

CS698L: Programming for Performance

Swarnendu Biswas

Semester 2019-2020-I

CSE, IIT Kanpur

Course Details



- CS698L:Programming for Performance
- Semester 2019-2020-I
- Class hours: WF 9:00-10:15 AM KD 102
- Office hours: TuTh 4-5 PM PM KD 302
- Webpage: <https://www.cse.iitk.ac.in/~swarnendu/courses/cs698l/>
- Discussion forum: **REGISTER** for CS698L on Piazza
- Submission portal: We will use Canvas

Instructor Details



- Name: Swarnendu Biswas
- Office: KD 302
- Webpage: <https://www.cse.iitk.ac.in/~swarnendu>
- Email: swarnendu@cse.iitk.ac.in

TA Details



- Name: Amit Kumar
- Email: amitkumr@iitk.ac.in

- Name: Sri Divya Yaddanapudi
- Email: srid@iitk.ac.in

- Name: Mohit Malhotra
- Email: mohitm@iitk.ac.in

Course Outline



Planned

- Single-Processor Performance
 - Cache and data locality analysis
 - Data dependences and fine-grained parallelism
 - Loop and data transformations
 - Vectorization
- Parallel Computing Platforms
- Parallel Programming Models
 - Shared-memory programming (Pthreads, OpenMP)
 - GPGPU architecture and CUDA programming

Tentative

- Architecture Basics
 - pipelining, OOO, superscalar, VLIW
 - Caches and cache coherence
- Task-Based Programming
 - e.g., Intel TBB

Prerequisites

- Background in OS and architecture is expected
 - CS 210, CS 330, CS 422 (or equivalent)
- Good knowledge in C/C++ (and Java) is desirable
 - We will have a few programming assignments
 - Primarily C/C++, OpenMP, CUDA

Course Policies

- Please be **ON TIME** to class!
- Please try to **AVOID** using laptops and/or mobile devices in class!
- Turn **OFF** your mobile phones!
- Email subjects **SHOULD** start with [CS698L]
- Submitting your assignments late will mean losing points automatically. You will lose 10% for each day that you miss, for up to three days.

Evaluation



- Class/piazza participation – 5%
- Assignments – 40%
- Mid semester exam – 20%
- End semester exam – 35%

Academic Integrity

- You **MAY** discuss concepts with classmates
- All assignments **MUST** be your own or your team's work when teamwork is permitted
- You **MAY NOT** search online for existing solutions related to the assignments, even as a reference
- Students caught **CHEATING/PLAGIARIZING** will be punished

Teaching Philosophy

- I am happy to clarify **EVERY REASONABLE** question
- Slides will primarily be pointers to concepts and materials
 - I will post optional reading material
- I encourage you to make use of the office hours

Resources



- Automatic Parallelization: An Overview of Fundamental Compiler Techniques – Samuel P. Midkiff
- Parallel Computer Architecture: A Hardware/Software Approach – D. E. Culler et al.
- An Introduction to Parallel Programming – P. Pacheco
- Computer Systems: A Programmer's Perspective – R. Bryant and D. O'Hallaron
- Compilers: Principles, Techniques and Tools – A. Aho, M. Lam, R. Sethi and J. Ullman
- Optimizing Compilers for Modern Architectures – R. Allen and K. Kennedy

- Other handouts and research papers

Resources



- Automatic Parallelization: An Overview of Fundamental Compiler Techniques – Samuel P. Midkiff
 - Parallel Computer Architecture: A Hardware/Software Approach – D. E. Culler et al.
 - An Introduction to the Theory of Computation
 - Computer Architecture: A Quantitative Approach
 - Compilers: Principles, Techniques, and Tools
 - Optimization: Principles and Practice
- You do not need to buy these books
just for this course!**
- Other handouts and research papers

What this course is not?

- This is not a programming tips and tricks course
 - We will discuss more generic abstract questions
- This is not an introductory course to any specific tool
 - However, we will probably make use of a few

Extending Collaborative Learning



- Make use of the office hours

Office hours: TuTh 4-5 PM KD 302

- You are welcome to **PROVIDE** feedback anytime

Questions?
