COMP322-Spring-2016

Home	Office Hours	HJlib Info	edX site	Other Resources

COMP 322: Fundamentals of Parallel Programming (Spring 2016)

Instructor:	Prof. Vivek Sarkar, DH 3080	**Head TA:**	Max Grossman
Admin Assistant:	Annepha Hurlock, annepha@rice.edu, DH 3080, 713-348-5186	**Graduate TAs:**	Prasanth Chatarasi, Arghya Chatterjee, Yuhan Peng, Jonathan Sharman
Co-Instructor:	Dr. Shams Imam	**Undergraduate TAs:**	Prudhvi Boyapalli, Peter Elmers, Nicholas Hanson-Holtry, Ayush Narayan, Timothy Newton, Alitha Partono, Tom Roush, Hunter Tidwell, Bing Xue
Piazza site:	https://piazza.com/class/iirz0u74egl2q9 (Piazza is the preferred medium for all course communications, but you can also send email to comp322-staff at rice dot edu if needed)	**Cross-listing:**	ELEC 323
Lecture location:	Herzstein Hall 210	**Lecture times:**	MWF 1:00pm - 1:50pm (followed by office hours in Duncan Hall 3092 during 2pm - 3pm)
Lab location:	DH 1064 (Section A01), DH 1070 (Section A02)	**Lab times:**	Wednesday, 07:00pm - 08:30pm

Course Syllabus

A summary PDF file containing the course syllabus for the course can be found here . Much of the syllabus information is also included below in this course web site, along with some additional details that are not included in the syllabus.

Course Objectives

The primary goal of COMP 322 is to introduce you to the fundamentals of parallel programming and parallel algorithms, by following a pedagogic approach that exposes you to the intellectual challenges in parallel software without enmeshing you in the jargon and lower-level details of today's parallel systems. A strong grasp of the course fundamentals will enable you to quickly pick up any specific parallel programming system that you may encounter in the future, and also prepare you for studying advanced topics related to parallelism and concurrency in courses such as COMP 422.

The desired learning outcomes fall into three major areas (course modules):

1) Parallelism: creation and coordination of parallelism (async, finish), abstract performance metrics (work, critical paths), Amdahl's Law, weak vs. strong scaling, data races and determinism, data race avoidance (immutability, futures, accumulators, dataflow), deadlock avoidance, abstract vs. real performance (granularity, scalability), collective & point-to-point synchronization (phasers, barriers), parallel algorithms, systolic algorithms.

2) Concurrency: critical sections, atomicity, isolation, high level data races, nondeterminism, linearizability, liveness/progress guarantees, actors, request-response parallelism, Java Concurrency, locks, condition variables, semaphores, memory consistency models.

3) Locality & Distribution: memory hierarchies, locality, cache affinity, data movement, message-passing (MPI), communication overheads

(bandwidth, latency), MapReduce, accelerators, GPGPUs, CUDA, OpenCL.

To achieve these learning outcomes, each class period will include time for both instructor lectures and in-class exercises based on assigned reading and videos. The lab exercises will be used to help students gain hands-on programming experience with the concepts introduced in the lectures.

To ensure that students gain a strong knowledge of parallel programming foundations, the classes and homeworks will place equal emphasis on both theory and practice. The programming component of the course will mostly use the Habanero-Java Library (HJ-lib) pedagogic extension to the Java language developed in the Habanero Extreme Scale Software Research project at Rice University. The course will also introduce you to real-world parallel programming models including Java Concurrency, MapReduce, MPI, OpenCL and CUDA. An important goal is that, at the end of COMP 322, you should feel comfortable programming in any parallel language for which you are familiar with the underlying sequential language (Java or C). Any parallel programming primitives that you encounter in the future should be easily recognizable based on the fundamentals studied in COMP 322.

Prerequisite

The prerequisite course requirements are COMP 182 and COMP 215. COMP 322 should be accessible to anyone familiar with the foundations of sequential algorithms and data structures, and with basic Java programming. COMP 321 is also recommended as a co-requisite.

