CS 595 – Advanced Compiler Design: Compiler Analysis and Optimization for High-Performance Architectures

Important facts:
Meeting Time: Thursdays, 2–4:50
Instructors: Mary W. Hall, Jacqueline Chame

Course Description:
This course is an in-depth study of the analyses and optimizations involved in compiling for high-performance parallel architectures. This area has traditionally been thought of as targeting big-iron supercomputers, but it has become increasingly important as almost every architecture today is a parallel architecture, including VLIW microprocessors, systems-on-a-chip, novel memory systems, FPGA-based systems, multimedia extension architectures and chip multiprocessors. The course will address issues such as managing complex memory hierarchies, exploiting parallelism both at the fine grain (such as for VLIW architectures and superword parallelism on multimedia extensions) and coarse grain (such as large scale parallel architectures). The prerequisite for the course is CS565 (Compiler Design), or approval by the instructors. Further, a background in computer architecture and parallel computing will significantly contribute to understanding the subject matter.

The course will be divided into two parts. The first part will focus on building core knowledge used in compilers for high-performance architectures, and the emphasis will be on lectures and homeworks, based on the textbook, High Performance Compilers for Parallel Computing by Michael Wolfe. In the second part of the course, we will explore research topics through lectures, paper presentations, and projects. Each student will present one paper, and develop a project. The primary purpose of the project is to gain hands-on experience in the issues facing compiler writers and explore research-related questions on some of the topics studied during the semester.

Suggested projects:
1. Coherence and communication libraries in Universal Parallel C.
2. Optimizations that adapt dynamically to performance monitoring in the SGI Origin.
3. Analysis/implementation of reduction operations for multimedia extension architectures such as MMX.
4. Compiler transformations for power reduction.
5. Pointer alias analysis, trading off performance, implementation complexity and precision.

Syllabus:
Week 1 Overview and Motivation
Week 2 Data Dependence Analysis
Week 3 Array Region Analysis
Week 4 Parallelization and Vectorization
Week 5 Loop Transformations
Week 6 Loop Transformations for Locality
Week 7,8 Analyzing Pointer Accesses
Week 9 Compiler-directed Software Prefetching
Week 10 Compiling for ILP
Week 12 Coherence
Week 13 Message Passing
Week 14 Architecture Case Study: DIVA
Week 15 Project Presentations