

COMP515-Fall-2011

COMP 515: Advanced Compilation for Vector and Parallel Processors (Fall 2011)

Instructor:	Prof. Vivek Sarkar , DH 3131	Graduate TA:	Sanjay Chatterjee
Assistant:	Amanda Nokleby, akn3@rice.edu , DH 3137, 713-348-2031	Office hours:	By appointment
Lectures:	Keck Hall (KH) 107	Time:	TTh 10:50am-12:05pm

Introduction

Are you curious about what makes a parallelizing compiler tick, and what program properties enable or prevent software from executing efficiently on vector and parallel computers?

This question has taken on new importance with the ubiquity of multicore processors, and the increased interest in heterogeneous accelerators such as GPGPU's and FPGA's. To answer this question, this course will cover the foundations of advanced compilation techniques for vector and parallel computer systems, including the analysis of program dependence, program transformations to enhance parallelism, compiler management of the memory hierarchy, and inter-procedural data flow analysis. At the conclusion of the course, the students should be knowledgeable about the strengths and limitations of state-of-the-art compilers, both from the viewpoint of the compiler user as well as of the compiler designer. The techniques taught in the course should be relevant to anyone interested in enabling software to execute efficiently on vector and parallel computers, whether by hand or through the use of compilers.

Recommended Prerequisite: [COMP 412](#) or equivalent

Lectures

1. [Lecture 1 \(08/22/2011\): Compiler Challenges for High Performance Architectures \(Chapter 1\)](#)
2. [Lecture 2 \(08/22/2011\): Dependence: Theory and Practice \(Chapter 2\)](#)
3. [Lecture 3 \(08/30/2011\): Dependence: Theory and Practice \(Chapter 2, contd\)](#)
4. [Lecture 4 \(09/01/2011\): Dependence Testing \(Chapter 3\)](#)
5. [Lecture 5 \(09/06/2011\): Dependence Testing \(Chapter 3, contd\)](#)
6. [Lecture 6 \(09/08/2011\): Dependence Testing & Preliminary Transformations \(Chapter 3, contd, Chapter 4\)](#)
7. [Lecture 7 \(09/13/2011\): Preliminary Transformations & Enhancing Fine-Grained Parallelism \(Chapter 4, contd, Chapter 5\)](#)
8. [Lecture 8 \(09/15/2011\): Vectorization Algorithm & Enhancing Fine-Grained Parallelism \(Chapter 2, contd, Chapter 5, contd\)](#)
9. [Lecture 9 \(09/21/2011\): Enhancing Fine-Grained Parallelism \(Chapter 5, contd\)](#)
10. [Lecture 10 \(09/22/2011\): Enhancing Fine-Grained Parallelism, Coarse-Grained Parallelism \(Chapter 5, contd, Chapter 6\)](#)
11. [Lecture 11 \(09/29/2011\): Coarse-Grained Parallelism \(Chapter 6, contd\)](#)
12. [Lecture 12 \(10/05/2011\): Coarse-Grained Parallelism \(Chapter 6, contd\), Control Dependence \(Chapter 7\)](#)
13. [Lecture 13 \(10/06/2011\): Midterm Review](#)
14. [Lecture 14 \(10/18/2011\): Control Dependence \(Chapter 7, contd\)](#)
15. [Lecture 15 \(10/20/2011\): Control Dependence \(Chapter 7, contd\), Compiler Improvement of Register Usage \(Chapter 8\)](#)
16. [Lecture 16 \(10/24/2011\): Compiler Improvement of Register Usage \(Chapter 8, contd\)](#)
17. [Lecture 17 \(11/01/2011\): Compiler Improvement of Register Usage \(Chapter 8, contd\), Managing Cache \(Chapter 9\)](#)
18. [Lecture 18 \(11/03/2011\): Managing Cache \(Chapter 9\)](#)
19. [Lecture 19 \(11/08/2011\): Automatic Selection of High-Order Transformations in the IBM XL Fortran Compiler](#)
20. [Lecture 20 \(11/10/2011\): Automatic Selection of High-Order Transformations in the IBM XL Fortran Compiler \(contd.\)](#)
21. [Lecture 21 \(11/15/2011\): Unimodular Transformations](#)
22. [Lecture 22 \(11/18/2011\): Polyhedral Compilation Framework \(Guest lecture by Dr. Louis-Noel Pouchet\)](#)
23. [Lecture 23 \(11/22/2011\): Interprocedural Analysis and Optimization \(Chapter 11\)](#)
24. [Lecture 24 \(11/29/2011\): Compiling Array Assignments \(Chapter 13\)](#)
25. [Lecture 25 \(12/01/2011\): End-Semester Summary \(Chapters 7, 8, 9, 11, 13\)](#)

References

1. [Course text: Optimizing Compilers for Modern Architectures](#), Randy Allen & Ken Kennedy
2. [A General Framework for Iteration-Reordering Loop Transformations](#). Vivek Sarkar and Radhika Thekkath. Proceedings of the ACM SIGPLAN '92 Conference on Programming Language Design and Implementation (PLDI), San Francisco, California, pages 175-187, June 1992.
3. [Optimal Weighted Loop Fusion for Parallel Programs](#). Nimrod Megiddo and Vivek Sarkar. Proceedings of the Ninth Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA), Newport, Rhode Island, June, 1997, pages 282-291.
4. [Array SSA form and its use in Parallelization](#). Kathleen Knobe and Vivek Sarkar. Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Diego, California, January 1998.
5. [Automatic Selection of High Order Transformations in the IBM XL Fortran Compilers](#). Vivek Sarkar. IBM Journal of Research and Development, 41(3), May 1997.

Past offerings of COMP 515

- [Spring 2009](#)
- [Spring 2006](#)

