COMP322

Home	Office Hours	HJlib Info	edX site	Autograder Guide	Other Resources

COMP 322: Fundamentals of Parallel Programming (Spring 2018)

Instructor:	Mackale Joyner, DH 2071	**Head TA:**	Abbey Baker
Co-Instructor:	Zoran Budimli, DH 3081	**Graduate TAs:**	Jonathan Sharman, Srdjan Milakovic
Admin Assistant:	Annepha Hurlock, annepha@rice.edu, DH 3122, 713-348-5186	**Undergraduate TAs:**	Ashok Sankaran, Austin Bae, Avery Whitaker, Aydin Zanager, Eduard Danalache, Frank Chen, Hamza Nauman, Harrison Brown, Jahid Adam, Jeemin Sim, Kitty Cai, Madison Lewis, Ryan Han, Teju Manchenella, Victor Gonzalez, Victoria Nazari
Piazza site:	https://piazza.com/class/j3w0pi8pl9s8s (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:	Sewall Hall 301	**Lecture times:**	MWF 1:00pm - 1:50pm
Lab locations:	Sewall Hall 301	**Lab times:**	Thursday, 4:00pm - 4:50pm

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 and Other Resources

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

- Module 1 handout (*Parallelism*)
- Module 2 handout (*Concurrency*)
- There is no lecture handout for Module 3 (*Distribution and Locality*). The instructors will refer you to optional resources to supplement the lecture slides and videos.

There are also a few optional textbooks that we will draw from during the course. 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

Finally, here are some additional resources that may be helpful for you:

- Slides titled "MPI-based Approaches for Java" by Bryan Carpenter

Lecture Schedule

Week	Day	Date (2018)	Lecture	Assigned Reading	Assigned Videos (see Canvas site for video links)	In-class Worksheets	Slides	Work Assigned	Work Due
1	Mon	Jan 08	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 10	Lecture 2: Computation on 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	**Homework 1**	
	Fri	Jan 12	Lecture 3: Abstract Performance Metrics, Multiprocessor Scheduling	Module 1: Section 1.4	Topic 1.4 Lecture, Topic 1.4 Demonstration	worksheet 3	lec3-slides		
2	Mon	Jan 15	No lecture, School Holiday (Martin Luther King, Jr. Day)						
	Wed	Jan 17	No lecture, Rice closed due to weather					**Quiz for Unit 1**	
	Fri	Jan 19	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		
3	Mon	Jan 22	Lecture 5: Future Tasks, Functional Parallelism ("Back to the Future")	Module 1: Section 2.1	Topic 2.1 Lecture, Topic 2.1 Demonstration	worksheet 5	lec5-slides		
	Wed	Jan 24	Lecture 7: Finish Accumulators	Module 1: Section 2.3	Topic 2.3 Lecture, Topic 2.3 Demonstration	worksheet 7	lec7-slides	**Homework 2**	**Homework 1**

	Fri	Jan 26	Lecture 8: Memoization, Map Reduce	Module 1: Section 2.2 & 2.4	Topic 2.2 Lecture, Topic 2.2 Demonstration, Topic 2.4 Lecture, Topic 2.4 Demonstration	worksheet 8	lec8-slides		**Quiz for Unit 1**
4	Mon	Jan 29	Lecture 9: 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 9	lec9-slides		
	Wed	Jan 31	Lecture 10: Java's Fork/Join Library	Module 1: Sections 2.7, 2.8	Topic 2.7 Lecture, Topic 2.8 Lecture,	worksheet 10	lec10-slides		
	Fri	Feb 02	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		
5	Mon	Feb 05	Lecture 12: Barrier Synchronization	Module 1: Section 3.4	Topic 3.4 Lecture , Topic 3.4 Demonstration	worksheet 12	lec12-slides		
	Wed	Feb 07	Lecture 13: Parallelism in Java Streams, Parallel Prefix Sums			worksheet13	lec13-slides	**Homework 3 (includes two intermediate checkpoints)**	**Homework 2**
-	Fri	Feb 09	Spring Recess						**Quiz for Unit 2**