Textbooks

There are no required textbooks for the class. Instead, lecture handouts are provided for each module as follows. The links to the latest versions on Owlspace are included below:

- Module 1 handout (*Parallelism*)
- Module 2 handout (*Concurrency*)
- Module 3 handout (*Distribution and Locality*)

You are expected to read the relevant sections in each lecture handout before coming to the lecture. We will also provide a number of references in the slides and handouts.

There are also a few optional textbooks that we will draw from quite heavily. You are encouraged to get copies of any or all of these books. They will serve as useful references both during and after this course:

- Fork-Join Parallelism with a Data-Structures Focus (FJP) by Dan Grossman (Chapter 7 in Topics in Parallel and Distributed Computing)
- Java Concurrency in Practice by Brian Goetz with Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes and Doug Lea
- Principles of Parallel Programming by Calvin Lin and Lawrence Snyder
- The Art of Multiprocessor Programming by Maurice Herlihy and Nir Shavit

Past Offerings of COMP 322

- Spring 2015 (Rice University)
- Spring 2014 (Rice University)
- Spring 2013 (Rice University)
- Fall 2012 (Harvey Mudd College CS 181E, half-semester class, co-instructor: Prof. Ran Libeskind-Hadas)
- Spring 2012 (Rice University)
- Spring 2011 (Rice University)
- Fall 2009 (Rice University)

Lecture Schedule

worksheet13

worksheet28

Week	Day	Date (2016)	Topic	Assigned Reading	Assigned Videos (Quizzes due by Friday of each week)	In-class Worksheets	Slides	Work Assigned	Work Due
1	Mon	Jan 11	Lecture 1: Task Creation and Termination (Async, Finish)	Module 1: Section 1.1	Topic 1.1 Lecture, Topic 1.1 Demonstration	worksheet 1	lec1-slides		
	Wed	Jan 13	Lecture 2: Computation Graphs, Ideal Parallelism	Module 1: Sections 1.2, 1.3	Topic 1.2 Lecture, Topic 1.2 Demonstration, Topic 1.3 Lecture, Topic 1.3 Demonstration	worksheet 2	lec2-slides		
	Fri	Jan 15	Lecture 3: Abstract Performance Metrics, Multiprocessor Scheduling	Module 1: Section 1.4	Topic 1.4 Lecture, Topic 1.4 Demonstration	worksheet 3	lec3-slides	**Homework 1** **(2 weeks)**	**Lecture & demo quizzes for topics 1.1, 1.2, 1.3, 1.4**
2	Mon	Jan 18	No lecture, School Holiday (Martin Luther King, Jr. Day)						
	Wed	Jan 20	Lecture 4: Parallel Speedup and Amdahl's Law	Module 1: Section 1.5	Topic 1.5 Lecture, Topic 1.5 Demonstration	worksheet 4	lec4-slides		
	Fri	Jan 22	Lecture 5: Future Tasks, Functional Parallelism	Module 1: Section 2.1	Topic 2.1 Lecture , Topic 2.1 Demonstration	worksheet 5	lec5-slides		**Lecture & demo quizzes for topics 1.5, 2.1 (topic 1.6 is optional)**

3	Mon	Jan 25	Lecture 6: Memoization	Module 1: Section 2.2	Topic 2.2 Lecture , Topic 2.2 Demonstration	worksheet 6	lec6-slides	BinomialCoefficient.java Worksheet5.java	
	Wed	Jan 27	Lecture 7: Finish Accumulators	Module 1: Section 2.3	Topic 2.3 Lecture , Topic 2.3 Demonstration	worksheet 7	lec7-slides		
	Fri	Jan 29	Lecture 8: Data Races, Functional & Structural Determinism	Module 1: Sections 2.5, 2.6	Topic 2.5 Lecture , Topic 2.5 Demonstration, Topic 2.6 Lecture , Topic 2.6 Demonstration	worksheet 8	lec8-slides	**Homework 2** **Homework 2 JARs (optional)** **(2 weeks)**	**Homework 1, Lecture & demo quizzes for topics 2.2, 2.3, 2.5, 2.6**
4	Mon	Feb 01	Lecture 9: Map Reduce	Module 1: Section 2.4	Topic 2.4 Lecture , Topic 2.4 Demonstration	worksheet 9	lec9-slides		
	Wed	Feb 03	Lecture 10: Java's Fork/Join Library	FJP chapter: Sections 7.3 & 7.5		worksheet 10	lec10-slides	ArraySum.java ArraySumFourWay.java	
	Fri	Feb 05	Lecture 11: Loop-Level Parallelism, Parallel Matrix Multiplication, Iteration Grouping (Chunking)	Module 1: Sections 3.1, 3.2, 3.3	Topic 3.1 Lecture , Topic 3.1 Demonstration , Topic 3.2 Lecture, Topic 3.2 Demonstration, Topic 3.3 Lecture , Topic 3.3 Demonstration	worksheet 11	lec11-slides		**Lecture & demo quizzes for topics 2.4, 3.1, 3.2, 3.3**
5	Mon	Feb 08	Lecture 12: Barrier Synchronization	Module 1: Section 3.4	Topic 3.4 Lecture , Topic 3.4 Demonstration	worksheet 12	lec12-slides		

