CS 498: Program Optimization

Fall 2006

University of Illinois at Urbana-Champaign

Course Organization

- Instructor:
  - Maria J. Garzaran
- Guest Lecturers
  - Paul Petersen, INTEL
  - Arch Robinson, INTEL
- Office: 4308. Email: garzaran@cs.uiuc.edu
- Office Hours: By appointment (send me an email)
- Course website: http://www.cs.uiuc.edu/class/fa06/cs498dp/

- Location: Room 1131, W-F 11:00 - 12:15
- Credits: 3 undergraduate hours; 3 or 4 graduate hours

Course Subject

- Program optimization
- One of the main topics in computer science although there are few courses focusing on this topic.
- Improve programs (or generate programs with good properties along one or more of the following dimensions)
  - Execution time
  - Power
  - Space
  - Reliability
  - Modularity, Readability
  - Accuracy/completeness

Course Subject

- Focus of this course is the first goal and some discussion of the second.
- Some issues discussed
  - Factors that affect performance
  - Program transformations
  - Tools
  - Automatic program synthesis (library generators)

Course Subject

- Importance of reducing execution time
  - Faster programs are better.
    - An interactive environment exacerbates the thirst for speed.
    - More science/better engineering designs in same amount of time
  - Although machines are getting more powerful, sometimes the only way to take advantage of this power is through program optimization.
    - Itanium instruction level parallelism
    - Multicore machines

Course Subject

- How is a program optimized ?
  - Manually
    - Algorithm choice
    - Library selection
    - Need tools to assess progress and understand what is happening.
    - Tradeoff: development and maintenance time vs. performance.
  - Automatically
    - By a compiler
    - By developing code generators that automatically search for best shape of the program. Here the choices are those of the programmer while in a compiler the choices are those of the compiler writer.

Course Subject

- The main focus of this course is manual optimization, but we need to understand compilers unless we are willing to program in assembly language.
- Understanding the manual approach is important for compiler writers since they should dominate the manual approach before trying to automate it.
- The manual approach is also important for machine designers so they can understand their choices when mapping programs to their designs.

Course Subject

- For performance and power optimization it is necessary to have a good understanding of target machine features.
- Program optimization is difficult for many reasons:
  - Machine features may interact in ways that are difficult to analyze. Explaining/predicting behavior is difficult and has become more difficult with increasing machine complexity.
  - There many ways to solve a problem and in many cases it is not clear which one is better. It depends on the class of machine, and the characteristics of the input data. For example, it is difficult to know when is quicksort better than radix sort.
  - In general, a “proof” of optimality is unrealistic. Usually difficult to know how much more could be done.

Tentative List of Topics

- Compiler Optimizations
- Locality Enhancement
- Parallellism
  - Threading
  - Race conditions (threading tools)
  - Instruction Level Parallelism
  - Vector Devices
- Library Generators (Matrix-matrix multiplication, sorting, spiral, sparsity, datamining)
- Power-Aware Compiler Techniques

Tentative Course Outline

- F 8/25: Introduction. Course Organization. What is is program optimization? Optimization and machine organization.
- W8/30: Optimization Challenges for High Performance Architectures.
- F 9/1: Compiler Optimizations
- W 9/6: Compiler Optimizations
- F 9/8: Vtune (Paul Petersen)
- W 9/13: Locality/Tiling
- F 9/15: Locality/Tiling
- W 9/20: Locality/Tiling

Tentative Course Outline

- F 9/22: Parallel Programming- Episodes I & II (Arch Robison)
- W 9/27: Parallel Programming- Episodes III & IV (Arch Robison)
- F 9/29: Parallel Programming- Episodes V & VI (Arch Robison)
- W 10/4: Threading Tools (Paul Petersen)
- F 10/6: Introduction to OpenMP
- W 10/10: Intruction-Level Parallelism
- F 10/13: Introduction to Library Generators. Empirical Search

Tentative Course Outline

- W 10/18: Midterm
- F 10/20: The ATLAS system
- W 10/25: The ATLAS system
- F 10/27: Matrix-multiplication for Graphics Processor unit
- W: 11/1: Sorting
- F: 11/3: Sorting
- W: 11/8: The SPIRAL system
- F: 11/10: The SPARSITY system
- W: 11/15: Datamining optimizations
- F: 11/17: Datamining optimizations

Tentative Course Outline

- W: 11/29: Programming SSE devices
- F: 12/1: Power-Aware Compiler Techniques
- W: 12/6: Project presentation
- F: 12/8: Project presentation

Format of the clas

- I will use slides (will try to put the slides in advance)
- There will be reading assignments for some of the topics covered

Project

- Teams of three students
- Each team chooses a program to optimize
- Milestones during the semester
- Final report is the only one that will be graded
- Contributes a 50% to the final grade

Exams

- Exams will be closed books and closed notes. Final will only cover the material not covered in the midterm

- Midterm: Wednesday 10/18
- Final: Date to be announced

Grading

- Final grades will be computed based on:
  50% project + 25% midterm + 25% final

Questions?