6	Mon	Feb 12	Lecture 14: 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	worksheet 14	lec14-slides		
	Wed	Feb 14	Lecture 15: Data-Driven Tasks, Point-to-Point Synchronization with Phasers	Module 1: Sections 4.5, 4.2, 4.3	Topic 4.5 Lecture Topic 4.5 Demonstration, Topic 4.2 Lecture , Topic 4.2 Demonstration, Topic 4.3 Lecture, Topic 4.3 Demonstration	worksheet 15	lec15-slides		
	Fri	Feb 16	Lecture 16: Phasers Review	Module 1: Sections 4.2	Topic 4.2 Lecture , Topic 4.2 Demonstration	worksheet 16	lec16-slides		**Quiz for Unit 3**
7	Mon	Feb 19	Lecture 17: Midterm Summary				lec17-slides		
	Wed	Feb 21	Midterm Review (interactive Q&A)						
	Fri	Feb 23	Lecture 18: Abstract vs. Real Performance			worksheet 18	lec18-slides		**Homework 3, Checkpoint-1**
8	Mon	Feb 26	Lecture 19: 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,	worksheet 19	lec19-slides		

	Wed	Feb 28	Lecture 20: Critical sections, Isolated construct, Parallel Spanning Tree algorithm, Atomic variables (start of Module 2)	Module 2: Sections 5.1, 5.2, 5.3, 5.4, 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, Topic 5.4 Lecture, Topic 5.4 Demonstration, Topic 5.6 Lecture, Topic 5.6 Demonstration	worksheet 20	lec20-slides		
	Fri	Mar 02	Lecture 21: Read-Write Isolation, Review of Phasers	Module 2: Section 5.5	Topic 5.5 Lecture, Topic 5.5 Demonstration	worksheet 21	lec21-slides		**Quiz for Unit 4**
9	Mon	Mar 05	Lecture 22: Actors	Module 2: 6.1, 6.2	Topic 6.1 Lecture , Topic 6.1 Demonstration , Topic 6.2 Lecture, Topic 6.2 Demonstration	worksheet 22	lec22-slides		

	Wed	Mar 07	Lecture 23: Actors (contd)	Module 2: 6.3, 6.4, 6.5, 6.6	Topic 6.3 Lecture, Topic 6.3 Demonstration, 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 23	lec23-slides		**Homework 3, Checkpoint-2**
	Fri	Mar 09	Lecture 24: Java Threads, Java synchronized statement	Module 2: 7.1, 7.2	Topic 7.1 Lecture, Topic 7.2 Lecture	worksheet 24	lec24-slides		**Quiz for Unit 5**
-	M-F	Mar 12 - Mar 16	Spring Break						
10	Mon	Mar 19	Lecture 25: Java synchronized statement (contd), wait/notify	Module 2: 7.2	Topic 7.2 Lecture	worksheet 25	lec25-slides		
	Wed	Mar 21	Lecture 26: Java Locks, Linearizability of Concurrent Objects	Module 2: 7.3, 7.4	Topic 7.3 Lecture, Topic 7.4 Lecture	worksheet 26	lec26-slides	**Homework 4** **(includes one intermediate checkpoint)**	**Homework 3 (all)**
	Fri	Mar 23	Lecture 27: Safety and Liveness Properties, Java Synchronizers, Dining Philosophers Problem	Module 2: 7.5, 7.6	Topic 7.5 Lecture, Topic 7.6 Lecture	worksheet 27	lec27-slides		**Quiz for Unit 6**

11	Mon	Mar 26	Lecture 28: Message Passing Interface (MPI), (start of Module 3)		Topic 8.1 Lecture, Topic 8.2 Lecture, Topic 8.3 Lecture,	worksheet 28	lec28-slides		
	Wed	Mar 28	Lecture 29: Message Passing Interface (MPI, contd)		Topic 8.4 Lecture, Topic 8.5 Lecture, Topic 8 Demonstration Video	worksheet 29	lec29-slides		
	Fri	Mar 30	Lecture 30: Distributed Map-Reduce using Hadoop and Spark frameworks		Topic 9.1 Lecture (optional, overlaps with video 2.4), Topic 9.2 Lecture, Topic 9.3 Lecture	worksheet 30	lec30-slides		**Quiz for Unit 7**
12	Mon	Apr 02	Lecture 31: TF-IDF and PageRank Algorithms with Map-Reduce		Topic 9.4 Lecture, Topic 9.5 Lecture, Unit 9 Demonstration	worksheet 31	lec31-slides		
	Wed	Apr 04	Lecture 32: Partitioned Global Address Space (PGAS) programming models			worksheet 32	lec32-slides		**Homework 4 Checkpoint-1**
	Fri	Apr 06	Lecture 33: Combining Distribution and Multithreading		Lectures 10.1 - 10.5, Unit 10 Demonstration (all videos optional – unit 10 has no quiz)	worksheet 33	lec33-slides		**Quiz for Unit 8**
13	Mon	Apr 09	Lecture 34: Task Affinity with Places			worksheet 34	lec34-slides		

