Topic 4
High-Performance Architectures and Compilers

Michael O'Boyle, François Bodin, Jose Gonzalez, and Lucian Vintan

Topic Chairs

Parallelism is now a central concern for architecture designers and compiler writers. Instruction-level parallelism and increasingly multi-cores are present in all contemporary processors. Furthermore, we are witnessing a convergence of interests with architects and compiler writers addressing large scale parallel machines, general-purpose platforms and specialised hardware designs such as graphic co-processors or low-power embedded systems. Modern systems require system software and hardware to be designed in tandem, hence this topic is concerned with architecture design and compilation. Twenty-four papers were submitted to the track of which five were accepted split over two sessions.

In "Starvation-Free Transactional Memory System", the authors focus on starvation effects that show up in transactional memory. Simple protocols are prone to failure if a first-come first-served policy is used. Low complexity hardware solutions are proposed that do not affect overall performance. In "Towards Real-Time Compression of Hyperspectral Images Using Virtex-II FPGAs" the authors develop an FPGA-based data compression technique. This approach depends on the concept of spectral un-mixing of pixels and sub-pixel targets in hyperspectral analysis. The authors use a two-stage approach which is implemented on an existing FPGA allowing on-board near real-time data compression.

The paper "Optimizing Chip Multiprocessor Work Distribution using Dynamic Compilation" examines how sequential applications benefit from chip multiprocessors. It develops an automatically parallelizing approach based on dynamic compilation which adaptively tunes work distribution. This is experimentally evaluated on the Jamaica chip multi-processor. In " Program Behavior Characterization through Advanced Kernel Recognition" the authors examine how to develop automatic techniques that summarize the behavior of real applications and hide implementation details. This is achieved by describing applications in terms of computational kernels using the XARK compiler. This is thoroughly evaluated on a large set of benchmarks.

In "Compositional Approach applied to Loop Specialization" the authors examine specialised code generation techniques that consider the impact of data sizes on code quality This can lead to code expansion and decision tree overhead. The authors develop a new folding method to specialize code at the assembly level while reducing overhead. They demonstrate the need for specialization on small loops and provide a detailed experimental evaluation.