CS 610: Programming for Performance

Swarnendu Biswas

Semester 2022-2023-I

CSE, IIT Kanpur

Course Details

- CS 610: Programming for Performance
- Semester 2022-2023-I
- Class hours: MTh 9:00-10:15 AM at KD 101
- Office hours: MTh 10:30-11:30 AM at KD 302
- Webpage: https://www.cse.iitk.ac.in/users/swarnendu/courses/autumn2022-cs610/
  - Follow the course calendar
  - LMS: **REGISTER** for CS 610 on Piazza and mooKIT

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	Email (AT CSE IITK)
Abhinav Kuruma	abhinav
Abhishek Revskar	abhishekdr
Akash Panzade	akashp
Arun KP	kparun
Ashutosh Patel	ashutoshp
Ayush Singh	ayushs
Suvam Basak	suvambasak

Course Outline

- Introduction: Challenges in parallel programming, correctness and performance errors, understanding performance, performance models
- Exploiting spatial and temporal locality with caches, analytical cache miss analysis
- Compiler transformations: Dependence analysis, Loop Transformations
- Shared-memory programming and Pthreads
- Compiler vectorization: vector ISA, auto-vectorizing compiler, vector intrinsics, assembly
- OpenMP: Core OpenMP and Advanced OpenMP
- Parallel Programming Models and Patterns
- Task-based Programming with Intel Threading Building Blocks

Course Outline

- GPGPU programming: GPU architecture and CUDA Programming
- Performance bottleneck analysis: PAPI counters, Using performance analysis tools

- Optional topics
  - Heterogeneous Programming with OpenMP
  - Java Fork-Join Parallelism
  - Concurrent data structures

- We may share additional handouts and research papers
  - Such resources are **part of the syllabus** for exams

Prerequisites

- Good background in the following courses (or equivalent) will help
  - CS 220 (Computer Organization)
  - CS 330 (Operating Systems)
  - CS 335 (Compiler Design)
  - CS 422 (Computer Architecture)

- Programming maturity in C/C++ is desirable
  - We will use OpenMP, TBB, and CUDA extensions
  - Java may be required if we discuss Fork-Join Parallelism

Course Policies

- Online Discussion
  - Be **ON TIME**!
  - Try to FOCUS and PARTICIPATE!
  - Avoid DISTRACTIONS!
  - Keep your mobile phones SILENT
  - Keep your laptops AWAY from you
- Email subjects **SHOULD** start with [CS610]
  - I may not respond otherwise
- Submitting your assignments late will mean losing marks automatically. You will lose 25% for each day that you miss, for up to two days.

Evaluation

Class participation/quizzes/paper critiques	5%
Assignments	40%
Midsem	25%
Endsem	30%

- This tentative allocation may change slightly depending on the strength of the class
- Grading will be relative

Academic Integrity

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

What this course is and is not?

- We will focus on a wide variety of topics related to performance on shared memory systems
- This is not a programming “tips and tricks” course
  - We will discuss more generic abstract concepts
- This is not an introductory course to any specific tool
  - However, we will probably make use of a few

Collaborative Learning

- Make use of Piazza and the office hours
  - We will try to clarify **EVERY REASONABLE** question

Office hours: MTh 10:30-11:30 AM at KD 302

- Slides will primarily be pointers to concepts and materials
  - I will post optional reading material
  - We may have questions from such resources during exams

- You are welcome to **PROVIDE** feedback about the course anytime

Resources

- Optimizing Compilers for Modern Architectures – R. Allen and K. Kennedy
- An Introduction to Parallel Programming – P. Pacheco
- Programming Massively Parallel Processors: A Hands-on Approach – David Kirk Wen-mei W. Hwu
- Intel Threading Building Blocks – James Reinders
- Pro TBB – Michael Voss, Rafael Asenjo, and James Reinders

- Other relevant books, handouts, and research papers

Questions?