Course Description

This course is about writing effective programs to harness the unprecedented power provided by modern parallel computers, so that the programs attain the highest possible levels of performance the machines are capable of. The parallel computers we focus on include multi-core processors as well as clusters and supercomputers made from them. The programming systems and methodologies we learn will include OpenMP, Pthreads, MPI and Charm++. However, the focus of the course is not so much on the mechanics of these programming systems as on how to use them to attain and improve high performance. This performance orientation pervades throughout the course, and is enhanced by several case studies, small enough to understanding the lecture format yet complex enough to illustrate performance issues and trade-offs. The course also teaches an adequate analytical framework for understanding performance, including performance models, scalability analysis, and iso-efficiency.

Course Goals and Objectives

By the end of the course, you will be able to: Write efficient parallel programs for multicore processors and distributed memory machines. Specifically:

- List architectural elements of modern processors and explain their impact on performance
- Analyze sequential performance, count cache misses, optimize cache performance
- Identify if a given loop is parallel and restructure loops to make them parallel (privatize, shared vars, etc), express loops using OpenMP
- Optimize performance using schedule clauses and explicit non-loop parallelism including tasks
- Explain the functionality of MPI primitives (including send/recv variants, collectives, sub-communicators, etc.)
- Describe and characterize the behavior of MPI programs
- Create MPI programs to accomplish a computational task
- Debug MPI programs
- Analyze and derive expressions for completion time of parallel programs based on the alpha-beta cost model
- Analyze scalability of parallel algorithms and derive their iso-efficiency function
- Explain workings of covered parallel algorithms and reason about their efficacy and of variants
- Answer questions about parallel application domains and their characteristics, as well as computational challenges in those domains

Textbook and Readings

There is no required textbook for this course.

Recommended sources and books:

- OpenMP 4.5 standard (from https://www.openmp.org/specifications/)
- MPI 3.1 report (https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf)

Course Outline

This 4-credit hour course is 16 weeks long. You should invest 6-10 hours every week in this course.

Week	Duration	Topic	Relevant Concepts and Techniques
1	1/14 - 1/20	Performance with Complexity	Moore's Law, Parallelism, Latency and Bandwidth, Pipelining, Branch Prediction
2	1/21 - 1/27	Caches and Memory Optimizations	Caches and Cache performance, Prefetching, Virtual Memory, Data Layout
3	1/28 - 2/3	Shared Memory Programming	Vectorization, SIMD, Tools, Shared Memory Machines, Parallel loops and OpenMP
4	2/4 - 2/10	Basic OpenMP	OpenMP parallel loop construct, Sharing of variables, Dependencies and restructuring, Loop schedules, Parallel construct (without "for")
5	2/11 - 2/17	Advanced OpenMP	Synchronization, Critical sections, Sequential consistency, Flush construct, Case studies
6	2/18 - 2/24	Performance Issues in OpenMP	Cache related performance issues, False sharing, Nested parallelism, Explicit dependencies, Tasks
7	2/25 - 3/3	Midterm Exam	
8	3/4 - 3/10	Distributed Memory Programming	Pthreads, C++11 Atomics, Parallel Queues, Distributed Memory Machines, Basic MPI
9	3/11 - 3/17	MPI Collectives	Send/Recv Variants, Collective operations, sub-communicators, parallel Prefix,
10	3/18 - 3/24	Spring Break - no class	

11	3/25 - 3/31	Other Distributed Models	Cost Model, One-sided communication, Hybrid programming (MPI + OpenMP)
12	4/1 - 4/7	Theoretical Models	Charm++
13	4/8 - 4/14	Parallel Algorithms	Sorting algorithms, Algorithms for Broadcast/Reduction and collective operations, Scalability and Isoefficiency
14	4/15 - 4/21	Distributed Parallel Applications	Matrix Multiplication, Interconnection Topologies, Fault Tolerance
15	4/22 - 4/28	Parallel Languages Overview	Parallel Discrete Event Simulations, Combinatorial Search, GPGPUs
16	4/29 - 5/5	Final Exam	Other Parallel languages, future perspectives

Assignment Deadlines

For all assignment deadlines, please refer to the **Course Assignment Deadlines, Late Policy, and Academic Calendar** page.

