COMP322

Home	Office Hours	HJlib Info	edX site	Autograder Guide	Other Resources

COMP 322: Fundamentals of Parallel Programming (Spring 2021)

Instructor:	Mackale Joyner, DH 2063	**Head TAs:**	
Admin Assistant:	Annepha Hurlock, annepha@rice.edu, DH 3122, 713-348-5186	**Undergraduate TAs:**	Elian Ahmar, Paul Gao, Timothy Goh, Skylar Neuendorff, Kelly Park, Tucker Reinhardt, Mantej Singh, Minh Vu, Thanh Vu, Robert Walsh, Frederick Wang, Xincheng Wang, Yidi Wang
Piazza site:	https://piazza.com/rice/spring2021/comp322 (Piazza is the preferred medium for all course communications)	**Cross-listing:**	ELEC 323
Lecture location:	Fully Online	**Lecture times:**	MWF 1:30pm - 2:25pm
Lab locations:	Fully Online	**Lab times:**	Tu 1:30pm - 2:25pm, Th 4:50pm - 5:45pm

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 (2021)	Lecture	Assigned Reading	Assigned Videos (see Canvas site for video links)	In-class Worksheets	Slides	Work Assigned	Work Due		
1	Mon	Jan 25	Lecture 1: Task Creation and Termination (Async, Finish)	Module 1: Section 1.1	Topic 1.1 Lecture, Topic 1.1 Demonstration	worksheet1	lec1-slides				

	Wed	Jan 27	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	worksheet2	lec2-slides	**Homework 1**			
	Fri	Jan 29	Lecture 3: Abstract Performance Metrics, Multiprocessor Scheduling	Module 1: Section 1.4	Topic 1.4 Lecture, Topic 1.4 Demonstration	worksheet3	lec3-slides				
2	Mon	Feb 01	Lecture 4: Parallel Speedup and Amdahl's Law	Module 1: Section 1.5	Topic 1.5 Lecture, Topic 1.5 Demonstration	worksheet4	lec4-slides	**Quiz for Unit 1**			
	Wed	Feb 03	Lecture 5: Future Tasks, Functional Parallelism ("Back to the Future")	Module 1: Section 2.1	Topic 2.1 Lecture, Topic 2.1 Demonstration	worksheet5	lec5-slides				
	Fri	Feb 05	Lecture 6: Finish Accumulators	Module 1: Section 2.3	Topic 2.3 Lecture, Topic 2.3 Demonstration	worksheet6	lec6-slides		**Quiz for Unit 1**		
3	Mon	Feb 08	Lecture 7: Map Reduce	Module 1: Section 2.4	Topic 2.4 Lecture, Topic 2.4 Demonstration	worksheet7	lec7-slides				

	Wed	Feb 10	Lecture 8: Data Races, Functional & Structural Determinism	Module 1: Section 2.5, 2.6	Topic 2.5 Lecture, Topic 2.5 Demonstration, Topic 2.6 Lecture, Topic 2.6 Demonstration	worksheet8	lec8-slides	**Homework 2**	**Homework 1**		
	Fri	Feb 12	Lecture 9: Java's Fork/Join Library	Module 1: Sections 2.7, 2.8	Topic 2.7 Lecture, Topic 2.8 Lecture	worksheet9	lec9-slides	**Quiz for Unit 2**			
4	Mon	Feb 15	Lecture 10: Loop-Level Parallelism, Parallel Matrix Multiplication	Module 1: Sections 3.1, 3.2	Topic 3.1 Lecture , Topic 3.1 Demonstration , Topic 3.2 Lecture, Topic 3.2 Demonstration	worksheet10	lec10-slides				
	Wed	Feb 17	Spring "Sprinkle" Day (no class)								
	Fri	Feb 19	Lecture 11: Iteration Grouping (Chunking), Barrier Synchronization	Module 1: Sections 3.3, 3.4	Topic 3.3 Lecture , Topic 3.3 Demonstration, Topic 3.4 Lecture , Topic 3.4 Demonstration	worksheet11	lec11-slides		**Quiz for Unit 2**		

