or locality are critical issues for performance. An experimental evaluation of the proposal is done on a NUMA multiprocessor architecture using a couple of small applications and a prototype runtime system implementation.

The last two papers in this issue present applications that make use of OpenMP. "Parallelism and Scalability in an Image Processing Application" by Morten S. Rasmussen, Matthias B. Stuart and Sven Karlsson, investigates parallelism and scalability for three parallel OpenMP implementations of an embedded image processing application. The paper focuses on the two major challenges faced when parallelizing it. First, the application has limited immediately available parallelism and further extraction of parallelism is limited by small data sets and a relatively high parallelization overhead. And second, load balance is difficult to obtain due to the limited parallelism and made worse by non-uniform memory latency. The evaluation shows speedups larger than 9 on a 16 processor system.

The sixth paper, “A Collaborative Approach For Multi-Threaded SAT Solving“ by Pascal Vander-Swalmen, Gilles Dequen and Michael Krajecki, presents a new parallel scheme to improve the main state-of-the-art enumerative Satisfiability Problem (SAT) solver, a well-known NP-Complete problem that is the core problem in mathematical logic and computing theory. SAT is an original and difficult application for OpenMP for two reasons: (1) is not a computation-intensive application and (2) it is an irregular application in terms of data structures and memory access.

There is still much work to do to evolve OpenMP in the new era of multicore parallelism, in terms of programming model definition, its implementation (compiler and runtime) on novel platforms, tools to make programming more productive, and its use in the emerging applications that will run on these systems. We hope that this collection of papers will prove to be interesting and useful to readers, and that the issues raised will stimulate further research in this area.

Enjoy!