

Engineering Parallel Software

Kurt Keutzer,

EECS, University of California, Berkeley

and Tim Mattson

Intel Research

Fall 2014

(Subject to change before the start of the course)

As the basic computing device ranging from single cell phones to racks of hardware in cloud computing, parallel processors are emerging as the pervasive computing platform of our time. This course will enable advanced undergraduate students to design, implement, optimize, and verify programs to run on present generations of parallel processors.

There are four principal themes that are pursued in this course:

- Software engineering
- Performance Programming
- Programming in Parallel Languages
- Course project

Software Engineering and Software Architecture

Our approach to this course reflects our view that a well-designed *software architecture* is a key to designing parallel software, and a key to software architecture is design patterns and a pattern language. Our course will use Our Pattern Language as the basis for describing how to design, implement, verify, and optimize parallel programs. Following this approach we will introduce each of the major patterns that are used in developing a high-level architecture of a program. Descriptions of these ten structural and thirteen computational patterns, together with other readings, may be found at: <https://patterns.eecs.berkeley.edu/>.

Performance Programming

Writing efficient parallel programs requires insights into the hardware architecture of contemporary parallel processors as well as an understanding as to how to write efficient code in general. As a result a significant amount of time in the course will be spent on looking “under the hood” of contemporary sequential and multiprocessors and identifying the key architectural details, such as non-uniform memory architecture (NUMA), that are necessary to write high performance code.

Programming in Parallel Languages

Other lectures and laboratories of the course will focus on implementation using contemporary parallel programming languages, verification of parallel software using invariants and testing, and performance tuning and optimization. Particular languages covered typically include OpenMP, MPI, and OpenCL.

Course Projects

The final third of the course will be an open-ended course project. These projects allow students to demonstrate their mastery of the course concepts mentioned above. Students will create their own projects in project teams of 4-6 students.

Prerequisites

Students should have taken, the following or equivalents:

- Basic programming course using Java, C or C++
- Undergraduate course on computer organization
- Linear algebra

It is recommended that students have taken:

- At least one upper division course that includes significant programming assignments (e.g. Compilers, Operating Systems, or Software Engineering)

Course Work and Grading

The course consists of twice-weekly lectures and once-weekly lab sessions. For the first two thirds of the course, there will be a series of programming assignments. There will be two examinations during the course.

Course Staff

Professor: Kurt Keutzer

Guest Lecturer: Tim Mattson, Intel

TA's: Paden Tomasello; Peter Jin

Recommended Course Textbook

Patterns for Parallel Programming, T. Mattson, B. Sanders, B. Massingill, Addison Wesley, 2005.

Course Assignments Will Be Selected From Among this List

1. **Computer Architecture** - Measure L1/L2/L3 bandwidth and latency on our lab machines. Also, investigate measured ILP for a handful of different SGEMM implementations. Performance in MFlops/s increases, but ILP drops. Also serves as a warmup / refresher for the small subset of C++ we use for the lab assignments. Follows the material from lecture 3 (sequential processor performance)
2. **Parallel Matrix Multiply (DGEMM)** - Write naive parallel DGEMM using OMP for loops, OMP tasks, and pthreads. Serves as a simple warm-up for the basic threading libraries. Advanced question on how GCC converts code with OpenMP pragmas into parallel code. Follows the material from lecture 2/4 (parallel programming on shared memory computers)
3. **Optimize Matrix Multiply (DGEMM)** - Optimize the naive parallel matrix multiply for both locality and data parallelism (using SSE2). Students get familiar with SSE2 intrinsics if they want to use them for their final projects. Follows the material from lecture 6/8 (memory subsystem performance)
4. **Introduction to OpenCL** - Students write both VVADD and SGEMM in OpenCL. They will write the kernels. Follows lecture 9 / 10. (Data parallelism and CUDA).
5. **OpenCL + OpenGL** - Students perform a handful of simple graphics operations on an image. Follows lecture 9 / 10. (Data parallelism and CUDA).
6. **Advanced OpenCL** - Students write a reduction routine using the ideas presented in class. They also write array compaction using scan. Follows lecture 9 / 10. (Data parallelism and CUDA).

