Spring 2008

CS 790-02: Optimizing Compilers for Modern Architectures

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

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
https://corescholar.libraries.wright.edu/cecs_syllabi/559

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
Computer Science (CS) 790
Optimizing Compilers for Modern Architectures

Spring Quarter 2008
Wright State University

Course Description

This course studies compiler optimization for modern architectures. 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: 1:30 – 3:00 pm Tuesday/Thursday
Email: meilin.liu@wright.edu
Web: www.wright.edu/~meilin.liu

Class
• Monday/Wednesday 8:00– 9:15 pm Joshi Research Ctr 193

Text

Reference
High Performance Compilers for Parallel Computing, Michael Wolfe, Addison-Wesley, 1996
Parallel Computer Architecture, David Culler and Jaswinder Pal Singh, Morgan Kaufmann, 1999

Required Work
Homework 20%
Quizzes 10%
Project 40%
Final Exam 30%

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.

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 webCT (wisdom.wright.edu) to post updates to the course, solutions, assignments, announcements, schedule, etc. Get in the habit of checking it regularly.

- Always make back ups of all of you work. Never have just one copy of anything!

- 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.

- If you need to meet me other than my office hours, better make an appointment by email or by phone beforehand.

**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.

**Schedule**

(The schedule may subject to change.)

<table>
<thead>
<tr>
<th>Week</th>
<th>Contents</th>
<th>Read, Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Architecture Overview</td>
<td>Chap. 1</td>
</tr>
<tr>
<td>2</td>
<td>Data Dependence Theory</td>
<td>Chap. 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Project1)</td>
</tr>
<tr>
<td>3</td>
<td>Data Dependence Analysis</td>
<td>Chap. 3</td>
</tr>
<tr>
<td>4</td>
<td>Data Flow Analysis</td>
<td>Chap 4</td>
</tr>
<tr>
<td>5</td>
<td>Control Dependence, Control Flow Graph, Basic Block</td>
<td>Chap 7</td>
</tr>
<tr>
<td>6</td>
<td>Dominator Frontier, Static Single Assignment</td>
<td>From Paper.</td>
</tr>
<tr>
<td>7</td>
<td>Program Transformations, Loop Transformations</td>
<td>Chap 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Project 1 Due)</td>
</tr>
<tr>
<td>8</td>
<td>Scheduling, Trace Scheduling</td>
<td>Chap.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Project 2)</td>
</tr>
<tr>
<td>9</td>
<td>Fine grained parallelization, Course grained parallelization</td>
<td>Chap. 5, 6</td>
</tr>
<tr>
<td>10</td>
<td>Register Management, Review</td>
<td>Chap. 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Project 2 Due)</td>
</tr>
</tbody>
</table>