EECS 583: Advanced Compilers

Fall 2019 Syllabus

Class and Instructor

Class: Mon/Wed, 10:30am-12:30pm, 2246 SRB
Prof. Scott Mahlke
Email: mahlke@umich.edu
Office: 4633 CSE Bldg.
Office hrs: MW 12-12:30 (right after class) in 2246 SRB or by appointment

Course Description

An in-depth study of compiler backend design for high-performance architectures. Basic topics include control-flow and dataflow analysis, optimization, instruction scheduling, modulo scheduling, and register allocation. Advanced topics include memory dependence analysis, automatic vectorization/thread extraction, streaming applications, predicated and speculative execution, dynamic compilation, and security. The focus is backend compilation, thus familiarity with computer architecture and compilers is recommended.

Reference Books

- *Advanced Compiler Design & Implementation*, Muchnick, Morgan Kaufmann, 1997.
- *Compilers: Principles, Techniques, and Tools (2nd edition)*, Aho, Lam, Sethi, Ullman, Pearson Addison-Wesley, 2007.

Prerequisites

Strong C++ programming skills (EECS 281), good background in computer architecture (EECS 370 at minimum), some familiarity with compilers (EECS 483 is desirable but not needed).

Grade

Midterm exam - 25%
Project - 45%
Homeworks – 15%
Paper presentation – 10%
Class participation – 5%

Midterm exam - There will be one in-class (2 hour) exam at about the 2/3 point of the class. The exact date is TBD. The exam will be open book/notes.

Project - The projects will consist of designing and implementing an advanced compiler technique within the LLVM compiler infrastructure (or other compiler system in certain cases). A report describing the project should be submitted along with a brief presentation and/or demonstration of the resulting implementation. Typical projects consist of 2-4 students, 1-person and 5-person projects are allowed under special circumstances. There will be a project proposal and project update for each group scheduled during the semester.

Homeworks – 2 programming assignments will be done in the early portion of the semester. Each homework will consist of implementing something within the LLVM compiler system and showing its operation on several test programs. Each student must do their own work and turn in their own assignment.

Paper presentation – During the research topic portion of the class, each project group will present a research paper to the class related to their project. Each group is responsible for selecting an appropriate conference paper and giving a 20-minute presentation to the class + 5 mins Q&A. Scheduling will be done in class in conjunction with the project proposals.

Class participation - Students are encouraged to take an active role in this class by asking questions or providing comments.

Rough Topic list

- Control flow analysis and optimization
  - Basics: control flow graphs, dominators, loop detection
  - Regions: traces, superblocks
  - Predicated execution: control dependence analysis, hyperblocks
  - Code layout, alignment
- Dataflow analysis and optimization
  - Basics: liveness, reaching defs
  - Static single assignment form
  - Classical and ILP optimization
  - Analysis applications (security, reliability)
- Code generation
  - Basics: dependences, latencies, ASAP/ALAP times
  - Instruction scheduling, superblock scheduling, control speculation
  - Modulo scheduling, rotating registers
  - Register allocation
- Compilation for multicore
  - Parallelization of loops: Vector, DOALL, DOACROSS, DSWP
  - Intro. to dynamic (JIT) compilation
- Research topics