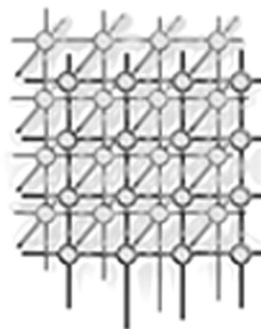

Special Issue: Current Trends in Compilers for Parallel Computers



This special issue of *Concurrency and Computation: Practice and Experience* contains a selection of the papers presented at the 12th International Workshop on Compilers for Parallel Computers (CPC'2006), held in A Coruña, Spain, January 9-11, 2006. The CPC Workshop series is well established as an invitational workshop for leading research groups in the field (mainly from Europe, North America and Asia-Pacific) to provide a forum for exchanging and developing new ideas in compiler design for parallel systems and related topics. The Workshop series began in 1989 in Oxford -UK- and continued every 18 months in an European city: Paris -France- (1990), Vienna -Austria- (1992), Delft -Netherlands- (1993), Málaga -Spain- (1995), Aachen -Germany- (1996), Linköping -Sweden- (1998), Aussois -France- (2000), Edinburgh -UK- (2001), Amsterdam -Netherlands- (2003), Chiemsee -Germany- (2004) and A Coruña -Spain- (2006). The next edition will be held in 2007 in Sintra (Lisbon) -Portugal-.

The 32 papers presented at the 12th Workshop on Compilers for Parallel Computers covered a wide scope of compiler topics with special focus on parallelization and optimization, ranging from instruction-level parallelism to large-scale parallel systems. The variety of the research topics addressed was reflected in the eleven sessions in which the papers were distributed: Embedded Systems, Compiling for VLIW Processors, Exploiting Accelerating Hardware, Memory Hierarchy Optimizations, Analysis Algorithms, Domain-Specific Optimization, Compiling Java, Synchronization and Consistency, Thread-Level Parallelism, Parallel Processing, and Optimizations.

This special issue comprises largely improved and carefully reviewed versions of seven selected papers from CPC'2006. Most of the submitted manuscripts were reviewed by three referees with expertise in the subject area of the manuscript, in a two-round review process of eight months in which fifty-two reviewers participated.

Being the CPC workshop specially focused on parallelizing compilers, it is no surprise that a large fraction of the papers in this issue, three out of seven, fall in this category. Also, it is interesting to notice that they all deal with the shared memory programming paradigm, and that two of them study the parallelization of applications written in an object-oriented language. The first characteristic is a sign of the times, with the appearance first of processors with simultaneous multithreading, and currently with the arrival to the



market of chip multiprocessors with increasing numbers of cores. No doubt these facts are pushing even further the interest of both industry and academia in the improvement of the compilers, runtime systems, and languages available to exploit parallelism in shared memory systems. As for object-oriented programming, it is regarded by many as essential to improve programmer productivity and to support the growing size and complexity of applications, particularly when this complexity is exacerbated in the case of parallel applications. A good proof of the growing interest nowadays in the productivity of programmers and computing systems is for example the DARPA *High Productivity Computing Systems* initiative (HPCS, <http://www.highproductivity.org/>).

The first paper in this special issue, authored by Liao *et al.*, presents OpenUH, a portable OpenMP compiler built on top of the Open64 open source infrastructure that is paired with the portable Tsinghua OpenMP runtime library. The compiler not only supports the complete OpenMP 2.5 specification and is available for the community to be evaluated and expanded, but it also includes many analysis and optimization passes. This turns it into a valuable practical compiler and research tool.

Klemm *et al.* introduce JaMP, an adaption of OpenMP to Java, which they implement in the Jackal research compiler, a tool that provides a software distributed shared memory view for Java. Besides paying attention to the standard semantics of the OpenMP directives and the required syntactical changes needed to implement them in Java, this paper proposes new extensions to OpenMP that are better suited to the object-oriented Java programming philosophy.

Parallel programming in Java with a shared memory paradigm is also addressed by Nishiyama and Nakajima. Concretely, they deal with the detection of quasi-immutable objects, which are those ones that are used much more often for read operations than for write operations. The existence of such objects can be exploited in the Java Virtual Machine to speed up considerably synchronization performance. This is achieved by enabling fast simultaneous read accesses from several threads to the objects that have not been modified in a critical section instead of serializing those threads.

Another trend reflected in this issue is the growing relevance and complexity of embedded processors. It is in this field, much more than in the high-performance computing domain, where the need for low power consumption, coupled with the computing requirements, lead to the development of a large variety of processor families. As a result of the relatively short life and high specialization of these families, binary compatibility is not a priority to the point it is in general purpose computing. This fact allows to experiment with novel features that can be very compiler-dependent such as VLIW and complex register file architectures. The difficulty to program these architectures and the strong real time response restrictions that many of these systems face lead many users to program them manually at low level. This way, the development and understanding of powerful analysis and heuristics for compilers oriented to these architectures is a hot research topic. This is testified for example by the existence of initiatives such as the European Network of Excellence on High-Performance Embedded Architecture and Compilation (HiPEAC, <http://www.hipeac.net/>).

The first representative of this research line in this issue is the paper by Kessler *et al.*, who identify and analyze the different classes of schedules for VLIW processors. Their study



establishes the relations between these schedules and their equivalent linearized forms, valid for traditional superscalar processors, and classifies the VLIW schedules in a hierarchy of classes.

Lin *et al.* propose in their paper a novel register allocation scheme for a clustered VLIW DSP in which some register files are banked. While this structure helps reduce power dissipation and die area, it imposes many restrictions to the register access across clusters and affects instruction-level parallelism. The authors validate successfully their approach, called *ping-pong aware local favorable* (PALF) register allocation, using a compiler based on the Open Research Compiler (ORC).

