StreamIt – A Programming Language for the Era of Multicores

For the last four decades, a byproduct of Moore's Law has been the continuous and dramatic increase in the performance of sequential applications. Unfortunately, in the current and future generations of processors, doubling the number of transistors is not leading to any increase in sequential performance due to power and complexity issues. Thus, all major processor vendors are moving towards multicore processors. While architects have known how to build parallel processors for over a half a century, the main stumbling block for their wider acceptance has been the difficulty in programming them. In the first part of the talk I will discuss the path to multicores, address why parallel programming has been such a difficult problem to solve and speculate on our ability to crack it this time around.

One promising approach to parallel programming is the use of novel programming language techniques – ones that reduce the burden on the programmers, while simultaneously increasing the compiler’s ability to get good parallel performance. In the second part of the talk, I will introduce StreamIt: a language and compiler specifically designed to expose and exploit inherent parallelism in "streaming applications" such as audio, video, and network processing. StreamIt provides novel high-level representations to improve programmer productivity within the streaming domain. By exposing the communication patterns of the program, StreamIt allows the compiler to perform aggressive transformations and effectively utilize parallel resources. StreamIt is ideally suited for multicore architectures; recent experiments on a 16-core machine demonstrate an 11x speedup over a single core.