	Wed	Feb 10	Lecture 13: Iterative Averaging Revisited, SPMD pattern	Module 1: Sections 3.5, 3.6	Topic 3.5 Lecture , Topic 3.5 Demonstration , Topic 3.6 Lecture, Topic 3.6 Demonstration	worksheet13	lec13-slides	Worksheet12.java	
	Fri	Feb 12	Lecture 14: Data-Driven Tasks and Data-Driven Futures	Module 1: Section 4.5	Topic 4.5 Lecture , Topic 4.5 Demonstration	worksheet14	lec14-slides	**Homework 3** **(5 weeks, with two intermediate checkpoints)**	**Homework 2, Lecture & demo quizzes for topics ~~3.4~~ , 3.5, 3.6, 4.5**
6	Mon	Feb 15	Lecture 15: Phasers, Point-to-point Synchronization	Module 1: Sections 4.2, 4.3	Topic 4.2 Lecture , Topic 4.2 Demonstration, Topic 4.3 Lecture, Topic 4.3 Demonstration	worksheet15	lec15-slides		
	Wed	Feb 17	Lecture 16: Pipeline Parallelism , Signal Statement, Fuzzy Barriers	Module 1: Sections 4.4, 4.1	Topic 4.4 Lecture , Topic 4.4 Demonstration, Topic 4.1 Lecture , Topic 4.1 Demonstration,	worksheet16	lec16-slides		
	Fri	Feb 19	Lecture 17: Abstract vs. Real Performance			worksheet17	lec17-slides		**Lecture & demo quizzes for topics 4.1, 4.2, 4.3, 4.4**
7	Mon	Feb 22	Lecture 18: Midterm Summary				lec18-slides		

	Wed	Feb 24	Midterm Review (Interactice Q&A using PollEverywhere)					**Exam 1 held during lab time (7:00pm - 10:00pm), scope of exam limited to lectures 1-18**	
	Fri	Feb 26	Lecture 19: Task Scheduling Policies		Topic 4.6 Lecture , Topic 4.6 Demonstration	worksheet 19	lec19-slides	Lec19HelpFirstWorkStealing.java	**Homework 3 Checkpoint-1, Lecture & demo quizzes for topic 4.6**
-	M-F	Feb 29-Mar 04	Spring Break						
8	Mon	Mar 07	Lecture 20: Critical sections, Isolated construct, Parallel Spanning Tree algorithm (start of Module 2)	Module 2: Sections 5.1, 5.2, 5.3, 5.6	Topic 5.1 Lecture, Topic 5.1 Demonstration, Topic 5.2 Lecture, Topic 5.2 Demonstration, Topic 5.3 Lecture, Topic 5.3 Demonstration	worksheet 20	lec20-slides		
	Wed	Mar 09	Lecture 21: Atomic variables, Read-Write Isolation	Module 2: Sections 5.4, 5.5	Topic 5.4 Lecture, Topic 5.4 Demonstration, Topic 5.5 Lecture, Topic 5.5 Demonstration, Topic 5.6 Lecture, Topic 5.6 Demonstration	worksheet 21	lec21-slides		

