Building a Concurrency and Resource Allocation Model into a Processor's ISA

Chris Jesshope

Computer Systems Architecture Group, Informatics Institute
University of Amsterdam
The Netherlands
jesshope@science.uva.nl

Abstract. We are now facing the prospect of no increases in computer system's performance unless we harness and efficiently exploit the concurrency that comes from multiple cores on a chip. It should be emphasised that the issues in exploiting concurrency are scale invariant and relate to a few simple parameters and issues. These are: the ratio of the throughput of computation and communication (both local and global), which determines how computation can be distributed and the cost of concurrency creation compared to computation. The latter determines the grain size of the computation. Finally we need virtual concurrency or parallel slack and an efficient data-driven scheduling mechanism, in order to tolerate the latency in any asynchronous activity in the computation, such as access to remote data and resource sharing. The concurrency model used must also be well behaved, i.e. provide determinism of the values computed (although not necessarily of the time required to compute them) and have safe composition. Although these concurrency issues are scale invariant it makes sense to implement them at the lowest scale possible, i.e. at the level of machine instructions, which have overheads measured in single cycles. In this way, all levels of concurrency may be exploited, which is important when dealing with legacy or constrained code. This presentation will explore work undertaken at the University of Amsterdam in designing and evaluating micro-grids of micro-threaded processors that meet these requirements. Moreover the concurrency model developed in this work, SVP, is free of deadlock under composition and has built into its implementations issues which are considered to be operating system ones. Namely it builds in the abstraction of a place, which capture resources and security both in using places and in controlling the execution of concurrency at a place. As the implementation of the concurrency model also manages mapping and scheduling of concurrency, it can truly be said that SVP is an operating system kernel built into the ISA of the processor.

Short Biography

Chris Jesshope is Professor of Computer Systems Engineering at the University of Amsterdam and has held this post since 2004. Prior to this, he has held

E. César et al. (Eds.): Euro-Par 2008 Workshops, LNCS 5415, pp. 129–130, 2009.

posts in a number of universities including a Readership at Southampton University and a Chair at Surrey University, two of the top Electronic Engineering schools in the UK. Professor Jesshope is a Chartered Engineer, a Fellow of the BCS, a Member of the IEEE and a Member of the IEEE Computer Society. His professional activities have included membership on various funding agency committees in both the UK and the Netherlands and the prestigious post of Editor of the IEE Proceedings part E (Computers and Digital techniques) over a 10 year period. He has been involved in numerous program committees, is the Founding Chair of the steering committee of the Microgrid International Workshop on on-chip concurrency and was the founding chair of the steering committee for the Euro-Par International Conference. He has also been general chair for nine international conferences and workshops. Professor Jesshope has given in excess of 50 invited papers or keynote presentations in his career, has written or edited 17 major works, including the very successful book Parallel Computers and published in excess of 160 refereed papers. Most of this work has been in the field of parallel computer architectures and concurrent programming.