Spring 2013

CS 7900-01: Optimizing Compilers for Modern Architectures

Meilin Liu
Wright State University - Main Campus, meilin.liu@wright.edu

Computer Science (CS) 7900 01
Optimizing Compilers for Modern Architectures

Spring Quarter 2013
Wright State University

Course Description

This course studies compiler optimization for modern architectures, program performance optimization. Between parsing the input program and generating the target machine code, optimizing compilers perform a wide range of program transformations on a program to improve its performance. In this course we focus on data dependence analysis, program transformations and loop transformations, loop scheduling, and a combination of these optimizing techniques.

Lecturer
Meilin Liu
Office: 353 Russ Engineering Center
Phone: 937-775-5061
Office Hours: 2:00 – 4:00 pm Tuesday/Thursday
Email: meilin.liu@wright.edu
Web: www.wright.edu/~meilin.liu

Class

- Tuesday/Thursday 6:30– 7:50 pm MC BS 105

Text

Optimizing Compilers for Modern Architectures, Rand Allen and Ken Kennedy, Second Printing, Morgan-Kauffman.

Reference

High Performance Compilers for Parallel Computing, Michael Wolfe, Addison-Wesley, 1996
Engineering a Compiler, Keith Cooper, Linda Torczon, Publisher: Morgan Kaufmann; first edition, 2003.

Prerequisite: CEG 320 or CEG 420 Computer Architecture or with the permission of instructor

Paper Reading, Review, and Presentations

Each student is required to present one paper during the last two weeks of the class. The presentation should be 15 to 20 minutes, followed by an 5 - 10 minute discussion. Students are required to read the paper carefully before presenting the paper, and write a mini review for the paper. Each review should include 1) the strength and the weakness of the paper 2) the background and motivation, 3) related work, 3) the assumptions, designs, and experimental results and analysis, 4) conclusion, and 5) future work. Paper reviews need to be submitted to the instructor one day before the presentation.

Required Work

Homework	15%
Quizzes	10%
Paper Review and Presentation	5%
Project	25%
Midterm Exam	20%
Final Exam	25%

Grading

The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary.

Tentative Schedule

(The schedule may subject to change.)

Week	Contents	Read, Project
1	Architecture Overview, Memory Hierarchy, Cache Organization	Chap. 1
2	Control Dependence, Control Flow Graph, Basic Block	Chap. 7
3	Data Dependence Theory Analysis	Chap. 2
4	Data Dependence Analysis	Chap 3
5	Instruction Scheduling, List Scheduling	Chap 10
6	Dominator Frontier, Static Single Assignment	From Paper. (Project 1 Due)
7	Program Transformations, Loop Transformations	Chap 4
8	Spring Break	
9	Program Transformations, Loop Transformations	Chap 4
10	Liveness Analysis, Register Allocation	From Paper.
11	Local Register Allocation	From Paper.
12	Software Pipelining	From Paper.
13	Software Prefetching, Review	Chap. 8
14	Paper Presentation by students	Selected papers
15	Paper Presentation by students	Selected papers (Project 2 Due)
	Final Exam: Thursday, April 25th, 2013, 5:45-7:45 pm.	

Policies and Notes

- Attendance: Attendance is not required, but recommended. For your own sake, you should not miss any of the classes. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!

- I will utilize Pilot (pilot.wright.edu) to post updates to the course, solutions, assignments, announcements, schedule, etc. Get in the habit of checking it regularly.

- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible.

- A penalty of 10% deduction each day for late submission of homework will be given and after one week, 0 point will be given.

Academic Misconduct

In this class, the only way to truly learn the concepts to is do the work yourself. I encourage working with other people on the course concepts. When you begin to do the homework and the projects, do it on your own.