EE599 – Principles of Concurrency and Parallelism

Spring 2017

Instructor: Xuehai Qian

E-mail: xuehai.qian@usc.edu; Office: EEB 204; Tel: (213) 740-4459

Office hours: Wed/Friday 11am-12pm

Syllabus

1. Overview

The course will approach multiprocessor programming from two complementary directions. In the first part of the course, we focus on foundations: what do our programs and machines need to provide in order to ensure that concurrent programs do what we expect. We use an idealized model of computation in which multiple concurrent threads manipulate a set of shared objects. This model is essentially the model presented by standard Java™ or C++ threads packages.

The foundations section is intended to build up the reader's intuition and confidence in understanding and reasoning about concurrency. We approach this goal using examples, counter-examples, models, and exercises. These elements are laid out in a structured and progressive manner, from simple machine instructions to powerful universal constructions, the equivalent of Turing machines for multiprocessors.

The second part of the course will be concerned with performance. Reasoning about the performance of concurrent programs and data structures is very different in flavor from reasoning about their sequential counterparts. Sequential programming is based on a well-established and well-understood set of abstractions. There is little or no need to understand the specifics of the underlying architecture. In multiprocessor programming, by contrast, there are no such well-established abstractions. It is impossible to reason effectively about the performance of a concurrent data structure without understanding the fundamentals of the underlying architecture.

The performance part of the course will revisit many of the issues first raised in the foundations section, but in a more realistic model that exposes those aspects of the underlying architecture that most influence performance. The course then goes through a sequence of fundamental data structures, the concurrent analogs of the data structures found in any undergraduate data structures course, and a few coordination structures that are unique to the world of multithreaded computation. These data structures are introduced in an incremental way, each one extending the techniques developed for its predecessors. Each of these data structures is useful in and of itself as a reference. Moreover, by the end, the student will have built up a solid understanding of the fundamentals of concurrent data structure design, and should be well-prepared to design and implement his or her own concurrent data structures.

Our hope is that at the end of the course students will have a basic understanding of both the foundations and the practice of multiprocessor and multicore programming.

2. Textbooks

1. Maurice Herlihy and Nir Shavit: “The Art of Multiprocessor Programming, Revised 1st Edition” Morgan Kaufmann, 2012. ISBN: 978-0123973375.

3. Prerequisites:

The basic understanding of Java and share-memory multiprocessor architecture are preferred.

Office hours:
Wed/Fri: 11-noon

We do not provide information about the graders.

Per department policies, complaints about homework and project grading issues should be addressed to the instructor.

Please resolve all grading issues promptly. Do not procrastinate!

4. Course Work:

1. Homework: There will be 4 homeworks. Homework is due two weeks from assignment. We use paperless homework submission, grading and return. Homework must be in PDF (scan it if you need to). If you handwrite your solutions, please make sure that your writing is legible and pages are in order. Late homework will incur 25% penalty of maximum grade for one-day delay or 50% penalty of maximum grade for two-day delay. No homework will be accepted after that.
2. Programming assignments: There will be two programming assignments. The goal of these assignment is to implement the principles from the course and apply them to real problems. For the first project, the students implement the data structures in a few chapters in the book, test them and write report on the implementation and evaluation. We will provide account in the cluster in our research group so that each student will be able to test the program in the same platform. For the second project, the students will work on TinySTM, a software transactional memory system.
3. Presentations: each student should read a paper and present it to the class. Each student should present the first project to the class.
4. Mid-term/Final Exam time: TBD

Please understand that there will be NO make-up exam, except in cases of personal medical emergency certified by a physician, or of personal accident. Other requests will be denied.

5. Grading Policy

a. 4 Homeworks: 32%

b. Presentation: 5%

c. Attendance: 3%

d. Projects: 20% (10% each)

e. Mid-Term Exam: 10%

f. Final Exam: 30%

There is no possibility to earn extra credit in this class. PLEASE DON'T ASK! The final grade will be computed as announced. There is already a lot of work in this class! Do the

best you can on each component of the course work you are graded on.

6. Statement for Students with Disabilities

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.-5:00 p.m., Monday through Friday. The phone number for DSP is (213) 740-0776.

7. Statement on Academic Integrity

USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one own academic work from misuse by others and to avoid using another's work as one's own. All students are expected to understand and abide by these principles. Please consult http://www.usc.edu/student-affairs/SJACS/pages/students/academic_integrity.html Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty.

8. Course Schedule:

Lecture	Date	Topic--Remarks
Lect 1	1/11	Administration-Introduction
Lect 2	1/13	Introduction
Lect 3	1/18	Multiprocessor Architecture Basics
Lect 4	1/20	Mutual Exclusion
Lect 5	1/25	Mutual Exclusion
Lect 6	1/27	Concurrent Objects
Lect 7	2/1	Concurrent Objects
Lect 8	2/3	Fundamental of Shared Memory
Lect 9	2/8	Fundamental of Shared Memory
Lect 10	2/10	Relative Power of Primitive Synchronization Ops
Lect 11	2/15	Relative Power of Primitive Synchronization Ops
Lect 12	2/17	Consensus
Lect 13	2/22	Consensus
Lect 14	3/1	Spin Lock
Lect 15	3/3	Spin Lock
Lect 16	3/8	Linked List
Lect 17	3/10	Linked List
Lect 18	3/15	Spring Recess
	3/17	Spring Recess
Lect 19	3/22	**Mid-Term: TBD**
Lect 20	3/24	Counting
Lect 21	3/29	Barrier/Future and Work Distribution
Lect 22	3/31	Transactional Memory
Lect 23	4/5	Project Presentation
Lect 24	4/7	Project Presentation
Lect 25	4/12	Project Presentation
Lect 26	4/14	Project Presentation
Lect 27	4/19	Paper Presentation
	4/21	Paper Presentation
Lect 28	4/26	Paper Presentation
	4/28	Paper Presentation
		FINAL: TBD