

Guest Editors' Editorial: Special Issue on the Second International Workshop on Microgrids

Sven-Bodo Scholz · Alex Shafarenko

Published online: 27 October 2009
© Springer Science+Business Media, LLC 2009

Arrays and streams, this is what multi-/manycore computing could soon be all about. Indeed array processing is the basis of much DSP, machine graphics and multimedia, while streams are a universal glue that keeps processing pipelines chugging away at full speed. Both possess regularity and both are amenable to heterogeneity. The regularity of arrays is spatial, and of streams temporal. Heterogeneity is a requirement that stems from specialisation of computing and communication resources according to stable patterns of usage.

The regularity makes it possible to develop compiler optimisations and hierarchical abstractions to achieve both expressiveness and performance. But it is heterogeneity that gives computing platforms their adaptivity and improves the ability to specialize at run time, when the power of static analyses and optimisations has been exhausted.

The second Microgrids Workshop that took place in Hertfordshire in December 2007 explored the issues of regularity and heterogeneity at great length. Out of 2 days of talks, four presentations have been selected for publication as journal papers, for which extended versions have been solicited from the authors. After substantial revisions, we present them here in a special issue of IJPP.

The motifs of array computing and stream processing permeate all four publications. The first, “Compilation techniques for high level parallel code” contributed by authors from the AMD Corp., ClearSpeed and XMOS, the latter two being parallel computing specialist ventures, focuses on retargeting existing compilers towards data-parallel computing platforms. This is done by extending the datatypes of conventional C by attributes that indicate whether a particular C statement is to be executed on each processor element or only once by the master thread. For a distributed control statement this introduces heterogeneity, which is the focus of this work. Heterogeneity

S.-B. Scholz · A. Shafarenko (✉)
Department of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK
e-mail: A.Shafarenko@herts.ac.uk

of control can be achieved using a standard compiler producing code for each processor element separately, but since the compiler is unaware of special properties of distributed control functions, optimisations are hampered—notably, repeated conditional executions, where the condition can be statically proven not to have changed. The paper proposes optimisation techniques that make heterogeneity of control amenable to optimisation (i.e. minimisation) in an otherwise standard sequential-code compiler.

The presentation of the special issue continues with an article “A Self Distributing Virtual Machine for Adaptive Multicore Environments” by Jan Hasse of the TU Vienna and his colleagues from JW Goethe University in Frankfurt. Here, the intention of the authors is in a way similar to that of the first paper of the volume. They are proposing a virtual machine that hides the heterogeneity of the computing platform underneath it, and presents a clear homogeneous view to a compiler backend. Only this time the focus is on the virtualization level above reconfigurable hardware, rather than a fixed multicore structure. The challenge in this is to be able to capture the adaptivity that stems from the reconfigurability in a transparent way. The middleware developed by the authors allows dynamic variations of the number of processors in a virtual processor array, which is illustrated by performance measurements.

The motif of stream processing is introduced by the next paper, “Asynchronous stream processing with S-Net” by C. Grelck et al. coming from the University of Hertfordshire, which organised the Workshop. The paper introduces a language S-Net, intended for coordinating data-parallel activities in multicore and many-core environments using streams. The language has sufficient expressive power for defining heterogeneous networks containing dynamically unfolding regular subnetworks, and it supports deterministic computing as well as utilising safe forms of nondeterminism. The concepts of S-Net are rather general, and the presentation introduces some abstract design principles that explain the trade-offs involved in coordination using streams. An important lesson of S-Net is that simple uniformly connected components: SISO (Single Input Single Output) entities can be linked into complex heterogeneous networks by harnessing the power of nondeterminism. Although the paper only briefly touches upon the near-equivalence of SISO entities with arbitrary-topology nets, the reader will see clearly that this form of network composition gives rise to many interesting optimisation problems.

A specific (and arguably one of the most important for stream processing) optimisation technique is presented in the concluding article “Run-time spatial mapping of streaming applications to heterogeneous multi-processor systems” from the University of Twente, authored by P. Hölzenspies et al. Just like the paper from Vienna/Frankfurt, this study focusses on run-time adaptation of an application running on a heterogeneous platform, only this time the platform itself is not adaptive: it is the *placement* of activities with the processing elements that adapts at run time. Similarly to the previous article from Hertfordshire, this paper attempts to build a model of distributed computing, or rather what it terms “spatial mapping” in it, and use that model to define a new algorithm for resource utilisation. The algorithm is heuristic (since the problem in its formal representation has been shown to be NP complete), and the quality of the results is demonstrated by performance data from experimental applications.

Although brief, the exposé of new results in array and stream processing that follows below will hopefully provide some markers that delineate the area of debate in the research domain and that indicate the directions of further work. With this in mind we commend the present special issue to the reader.