Stanford CS149, Fall 2020

PARALLEL COMPUTING

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.

Basic Info

Tues/Thurs 2:30-3:50pm
Virtual Course Only
Instructors: **Kayvon Fatahalian** and **Kunle Olukotun**

See the **course info** page for more info on policies and logistics.

Fall 2020 Schedule

Sep 15	**Why Parallelism? Why Efficiency?**	Motivations for parallel chip decisions, challenges of parallelizing code
Sep 17	**A Modern Multi-Core Processor**	Forms of parallelism: multicore, SIMD, threading + understanding latency and bandwidth
Sep 22	**Parallel Programming Abstractions**	Ways of thinking about parallel programs, and their corresponding hardware implementations, ISPC programming
Sep 24	**Parallel Programming Basics**	Thought process of parallelizing a program in data parallel and shared address space models

Sep 29	**Performance Optimization I: Work Distribution and Scheduling** Achieving good work distribution while minimizing overhead, scheduling Cilk programs with work stealing
Oct 01	**Performance Optimization II: Locality, Communication, and Contention** Message passing, async vs. blocking sends/receives, pipelining, increasing arithmetic intensity, avoiding contention
Oct 06	**GPU architecture and CUDA Programming** CUDA programming abstractions, and how they are implemented on modern GPUs
Oct 08	**Data-Parallel Thinking** Data parallel thinking: map, reduce, scan, prefix sum, groupByKey
Oct 13	**Distributed Computing using Spark** Producer-consumer locality, RDD abstraction, Spark implementation and scheduling
Oct 15	**Cache Coherence** Definition of memory coherence, invalidation-based coherence using MSI and MESI, false sharing
Oct 20	**Memory Consistency + Implementation Synchronization** Consistency vs. coherence, relaxed consistency models and their motivation, acquire/release semantics, implementing locks and atomic operations
Oct 22	**Fine-Grained Synchronization and Lock-Free Programming** Fine-grained snychronization via locks, basics of lock-free programming: single-reader/writer queues, lock-free stacks, the ABA problem, hazard pointers
Oct 27	**Midterm Exam** good luck to everyone
Oct 29	**Transactional Memory** Motivation for transactions, design space of transactional memory implementations, lazy-optimistic HTM
Nov 03	**Heterogeneous Parallelism and Hardware Specialization** Energy-efficient computing, motivation for heterogeneous processing, fixed-function processing, FPGAs, mobile SoCs
Nov 05	**Domain-Specific Programming Systems** Motivation for DSLs, case study on Halide image processing DSL
Nov 10	**Parallel Graph Processing Frameworks + How DRAM Works** GraphLab, Ligra, and GraphChi, streaming graph processing, graph compression
Nov 12	**Programming for Hardware Specialization** Performance programming for FPGAs and CGRAs

Nov 17	**Efficiently Evaluating DNNs** Scheduling convlayers, exploiting precision and sparsity, DNN acelerators (e.g., GPU TensorCores, TPU)
Nov 19	**Parallel DNN Training + Course Wrap Up** Enjoy your Winter holiday break!

Programming Assignments

Sep 25	**Assignment 1: Analyzing Parallel Program Performance on a Quad-Core CPU**
Oct 8	**Assignment 2: Scheduling Task Graphs**
Oct 23	**Assignment 3: A Simple Renderer in CUDA**
Nov 10	**Assignment 4: Big Graph Processing in OpenMP**
Nov 19	**Assignment 5: Optional Assignment**

Written Assignments

Oct 6	**Written Assignment 1**
Oct 13	**Written Assignment 2**
Oct 20	**Written Assignment 3**
Nov 5	**Written Assignment 4**
Nov 17	**Written Assignment 5**

Copyright 2020 Stanford University