	Fri	Mar 11	Lecture 22: Parallelism in Java Streams, Parallel Prefix Sums			worksheet 22	lec22-slides		**Homework 3 Checkpoint-2, Lecture & demo quizzes for topics 5.1 to 5.6**
9	Mon	Mar 14	Lecture 23: Java Threads, Java synchronized statement		Topic 7.1 Lecture, Topic 7.2 Lecture	worksheet 23	lec23-slides		
	Wed	Mar 16	Lecture 24: Java synchronized statement (contd), wait/notify		Topic 7.3 Lecture	worksheet 24	lec24-slides		
	Fri	Mar 18	Lecture 25: Concurrent Objects, Linearizability of Concurrent Objects		Topic 7.4 Lecture	worksheet 25	lec25-slides		**Homework 3, Lecture quizzes for topics 7.1 - 7.4**
10	Mon	Mar 21	Lecture 26: Linearizability (contd), Java locks		Topic 7.3 Lecture (recap), Topic 7.4 Lecture (recap)	worksheet 26	lec26-slides	**Homework 4** **(3 weeks, with one intermediate checkpoint)**	
	Wed	Mar 23	Lecture 27: Parallel Design Patterns, Safety and Liveness Properties		Topic 7.5 Lecture	worksheet 27	lec27-slides		

	Fri	Mar 25	Lecture 28: Actors		Topic 6.1 Lecture , Topic 6.1 Demonstration , Topic 6.2 Lecture, Topic 6.2 Demonstration, Topic 6.3 Lecture, Topic 6.3 Demonstration	worksheet 28	lec28-slides		**Lecture & demo quizzes for topics 7.5**
11	Mon	Mar 28	Lecture 29: Actors (contd)		Topic 6.4 Lecture , Topic 6.4 Demonstration , Topic 6.5 Lecture, Topic 6.5 Demonstration, Topic 6.6 Lecture, Topic 6.6 Demonstration	worksheet 29	lec29-slides	Lec29Slide2ThreadRing.java Lec29Slide4EchoActor.java Lec29Slide6Pipeline.java Lec29Slide15ReqReplyActor.java Lec29Slide15SyncReplyActor.java	
	Wed	Mar 30	Lecture 30: Java Synchronizers, Dining Philosophers Problem		Topic 7.6 Lecture	worksheet 30	lec30-slides		
-	Fri	Apr 01	Midterm Recess						**Lecture quiz for topic 7.6**
12	Mon	Apr 04	Lecture 31: Eureka-style Speculative Task Parallelism			worksheet 31	lec31-slides		**Homework 4 Checkpoint-1**
	Wed	Apr 06	Lecture 32: Task Affinity with Places (start of Module 3)			worksheet 32	lec32-slides		

	Fri	Apr 08	Lecture 33: Message Passing Interface (MPI)			worksheet 33	lec33-slides		
13	Mon	Apr 11	Lecture 34: Message Passing Interface (MPI, contd)			worksheet 34	lec34-slides		**Homework 4**
	Wed	Apr 13	Lecture 35: GPU Computing			worksheet 35	lec35-slides	**Homework 5** **(Due April 22nd, with automatic extension till May 2nd after which slip days may be used)**	
	Fri	Apr 15	Lecture 36: Partitioned Global Address Space (PGAS) programming models			worksheet 36	lec36-slides		
14	Mon	Apr 18	NO CLASS TODAY DUE TO INCLEMENT WEATHER						
	Wed	Apr 20	Lecture 37: Apache Spark framework			worksheet 37	lec37-slides		
	Fri	Apr 22	Lecture 38: Course Review (lectures 20-37), Last day of classes				lec38-slides		**Homework 5 (automatic extension till May 2nd)**

-	Mon	Apr 25	Review session / Office Hours, 1pm - 3pm, Herzstein 212 (different room from usual)						
-	Wed	Apr 27	Review session / Office Hours, 1pm - 3pm, Herzstein 212 (different room from usual)						
-	Thu	Apr 29	Review session / Office Hours, 1pm - 3pm, Herzstein 212 (different room from usual)						
-	Tue	May 3	**Scheduled final exam in Herzstein Hall Auditorium, 9am - 12noon, May 3rd (Exam 2 – scope of exam limited to lectures 20-37)**						

