Special Guest Lectures

How to Have Your Cake and Eat It Too: High-Productivity and High-Performance with Multicores

Sven-Bodo Scholz, University of Hertfordshire, UK

Johnston Hall 338
January 26, 2011 - 01:00 pm

Abstract:
With the increasing diversity of multicore architectures available it is no longer economically viable to tailor-make programs to the executing hardware. Instead, we have to find ways to automatically create such hardware-specific codes from generic program specifications. We have been working on these issues for the last 15 years. One result of these efforts is SaC, a functional array language (www.sac-home.org), and its attendant compiler technology which combines high programmer productivity with highly efficient executions on various multicore architectures. In this talk, I will highlight the key issues we have encountered on our journey towards compiling high-level, Matlab-like programs into high-performance code. In particular, I will focus on the different requirements and challenges found alongside the road when targeting diverse platforms: My journey will go from general purpose hardware, via restrained settings like GPGPUs towards cutting edge, massively parallel systems where concurrency lies at the heart of performance. This lecture will be via Access Grid. Live presentation will be taking place at 234 Nethken Hall, Louisiana Tech University, Ruston, LA.

Speaker's Bio:
Sven-Bodo Scholz received his PhD and a German Habilitation from the University of Kiel in 1996 and in 2004, respectively. Currently, he is Reader at the University of Hertfordshire, UK. Sven-Bodo Scholz's research is driven by the desire to combine high-level program specifications with highly efficient executions on multiprocessor systems. Theoretical advances and new technologies that affect the interplay of these goals are in the focus of his research. This includes novel type systems, partial evaluation techniques, memory management techniques as well as approaches to generic programming. He has led several research projects that focused around the design and implementation of functional programming systems all of which contributed to several ready-to-use systems in the public domain. He is a key contributor to the design and implementation of the functional array language SaC whose runtime performance competes with that of industrial Fortran compilers. The design of SaC targets various cycle-intensive computations from diverse application areas such as scientific computing, financial modeling or medical image processing. Several SaC-related projects over the last 10 years have led to a substantial compiler framework for SaC. Its outstanding features are the advanced optimisation techniques that are being applied and the auto-parallelizing capabilities for shared memory machines, GPUs, and other novel multicore architectures.

Center for Computation & Technology
2003 Digital Media Center • Telephone: +1 225/578-5890 • Fax: +1 225/578-8957