CS 612: Software Design for High-Performance Architectures

Instructor: Professor Keshav Pingali (pingali@cs.cornell.edu) 457 Rhodes Hall

Time: Tuesdays & Thursdays, 10:10 - 11:25 AM, Upson 109

Prerequisites: An undergraduate compiler course (like CS412) is helpful but not required. Programming experience is essential.

Course content: This semester, we will focus on the problem of designing intelligent software systems. Intelligence may be defined as the ability of an entity to adapt to its environment to increase its efficiency or to increase its chances for survival. By this definition, existing software systems are not very intelligent; yet, emerging hardware platforms and programming paradigms demand more intelligence from software systems.

The first part of the course will focus on software adaptation for efficiency. We will study sophisticated program analysis and transformation tools that are used to make programs run efficiently on modern computing platforms that have multiple processors, deep memory hierarchies, multithreading etc. Some of this technology was developed at Cornell, and is in use in production compilers at SGI, IBM, Intel and other companies. We will then study the role of AI techniques such as intelligent search for adaptively choosing optimal values for transformation parameters.

The second part of the course will focus on software adaptation for survival. We will survey system-level protocols developed in the distributed systems community for making fault-tolerant systems. We will argue that application-level protocols are more appropriate for making programs fault-tolerant, and then discuss such protocols and the program analysis problems that must be solved to implement them efficiently.

Course-work includes three or four programming assignments, and a final project.

Topics covered in the course include the following.

1. High-performance architectures
   - Shared/distributed memory machines
   - Memory hierarchies
   - Multithreading
2. Adaptation for efficiency
   - Scalar analysis: lattice algebra, control dependence, classical and sparse data flow analysis, inter-procedural dataflow analysis
   - Array analysis: polyhedral algebra, Diophantine equations, array dependence analysis
   - Pointer analysis: alias relations and their representations, computing interprocedural alias relations
   - Fractal symbolic analysis
   - Transformation theories for perfectly-nested and imperfectly-nested loops
   - Empirical optimization
3. Adaptation for survival
   - System-level protocols for checkpointing and message-logging
   - Application-level protocols
   - Program analysis problems for optimizing application-level protocols