Elements of This Course

The course is comprised of the following elements:

- **Lecture Videos.** In each week, the concepts you need to know will be presented through a collection of short video lectures. You may stream these videos for playback within the browser by clicking on their titles or download the videos. You may also download the slides that go along with the videos. The videos usually total 1.5 to 3 hours each week. You should generally spend at least the same amount of time digesting content in the video. The actual amount of time needed to digest the content will vary based on your background.
- **Orientation Quiz.** The purpose of the orientation quiz is to ensure that you have gone through the orientation module and acquired the necessary information about the course before you start it. The orientation quiz is a required activity, but it's not part of the course grading. You have unlimited attempts on the orientation quiz. You need to answer all questions correctly in order to pass the orientation quiz.
- **Graded Quizzes**. Each week concludes with a graded quiz. You will be allowed unlimited attempts for each graded quiz with your highest attempt score used towards your final grade. There is no time limit on how long you take to complete each attempt at the quiz. Graded quizzes will be used when calculating your final score in the class.

- **Programming Assignments.** There are 4 total programming assignments in this course. The first one (MP0) requires no coding and is mainly meant to make sure you all have access to course resources (campus cluster and VM-Farm). You may invest 5-7 hours on each of the programming assignments. For more information about the programming assignments, please read the instructions on programming assignment in respective weeks.
- **Homework Assignments**. There are 4 homework assignments.
- **Project.** The project will be a significantly larger programming assignment compared to the four machine problems and will require much more time. It will be assigned towards the end of the semester.
- **Proctored Exams**. There are 2 proctored exams in this class. The exams will be proctored via a proctoring service called ProctorU. For more information about ProctorU and the proctor exams, read the **Proctored Exam** page.

Please note, in order to access course materials and assignments, you will need to pay the Coursera fee $158 ($79 per MOOC equivalent) for this course (a degree course equals to approximately two MOOCs) in addition to the University of Illinois tuition.

Grading Distribution and Scale

Grading Distribution

Assignment	Occurrence	Percent of the Final Grade
Graded Quizzes	Weeks 1-6, 8, 9, 11-14	12%
Programming Assignments	1/27, 2/11, 3/14, 4/14	17% (2%, 4%, 5% ,6%)
Homework	Weeks 2, 4, 8, 14	16% (4% each)
Midterm Exam	Week 7	15%
Final (comprehensive) exam	Week 16	25%
Project	Week 16	15%

Grading Scale

Letter Grade	Percent Needed
A	**85%**

B	**70%**
C	**60%**

Please note: This course will be graded on a curve. I.e. depending on the number of students within each grade range, and the relative difficulty of the components such as exams and project, the instructor may choose a more lenient (but never stricter) scale than the above.

Your final grade will be calculated based on the activities listed in the table below. Your official final course grade will be listed in Enterprise. The course grade you see displayed in Coursera may not match your official final course grade.

Student Code and Policies

A student at the University of Illinois at the Urbana-Champaign campus is a member of a University community of which all members have at least the rights and responsibilities common to all citizens, free from institutional censorship; affiliation with the University as a student does not diminish the rights or responsibilities held by a student or any other community member as a citizen of larger communities of the state, the nation, and the world. See the University of Illinois Student Code for more information.

Academic Integrity

All students are expected to abide by the campus regulations on academic integrity found in the Student Code of Conduct. These standards will be enforced and infractions of these rules will not be tolerated in this course. Sharing, copying, or providing any part of a homework solution or code is an infraction of the University's rules on academic integrity. We will be actively looking for violations of this policy in homework and project submissions. Any violation will be punished as severely as possible with sanctions and penalties typically ranging from a failing grade on this assignment up to a failing grade in the course, including a letter of the offending infraction kept in the student's permanent university record.

Again, a good rule of thumb: *Keep every typed word and piece of code your own*. If you think you are operating in a gray area, you probably are. If you would like clarification on specifics, please contact the course staff.

Disability Accommodations

Students with learning, physical, or other disabilities requiring assistance should contact the instructor as soon as possible. If you're unsure if this applies to you or think it may, please contact the instructor and Disability Resources and Educational Services (DRES) as soon as possible. You can contact DRES at 1207 S. Oak Street, Champaign, via phone at (217) 333-1970, or via email at disability@illinois.edu.

;