	Wed	Apr 11	Lecture 35: Eureka-style Speculative Task Parallelism			worksheet 35	lec35-slides	**Homework 5**	**Homework 4 (all)**
	Fri	Apr 13	Lecture 36: Algorithms based on Parallel Prefix (Scan) operations			worksheet 36	lec36-slides		**Quiz for Unit 9**
14	Mon	Apr 16	Lecture 37: Algorithms based on Parallel Prefix (Scan) operations, contd.			worksheet 37	lec37-slides		
	Wed	Apr 18	Lecture 38: GPU Computing			worksheet 38	lec38-slides		
	Fri	Apr 20	Lecture 39: Course Review (Lectures 18-38)				lec39-slides		**Homework 5**
-									

Lab Schedule

Lab #	Date (2017)	Topic	Handouts	Code Examples
0		Infrastructure Setup	lab0-handout	-
1	Jan 11	Async-Finish Parallel Programming with abstract metrics	lab1-handout	lab_1.zip
2	Jan 18	Futures and HJ-Viz	lab2-handout, lab2-slides	lab_2.zip
3	Jan 25	Cutoff Strategy and Real World Performance	lab3-handout, lab3-slides	lab_3.zip
4	Feb 01	Java's ForkJoin Framework	lab4-handout, lab4-slides	lab_4.zip
5	Feb 08	Loop-level Parallelism	lab5-handout, lab5-slides	lab_5.zip
6	Feb 15	Phasers	lab6-handout	lab_6.zip
-	Feb 22	No lab this week — Exam 1	-	-

7	Mar 01	Isolated Statement and Atomic Variables	lab7-handout, lab7-slides	
8	Mar 08	Actors	lab8-handout	
-	Mar 15	No lab this week — Spring Break		
9	Mar 22	Java Threads, Java Locks	lab9-handout	
-	Mar 29	TBD		
10	Apr 05	Message Passing Interface (MPI)	lab10-handout	
11	Apr 12	Apache Spark	lab11-handout	
12	Apr 19	Eureka-style Speculative Task Parallelism	lab12-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), online quizzes (weighted 5% in all), and in-class worksheets (weighted 5% in all).

The purpose of the homeworks is to give you practice in solving problems that deepen your understanding of concepts introduced in class. Homeworks are due on the dates and times specified in the course schedule. No late submissions (other than those using slip days mentioned below) will be accepted.

The slip day policy for COMP 322 is similar to that of 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.). Slip days will be automatically tracked through the Autograder, more details are available later in this document and in the Autograder user guide. 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.

Labs must be checked off by a TA by the following Monday at 11:59pm.

Worksheets should be completed in class for full credit. For partial credit, a worksheet can be turned in before the start of the class following the one in which the worksheet for distributed, so that solutions to the worksheets can be discussed in the next class.

You will be expected to follow the Honor Code in all homeworks and exams. The following policies will apply to different work products in the course:

- In-class worksheets: You are free to discuss all aspects of in-class worksheets with your other classmates, the teaching assistants and the professor during the class. You can work in a group and write down the solution that you obtained as a group. If you work on the worksheet outside of class (e.g., due to an absence), then it must be entirely your individual effort, without discussion with any other students. If you use any material from external sources, you must provide proper attribution.
- Weekly lab assignments: You are free to discuss all aspects of lab assignments with your other classmates, the teaching assistants and the professor during the lab. However, all code and reports that you submit are expected to be the result of your individual effort. If you work on the lab outside of class (e.g., due to an absence), then it must be entirely your individual effort, without discussion with any other students. If you use any material from external sources, you must provide proper attribution (as shown here).
- Homeworks: All submitted homeworks are expected to be the result of your individual effort. You are free to discuss course material and approaches to 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.
- Quizzes: Each online quiz will be an open-notes individual test. The student may consult their course materials and notes when taking the quizzes, but may not consult any other external sources.
- Exams: Each exam will be a closed-book, closed-notes, and closed-computer individual written test, which must be completed within a specified time limit. No class notes or external materials may be consulted when taking the exams.

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 7 days.

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.