Finally, data locality and memory hierarchy exploitation is the core of a series of optimizations that have become as essential as recurrent, since the memory wall problem plays a key role in the performance of current computers and embedded systems. This is a well-known fact, whose relevance has done nothing but increase during the past years. The already traditional gap between the processor speed and the memory speed is now being joined by the increase in the number of computing cores that demand data from the same memory. As a result, the understanding of the complex interaction between the software and the memory hierarchy hardware, and the implementation of automatic compiler techniques that help attain an optimal exploitation of this hierarchy are, and will continue to be, a focal area of interest in the years to come. Two papers in this special issue tackle this problem from very different perspectives.

Andrade *et al.* develop a general model of the cache behavior as well as the code analysis infrastructure required to predict the number of misses during the execution of codes with irregular access patterns due to the existence of indirections. These codes are of particular interest, since most research has focused on codes with regular access patterns, while codes with irregular access patterns exploit worse current caches due to the lack of locality of their accesses. Their tool, built on top of the XARK research compiler developed by the authors, is the first one that can yield good estimations of the memory hierarchy behavior for this kind of codes in an automated way without resorting to profiling.

As for Gao *et al.*, they exploit in their paper domain-specific features of the tensor contraction expressions to guide memory hierarchy optimizations. Specifically, such knowledge is used to develop an integrated framework that facilitates the exploration of the search space for the fusion and tiling transformations in order to minimize disk accesses when such expressions involve arrays too large to fit in main memory. Namely, a strategy to reduce the number of loop structures to be evaluated for subsequent optimizations is presented.

In all, we believe that the selection of papers covered in this issue provides a representative sample of the current challenges and interests of the compiler and high-performance computing communities. We anticipate that compiler development will continue to be driven by the appearance of new needs and novel hardware architectures and programming languages and paradigms. In particular, in our opinion, the elusive issue of effective and easy to develop parallel programs will be paid special attention in the years to come thanks to the growing availability of parallelism at different levels in computing systems.

We want to acknowledge and thank the following reviewers of the special issue for their insights and valuable comments to the manuscripts: Wolfram Amme (Friedrich-Schiller University of Jena, Germany), Eduard Ayguadé (Universidad Politécnica de Cataluña, Spain), Javier D. Bruguera (University of Santiago de Compostela, Spain), Timothy J. Callahan



(Carnegie Mellon University, USA), Marcelo Cintra (University of Edinburgh, UK), Pedro Diniz (University of Southern California, USA), Rudolf Eigenmann (Purdue University, USA), M. Anton Ertl (Vienna University of Technology, Austria), María J. Garzarán (University of Illinois at Urbana-Champaign, USA), David Gregg (University of Dublin, Ireland), Minyi Guo (University of Aizu, Japan), Marjan Hericko (University of Maribor, Slovenia), Martin Hirzel (IBM T.J. Watson Research Center, USA), Wei-Chung Hsu (University of Minnesota, USA), Yuan-Shin Hwang (National Taiwan Ocean University, Taiwan), Ben Juurlink (Delft University of Technology, Netherlands), Paul H.J. Kelly (Imperial College London, UK), Christoph Kessler (Linköping University, Sweden), Gerwin Klein (University of New South Wales, Australia), Peter Knijnenburg (University of Amsterdam, Netherlands), Andreas Krall (Vienna University of Technology, Austria), Ulrich Kremer (Rutgers University, USA), Dominique Lavenier (IRISA, France), Jenq-Kuen Lee (National Tsing-Hua University, Taiwan), Gary Lindstrom (University of Utah, USA), José M. Llabería (Universidad Politécnica de Cataluña, Spain), María J. Martín (University of A Coruña, Spain), Seung-Jai Min (Purdue University, USA), Matthias S. Müller (Dresden University of Technology, Germany), Yunheung Paek (Seoul National University, Korea), John Potter (University of New South Wales, Australia), José Renau (University of California, Santa Cruz, USA), Sanjiv Shah (Intel Corporation, USA), Henk Sips (Delft University of Technology, Netherlands), Yan Solihin (North Carolina State University, USA), Michael Stal (Siemens Corporate Technology, Germany), Zehra Sura (IBM T.J. Watson Research Center, USA), Guillermo L. Taboada (University of A Coruña, Spain), Francisco Tirado (Universidad Complutense de Madrid, Spain), Jesper Larsson Träff (NEC Europe, Germany), Theo Ungerer (University of Augsburg, Germany), David Walker (Princeton University, USA), Ian Watson (University of Manchester, UK), Reinhard Wilhelm (Saarland University, Germany), Peng Wu (IBM T.J. Watson Research Center, USA), Jingling Xue (University of New South Wales, Australia), and Xin Yuan (Florida State University, USA).

Finally, we also want to thank Geoffrey Fox, the Editor-in-Chief, for his assistance in all stages of the edition process, and to the members of the CPC'2006 Steering Committee that supervised the review process: Alain Darte (Ecole Normale Supérieure de Lyon, France), Michael Gerndt (Technical University of Munich, Germany), Peter Knijnenburg (University of Amsterdam, Netherlands), Tom Lake (InterGlossa Ltd., UK), Michael O'Boyle (University of Edinburgh, UK), David Padua (University of Illinois at Urbana-Champaign, USA), Henk Sips (Delft University of Technology, Netherlands) and Hans Zima (University of Vienna, Austria).

JUAN TOURIÑO, BASILIO B. FRAGUELA, RAMÓN DOALLO, MANUEL ARENAZ
Special Issue Editors

*Computer Architecture Group
Department of Electronics and Systems, University of A Coruña
Campus de Elviña s/n, 15071 A Coruña
Spain*

E-mail: {juan, basilio, doallo, arenaz}@udc.es