Parallel Computer Architecture and Programming
Tsinghua, Summer 2017

Instructors

Kayvon Fatahalian
Carnegie Mellon University
http://www.cs.cmu.edu/~kayvonf/

Wei Xue
Tsinghua University
http://hpc.cs.tsinghua.edu.cn/research/cluster/xuewei.html

Teacher's Assistants

Yong He
Ph.D. student (Carnegie Mellon University)
http://www.csyong.net

Yan Gu
Ph.D. student (Carnegie Mellon University)
http://cs.cmu.edu/~ygu1

Xu Ji
Ph.D. student (Tsinghua)

Ping Xu
M.S. student (Tsinghua)

Course Description

From smart phones, to multi-core CPUs and GPUs, to the world's largest supercomputers and web sites, parallel processing is ubiquitous in modern computing. The goal of this course is to provide a deep understanding of the fundamental principles and engineering trade-offs involved in designing modern parallel computing systems as well as to teach parallel programming techniques necessary to effectively utilize these machines. Because writing good parallel programs requires an understanding of key machine performance characteristics, this course will cover both parallel hardware and software design.

Textbook

There is no required textbook for this course. However, you may find the following textbooks helpful. However, in general these days there are plenty of great parallel programming resources available for free on the web, and I encourage you to read those.

David E. Culler and Jaswinder Pal Singh, with Anoop Gupta
Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann, 1998

John L. Hennessy and David A. Patterson
Computer Architecture, Fifth Edition: A Quantitative Approach. Morgan Kaufmann, 2011

Jason Sanders
CUDA by Example: An Introduction to General-Purpose GPU Programming. 1st Edition. Addison-Wesley. 2010

David Kirk and Wen-mei Hwu
Programming Massively Parallel Processors, Second Edition: A Hands-on Approach. 2nd Edition. Morgan Kauffmann. 2012

Assignments and Homework

There is no final exam for this course.

Programming assignments. Students will complete three programming assignments. These assignments will involve programming in C-like languages for CPUs and GPUs such as C, ISPC, and CUDA. Each assignment will take about one week. **Programming assignment grades will be based on code performance** and will be automatically graded vis script so that students know an estimate of their grade at the time of submission.

Written assignments. Students will complete four written assignments that will provide practice with key course ideas and topics. These assignments will take about three days each, and will graded to provide students feedback. Students will then have one opportunity to correct their original written handin and resubmit corrected assignments for 90% of the original credit. There will be further opportunities for students to earn up to 100% of the original credit.

60% of the course grade will be based on the programming assignments.

40% of the course grade will be based on the written assignments.

Collaboration Policy

Students in this course are absolutely encouraged to talk to each other, to the TAs, to the instructors, or to anyone else about course assignments. Any assistance, though, must be limited to discussion of the problems and sketching general approaches to a solution. Each student must write their own code and produce their own writeup. **Consulting another student's solution, or solutions from the internet, is prohibited on assignments. These and any other form of collaboration constitute cheating.** If you have any question about whether some activity would constitute cheating, just be cautious and ask the instructors before proceeding!

You may not supply code, assignment writeups, or exams you complete during the course to other students in future instances of this course or make these items available (e.g., on the web) for use in future instances of this course (just as you may not use work completed by students who've taken the course previously). We encourage you to use public source control hosts like Github for your assignments, however please be sure to make your programming assignment repositories private.