Parallel Architecture and Compilation Techniques: Selection of Workshop Papers, Guests' Editors Introduction

S. Bartolini*, R. Giorgi+, J. Protic#, C. A. Prete*, and M. Valero^

* University of Pisa, Pisa, Italy
+ University of Siena, Siena, Italy
# University of Belgrade, Belgrade, Yugoslavia
^ Universitat Politecnica de Catalunya - UPC, Barcelona, Spain

In this issue, we present a selection of papers from several workshops held in September 2001 in Barcelona, Spain. The workshops were hosted within the PACT (Parallel Architecture and Compilation Techniques) Conference [1], [2].

The advances in technology are improving the processing power and the computing speed of systems. As addressed by keynote speakers, the time has never been so propitious to explore the potentials of compilers on the architecture and vice versa, due to the strong demand for advances in the interaction of these two areas. The increasing interest is also shown by the record number of attendees this year. This is also due to the high-quality workshops focused on hot topics in Compiler and Computer Architecture research areas.

This year 2001, five different workshops covered hot research themes: the *Compilers and Operating Systems for Low Power* (COLP) workshop, the *European Workshop on OpenMP* (EWOMP), the *MEmory DEcoupling Architecture* workshop (MEDEA), the *Ubiquitous Computing and Communication* (UCC) workshop, and the *Workshop on Binary Translation* (WBT). For copyright reasons, we cannot include papers from the Workshop on Binary Translation (WBT). Anyway, the organizers - E. R. Altman, and D. R. Kaeli provided a paper that contains a nice summarization of the works they received. For the other workshops, within the limits given us by the Editor, we wanted to provide some of the presented research papers. The reader will find further updates on each scope of the PACT workshops, since some authors provided an improved version of their original paper.

Following, we will briefly introduce the contributions presented in this issue, so that the reader will locate immediately the contributions of major interest, while not loosing an overall vision of the works in this collection.

Workshop on Compilers and Operating Systems for Low Power (COLP)

Power consumption management while delivering acceptable levels of performance is the challenge in wireless and embedded digital signal processing domains. Managing the extreme power density dissipated in the core in high performance general-purpose systems was the other main theme of this workshop. Those problems were highlighted in the paper by F. Vahid, G. Stitt, and R. Patel, *Propagating Constants Past Software to Hardware Peripherals in Fixed-Application Embedded Systems.* They examined the possibility of propagating the initialization constants from the program to the peripherals

during the synthesis of embedded systems chips. In synthesis tools, constant propagation is commonly done at hardware level, but recognizing software constants as really being hardware constants carries many benefits. The results highlight 2-3 times reductions in peripheral size, and 10%-30% savings in power consumption.

A. Acquaviva, L. Benini and B. Ricco' propose a methodology to analyze the energy overhead of a wearable device due to a real-time operating system. In their paper, the authors analyze the key factors that influence the energy overhead of operating systems services and drivers (I/O data burstiness, thread switch frequency, etc.). Moreover, they addressed the relationship between the voltage scaling, frequency setting techniques and the RTOS power requirements. In this way it is possible to suggest a way to develop OS-aware energy optimization policies. The paper shows experimental results for eCos, an open-source embedded OS, running on a wearable device.

In the paper *Improving Energy Saving in Hard Real Time Systems via a Modified Dual Priority Scheduling*, M.A. Moncusi, A. Arenas, J. Labarta present a modification to the Dual priority scheduling algorithm aimed to improve power consumption. The basic idea is to use the priority scheme to lengthen the run-time of the tasks up to the maximum allowed by the real-time constraint, while accordingly reducing processor frequency and voltage.

European Workshop on OpenMP (EWOMP)

OpenMP is a set of flexible and comprehensive compiler directives, library routines and environment variables, portable across the majority of parallel platforms that use a shared memory, multi-thread system model. OpenMP facilitates parallel programming in FORTRAN and C/C++, speeds up initial programming efforts and removes the requirement of developing a new parallel library when an application is moved to a different machine. The topics covered this year were tools, benchmarks, applications and performance issues of OpenMP, as well as its implementation and extensions for cc-NUMA and SMP clusters. In this special issue, we present one paper from each of the major categories.