Lab Schedule

Lab #	Date (2016)	Topic	Handouts	Code Examples
0		Infrastructure Setup	lab0-handout	-
1	Jan 13	Async-Finish Parallel Programming	lab1-handout, lab1-slides	lab_1.zip

2	Jan 20	Abstract performance metrics with async & finish	lab2-handout, lab2-slides	lab_2.zip
3	Jan 27	DIY HJ-lib Programming, Futures, HJ-Viz	lab3-handout, lab3-slides	lab_3.zip
4	Feb 03	Finish Accumulators and Loop-Level Parallelism	lab4-handout and lab4-slides	lab_4.zip
5	Feb 10	Loop Chunking and Barrier Synchronization	lab5-handout and lab5-slides	lab_5.zip
6	Feb 17	Data-Driven Futures and Phasers	lab6-handout	lab_6.zip
-	Feb 24	No lab this week — Exam 1	-	-
-	Mar 02	No lab this week — Spring Break	-	-
7	Mar 09	Isolated Statement and Atomic Variables	lab7-handout	
8	Mar 16	Java Threads	lab8-handout	
9	Mar 23	Java Locks	lab9-handout	
10	Mar 30	Actors and Selectors	lab10-handout	
11	Apr 06	Eureka-style Speculative Task Parallelism	lab11-handout	
12	Apr 13	Message Passing Interface (MPI)	lab12-handout	
13	Apr 20	Apache Spark	lab13-handout	

Grading, Honor Code Policy, Processes and Procedures

Grading will be based on your performance on five homeworks (weighted 40% in all), two exams (weighted 40% in all), weekly lab exercises (weighted 10% in all), and class participation including worksheets, in-class Q&A, Piazza participation, and online quizzes (weighted 10% in all).

The purpose of the homeworks is to train you to solve problems and to help deepen your understanding of concepts introduced in class. Homeworks are due on the dates and times specified in the course schedule. Please turn in all your homeworks using the subversion system set up for the class. Homework is worth full credit when turned in on time. No late submissions (other than those using slip days mentioned below) will be accepted.

As in COMP 321, all students will be given 3 slip days to use throughout the semester. When you use a slip day, you will receive up to 24 additional hours to complete the assignment. You may use these slip days in any way you see fit (3 days on one assignment, 1 day each on 3 assignments, etc.). If you use slip days, you must submit a SLIPDAY.txt file in your SVN homework folder before the actual submission deadline indicating the number of slip days that you plan to use. Other than slip days, no extensions will be given unless there are exceptional circumstances (such as severe sickness, not because you have too much other work). Such extensions must be requested and approved by the instructor (via e-mail, phone, or in person) before the due date for the assignment. Last minute requests are likely to be denied. If you do receive an extension from the instructor, please indicate this by placing an EXTENSION.txt file in your SVN homework folder before the actual submission deadline indicating the date that the extension was granted by the instructor as well as the length of the extension.

Labs must be checked off by a TA prior to the start of the lab the following week.

Worksheets are due by the beginning of the class after they are distributed, so that solutions to the worksheets can be discussed.

You will be expected to follow the Honor Code in all homeworks and exams. All submitted homeworks are expected to be the result of your individual effort. You are free to discuss course material and approaches to homework problems with your other classmates, the teaching

assistants and the professor, but you should never misrepresent someone else's work as your own. If you use any material from external sources, you must provide proper attribution (as shown here). Exams 1 and 2 test your individual understanding and knowledge of the material. Exams are closed-book, and collaboration on exams is strictly forbidden. Finally, it is also your responsibility to protect your homeworks and exams from unauthorized access.

Graded homeworks will be returned to you via email, and exams as marked-up hardcopies. If you believe we have made an error in grading your homework or exam, please bring the matter to our attention within one week.

Accommodations for Students with Special Needs

Students with disabilities are encouraged to contact me during the first two weeks of class regarding any special needs. Students with disabilities should also contact Disabled Student Services in the Ley Student Center and the Rice Disability Support Services.