Welcome!

Simone Campanoni
simonec@eecs.northwestern.edu
Prerequisites

Source code (e.g., C++)

Front-end

IR

Middle-end

IR

Back-end

Machine code

**EECS 322: Compiler Construction**

myVarX = 40
myVarY = myVarX + 2

**EECS 323: Code analysis and transformation**

myVarY = 42

**EECS 322: Compiler Construction**

010101110101010101
Outline

• Structure of the course

• This year’s topics

• Software framework
ATC

- You learned the internals of modern production-quality compilers
  - Data-flow analysis, constant propagation, memory alias analysis (CS 323)
  - Instruction selection, register allocation (CS 322)
- Research labs include advance techniques not yet included in production-quality compilers
  - They aren’t as robust as production-quality compilers need to be
- ATC:
  - You will learn some of these advanced techniques
    - They are organized in topics.
    - Each year we look at techniques of two topics
    - Each year ATC is different
ATC assignments

• You will learn some advanced compilation techniques
  • You need to read and learn two research papers per week (you will have these two papers the week before they are due)
  • You will present a few of them and you will need to defend them (as they are your papers)

• You will do a project you choose from a set
  • The set of projects are related to the topics of the current year
  • You can work in a team (maximum 3 people per team)
  • You will develop your project during the quarter
  • Each team will meet me weekly to discuss progress and roadblocks
  • You will present your project to the rest of the class during the last 2 weeks
Before the first lecture of this week, you need to choose the project to work on and your team.

At the end of a week, you’ll have the 2 papers to read (and the presenters) for the next week when we will discuss them.
Before the first lecture of this week, you need to choose the project to work on and your team.

You develop your project.

You present your improvements (description, empirical evaluation, code walk).

The ATC structure:

- Basics for project
- Topic & project

Today

Week

- Tuesday: Project
- Thursday: Project
Materials

• Software:
  • NOELLE: it can be downloaded from [here](#)
  • VIRGIL: it can be downloaded from [here](#)
  • Not ours: from the web

• Documentation:
  • Papers: either from the web (when available) or from Canvas
  • Software docs: from the web (if they exist)
Outline

• Structure of the course

• This year’s topics

• Software framework
Topics

• Every year ATC covers two different topics

• Topics we will cover this year:
  • Memory-centric compilers
  • Compilers for heterogeneous platforms
Papers

• Distributed via Canvas every Friday

• Papers shared before a Friday are due the week after (one per class)

• Each paper will be presented by one of you
  • Everyone needs to read all papers before the class and participate in the discussion

• Roles:
  • Presenter: brief description of the paper focusing on the novelty of the work
  • Everyone altogether: paper discussion
Mindset to use

• In this class: you will read 1 paper per lecture
  • Not 2-4

• Expectation: you will read the single paper very careful
  • You need to understand it in depth
  • You need to be ready to discuss it with the rest of the class
    (even if you are not the presenter 😊)

• Each paper will be presented by one of you
  • I will select the presenter the day before class
Outline

• Structure of the course

• This year’s topics

• Software framework
Your project

• This class requires paper reading and a project
  • Can be done in a team or not

• You will work at the frontier of compiler research
  • Edges at the frontier are sharp
  • I’ll guide you through your project to avoid getting cut

• You will provide weekly update on your progress
  • Where: Zoom
  • When: please use the doodle link at my [website](#) to find 30 minutes time
Software framework

• You will need to use/extend one of our software framework
  Which one depends on the project you will choose to do

• NOELLE: our set of abstractions/transformations/analyses
  that can be used by an LLVM middle-end pass
  A collection of
  state-of-the-art parallelization techniques,
  program analyses and transformations

• VIRGIL: our simple runtime used by the parallelization techniques
  included in NOELLE
Papers to read
Papers to read for next Thursday

• Paper:
  “NOELLE Offers Empowering LLVM Extensions”
  • Download from here: https://arxiv.org/abs/2102.05081
  • Presenter: Tommy

• Tuesday will be a tutorial about NOELLE