Syllabus, Fall 2014: Classes are at 2:00 – 3:30PM PDT/PST

<i>Week</i>	<i>Date</i>	<i>What</i>	<i>Topic</i>
Week 1	Tuesday 8/26	No class	
	Thursday 8/28	Lecture 1	First Lecture: Intro, Background, Course Objectives and Course Projects --Keutzer
Week 2	Tuesday 9/2	Lecture 2	A programmer's introduction to parallel computing: Amdahl's law, Concurrency vs. Parallelism, and the jargon of parallel computing. Getting started with OpenMP and Pthreads. --Mattson
	Thursday 9/4	Lecture 3	Sequential Processor Performance: Notions of performance: Insufficiency of Big-O, Example; Pipelining, Superscalar, etc.; Compiler Optimizations; Processor "Speed of Light" --Keutzer
	Monday 9/8	Discussion 1	Intro to the Lab Environment. Assignment 1 goes out.
	Tuesday 9/9	Lecture 4	C++ for Java/C Programmers; Working with OpenMP and Pthreads. Assignment 1 due. Assignment 2 goes out.
	Thursday 9/11	Lecture 5	Sequential Processor Performance Part 2 --Keutzer

	Monday 9/15	Discussion 2	Assignment 1 due. Assignment 2 goes out.
	Tuesday 9/16	Lecture 6	Parallel Processor Architecture --Keutzer
	Thursday 9/18	Lecture 7	Patterns – Another Way to Think About Parallel Programming - Keutzer

	Monday 9/22	Discussion 3	Assignment 2 due. Study for midterm.
	Tuesday 9/23	Lecture 8	Memory optimization and Optimizing Matrix Multiply, --Mattson
	Thursday 9/25	Lecture 9	Synchronization and concurrency issues --Mattson

Week 6	Monday 9/29	Discussion 4	Assignment 3 goes out.
	Tuesday 9/30	Midterm	<u>Midterm 1</u>
	Thursday 10/2	Lecture 10	Data Parallelism --Keutzer

Week 7	Monday 10/6	Discussion 5	Assignment 3 due. Assignment 4 goes out.
	Tuesday 10/7	Lecture 11	CUDA and OpenCL --Jin
	Thurs 10/9	Lecture 12	Structured grid, and MPI --Mattson

Week 8	Monday 10/13	Discussion 6	Assignment 4 due. Assignment 5 goes out.
	Tuesday 10/14	Lecture 13	Mid-semester review/Project proposals due.
	Thursday 10/16	Lecture 14	Parallelizing structural patterns - part 1 - Keutzer

Week 9	Monday 10/20	Discussion 7	Assignment 5 due. Assignment 6 goes out.
	Tuesday 10/21	Lecture 15	Parallelizing structural patterns - part 2--Keutzer
	Thurs 10/23	Lecture 16	Parallelizing logic optimization - part 1 --Keutzer

Week 10	Monday 10/27	Discussion 8	Assignment 6 due. Midterm 2 review and discussion.
	Tuesday 10/28	Lecture 17	Parallelizing logic optimization - part 2 Keutzer
	Thursday 10/30	Midterm 2	<u>Midterm 2</u>

Week 11	Monday 11/3	Discussion 9	Project meetings: show up with evidence of work!
	Tuesday 11/4	Lecture 18	Dense linear algebra - pt1 - Jin
	Thursday 11/6	Lecture 19&20	Dense linear algebra - pt2 - Jin Sparse linear algebra - pt1

Week 12	Monday 11/10	Discussion 10	Project meetings: show up with evidence of work!
	Tuesday 11/11	Holiday	
	Thursday 11/13	Lecture 21	Sparse linear algebra - pt2 - Jin

Week 13	Monday 11/17	Discussion 11	Project meetings: show up with evidence of work!
	Tuesday 11/18	Lecture 23	Speech recognition - pt1 --Keutzer
	Thurs 11/20	Lecture 24	Speech recognition - pt2 --Keutzer

Week 14	Tues 11/25	Lecture 25	Your career in software - Keutzer
---------	---------------	------------	-----------------------------------

Week 15	Monday 12/1	Discussion 12	discuss projects
	Tuesday 12/2	Presentations	Project presentations
	Thurs 12/4	Presentations	Project presentations