The need for parallel benchmarks was addressed by V. Aslot and R. Eigenmann, in their paper *Performance Characteristics of the SPEC OMP2001 Benchmarks.* They describe the first attempt at studying the performance characteristics of eleven applications written in FORTRAN and C, representing modern parallel computer applications, from computational chemistry to finite-element crash simulation and shallow water modeling. Their work examines static and runtime characteristics of the benchmarks, using data gathered with high-resolution timers on Solaris 5.8 and the hardware counters available on the UltraSPARC II processors of a quad processor Sun Enterprise 450 SMP system. In a detailed analysis for all eleven programs, authors identified the increase in memory system stalls as the most important reason for the speedup loss

In *A Microbenchmark Suite for OpenMP 2.0,* authors J. M. Bull and D. O'Neill addressed the subject of benchmarking from another point of view, as they proposed a set of extensions to an existing microbenchmark suite for OpenMP 1.0. The main goal was to investigate performance of new directives introduced by OpenMp 2.0, as well as handling

of thread-private data structures. Some extensions were also proposed for OpenMP 1.0 to include clauses with array arguments. Through the comparison of time taken for a sequential execution of a code section to the time taken for parallel execution of the same code enclosed in a given directive, overheads of various directives were determined and analyzed on a Sun HPC 6500 system and SGI Origin 3000.

Although they do not formally propose an OpenMP extension, D. Nikolopoulos, E. Artiaga, E. Ayguade, and J. Labarta in their paper *Exploiting Memory Affinity in OpenMP through Schedule Reuse* suggest the implicit construction of affinity links between threads and data accesses, which can be reused along the execution of the program in both regular and irregular codes. The experiments on a 64-processor SGI Origin2000 included measurements in the three irregular kernels from a complex weather forecasting code and a simple LU decomposition. The results show the possibility of using customizable loop schedules in OpenMP in order to implement arbitrary data distribution based on the first-touch page placement algorithm. The authors concluded that due to the simple methodology for improving memory access locality, based on the loop schedule reuse, the OpenMP without extensions can scale well on tightly-coupled NUMA architectures, while the future research has to be done for the loosely-coupled NUMA architectures.

MEmory DEcoupling Architecture workshop (MEDEA)

This workshop aimed to revive the Memory Decoupling Architecture concepts from the original idea of J. E. Smith [3] in the context of more advanced processor architectures such as Superscalar and VLIW. An architecture for accelerating general purpose workloads was proposed in the work by M. Sung, R. Krashinsky, and K. Asanovic. They recognized that the primary limitation of decoupled architectures derives from "Loss Of Decoupling" (LOD) events and propose architecture solutions that augment decoupling through multithreading. This is found effective when sufficient thread level parallelism is available.

Multithreading and decoupling also find many sources of Thread Level Parallelism (TLP) in the case of multimedia applications. In their contribution, D. Talla and L. K. John introduced the MediaBreeze architecture for accelerating Multimedia applications. Since this kind of applications is very structured and regular, they easily generate the parallelism that enable the architecture to decouple "true" computations from Multimedia computations. Their results show performance improvements reaching 16x over a conventional SIMD enhanced processor. With the combination with slip-based data prefetching this increase reaches 28x.

Ubiquitous Computing and Communication (UCC) Workshop

Interaction between mobile devices - ranging from handheld to wearable computing - through communication networks is one of the most growing areas of research, due to high number of applications.

New paradigms, protocols, and interactions have to be analyzed. For example, D. Touzet, J.-M. Menaud, F. Weis, P. Couderc, M. Banatre present the SIDE-Surfer system which enriches casual meetings, as facilitated by the emergence of proximity and

wireless communications, by 'proximate' and 'spontaneous' web interactions. Thus they design algorithms and protocols to enable this environment. T. Nakajima describes a new user interface system supporting flexible user interaction for networked home appliances. The presented architecture is based on a stateless thin client system to translate input and output events according to user interaction devices such as keyboards and mice.

References

[1] PACT-2001 Conference, http://research.ac.upc.es/pact01/

[2] PACT Conferences, http://www.pactconf.org/

[3] J.E. Smith, "Decoupled Access/Execute Architectures", *Proceedings of 9th International Symposium on Computer Architecture*, pp. 112-119, May 1982.