5	Mon	Feb 22	Lecture 12: Parallelism in Java Streams, Parallel Prefix Sums	Module 1: Section 3.7	Topic Topic 3.7 Java Streams, Topic 3.7 Java Streams Demonstration	worksheet12	lec12-slides				
	Wed	Feb 24	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	**Homework 3 (includes one intermediate checkpoint)** **Quiz for Unit 3**	**Homework 2**		
	Fri	Feb 26	Lecture 14: Data-Driven Tasks	Module 1: Sections 4.5	Topic 4.5 Lecture Topic 4.5 Demonstration	worksheet14	lec14-slides				
6	Mon	Mar 01	Spring "Sprinkle" Day (no class)								
	Wed	Mar 03	Lecture 15: Point-to-point Synchronization with Phasers	Module 1: Section 4.2, 4.3	Topic 4.2 Lecture, Topic 4.2 Demonstration, Topic 4.3 Lecture, Topic 4.3 Demonstration	worksheet15	lec15-slides				

	Fri	Mar 05	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	**Quiz for Unit 4**	**Quiz for Unit 3**		
7	Mon	Mar 08	Lecture 17: Midterm Review				lec17-slides				
	Wed	Mar 10	Lecture 18: Abstract vs. Real Performance			worksheet18	lec18-slides				
	Fri	Mar 12	Lecture 19: Critical Sections, Isolated construct (start of Module 2)	Module 2: Sections 5.1, 5.2, 5.6,	Topic 5.1 Lecture, Topic 5.1 Demonstration, Topic 5.2 Lecture, Topic 5.2 Demonstration, Topic 5.6 Lecture, Topic 5.6 Demonstration	worksheet19	lec19-slides		**Homework 3, Checkpoint-1**		

8	Mon	Mar 15	Lecture 20: Parallel Spanning Tree algorithm, Atomic variables	Module 2: Sections 5.3, 5.4, 5.5	Topic 5.3 Demonstration, Topic 5.4 Lecture, Topic 5.4 Demonstration, Topic 5.5 Lecture, Topic 5.5 Demonstration	worksheet20	lec20-slides				
	Wed	Mar 17	Lecture 21: Actors	Module 2: 6.1, 6.2	Topic 6.1 Lecture , Topic 6.1 Demonstration , Topic 6.2 Lecture, Topic 6.2 Demonstration	worksheet21	lec21-slides				
	Fri	Mar 19	Lecture 22: Actors (contd)	Module 2: 6.3, 6.4, 6.5	Topic 6.3 Lecture, Topic 6.3 Demonstration, Topic 6.4 Lecture , Topic 6.4 Demonstration, Topic 6.5 Lecture, Topic 6.5 Demonstration	worksheet22	lec22-slides		**Quiz for Unit 4**		
9	Mon	Mar 22	Lecture 23: Actors (contd)	Module 2: 6.6	Topic 6.6 Lecture, Topic 6.6 Demonstration		lec23-slides	**Quiz for Unit 5**			

	Wed	Mar 24	Lecture 24: Java Threads, Java synchronized statement	Module 2: 7.1, 7.2	Topic 7.1 Lecture, Topic 7.2 Lecture		lec24-slides				
	Fri	Mar 26	Spring "Sprinkle" Day (no class)								
10	Mon	Mar 29	Lecture 25: Java Threads, Java synchronized statement (contd), wait/notify	Module 2: 7.1, 7.2	Topic 7.1 Lecture, Topic 7.2 Lecture		lec25-slides				
	Wed	Mar 31	Lecture 26: Java Threads (exercise)				lec26-handout		**Homework 3 (all)**		
	Fri	Apr 02	Lecture 27: Java Locks	Module 2: 7.3	Topic 7.3 Lecture		lec27-slides	**Quiz for Unit 6**	**Quiz for Unit 5**		
11	Mon	Apr 05	Lecture 28: Linearizability of Concurrent Objects	Module 2: 7.4	Topic 7.4 Lecture		lec28-slides	**Homework 4 (includes one intermediate checkpoint)**			
	Wed	Apr 07	Lecture 29: Java Locks (exercise)				lec29-handout				

