Compiler Analysis and Optimization of Habanero-Java Programs

Rajkishore Barik
Rice University
rb5@rice.edu

Power, frequency, and memory wall challenges have led to a major shift in mainstream computing by introducing processors that contain multiple low power cores. As multi-core processors become ubiquitous, several efforts for software trends in parallel programming languages have emerged in recent years. Most notably, OpenMP, Cilk ++, X10, TBB, Java Concurrency, TPL, Chapel, and Fortress have been developed to expose the available parallelism on a multi-core processor to the application programmer. Along with new programming languages for parallelism, there is a need for a new paradigm shift in compilers to address the challenge of code optimization of parallel programs, regardless of whether the parallelism is implicit or explicit in the programming model. In this talk, we discuss the need for new compiler analysis and optimization for the Habanero − Java programming language to address the memory wall challenge. The Habanero − Java programming language that is being developed in the Habanero Multicore Software Research project at Rice University focuses on addressing both compiler optimization and runtime implementation challenges for the core constructs of X10 v1.5 language on multi-core processors, with programming model extensions as needed (such as phasers and isolated blocks). The high-level compiler optimizations we will describe in our talk include new approaches to May-Happen-in-Parallel analysis and Side-Effect analysis for parallel programs, and a novel parallelism-aware Scalar Replacement for Load Elimination transformation.