	Fri	Apr 09	Lecture 30: Safety and Liveness Properties, Java Synchronizers, Dining Philosophers Problem	Module 2: 7.5, 7.6	Topic 7.5 Lecture, Topic 7.6 Lecture		lec30-slides	**Quiz for Unit 7**	**Quiz for Unit 6**		
12	Mon	Apr 12	Lecture 31: Message Passing Interface (MPI), (start of Module 3)		Topic 8.1 Lecture, Topic 8.2 Lecture, Topic 8.3 Lecture		lec31-slides				
	Wed	Apr 14	Lecture 32: Message Passing Interface (MPI, contd)		Topic 8.4 Lecture		lec32-slides		**Homework 4 Checkpoint-1**		
	Fri	Apr 16	Lecture 33: Message Passing Interface (MPI, contd)		Topic 8.5 Lecture, Topic 8 Demonstration Video		lec33-slides				
13	Mon	Apr 19	Lecture 34: Task Affinity with Places				lec34-slides	**Quiz for Unit 8**	**Quiz for Unit 7**		
	Wed	Apr 21	Lecture 35: Eureka-style Speculative Task Parallelism				lec35-slides				

	Fri	Apr 24	Lecture 36: Algorithms based on Parallel Prefix (Scan) operations				lec36-slides				
14	Mon	Apr 26	TBD								
	Wed	Apr 28	Lecture 38: Course Review (Lectures 19-34)				lec38-slides		**Homework 4 (all)**		
	Fri	Apr 30	TBD						**Quiz for Unit 8**		

Lab Schedule

Lab #	Date (2021)	Topic	Handouts	Examples
0		Infrastructure Setup	lab0-handout	
1	Jan 26	Async-Finish Parallel Programming with abstract metrics	lab1-handout	
-	Feb 02	No lab this week		
2	Feb 09	Futures	lab2-handout	
3	Feb 16	Cutoff Strategy and Real World Performance	lab3-handout	
4	Feb 23	DDFs	lab4-handout	
-	Mar 02	No lab this week		
5	Mar 09	Loop-level Parallelism	lab5-handout	lab5-intro
-	Mar 16	No lab this week (Spring "Sprinkle" Day)		
-		Isolated Statement and Atomic Variables		
-		Actors		
-		Java Threads, Java Locks		
-		Message Passing Interface (MPI)		
-		Apache Spark		

-		Eureka-style Speculative Task Parallelism		
-		Java's ForkJoin Framework		

Grading, Honor Code Policy, Processes and Procedures

Grading will be based on your performance on four homework assignments (weighted 40% in all), two exams (weighted 40% in all), lab exercises (weighted 10% in all), online quizzes (weighted 5% in all), and in-class worksheets (weighted 5% in all).

The purpose of the homework is to give you practice in solving problems that deepen your understanding of concepts introduced in class. Homework is 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 tracked using the README.md file. 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 submitted by the following Monday at 11:59pm. Labs must be checked off by a TA.

Worksheets should be completed in Canvas before the start of the following class (for full credit) so that solutions to the worksheets can be discussed in the next class.

You will be expected to follow the Honor Code in all homework 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).
- Homework: All submitted homework is 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 open-book, open-notes, and open-computer individual test, which must be completed within a specified time limit. No external materials may be consulted when taking the exams.

For grade disputes, please send an email to the course instructors within 7 days of receiving your grade. The email subject should include COMP 322 and the assignment. Please provide enough information in the email so that the instructor does not need to perform a checkout of your code.

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.