COMP322-Spring-2014

COMP 322: Fundamentals of Parallel Programming (Spring 2014)

Instructor:	Prof. Vivek Sarkar, DH 3080	**Graduate TA:**	Kumud Bhandari
	Please send all emails to comp322-staff at rice dot edu	**Graduate TA:**	Rishi Surendran
Assistant:	Penny Anderson, anderson@rice.edu, DH 3080	**Graduate TA:**	Yunming Zhang
		Undergrad TA:	Wenxuan Cai
		Undergrad TA:	Kyle Kurihara
Cross-listing:	ELEC 323	**Undergrad TA:**	Max Payton
		Course consultants:	Vincent Cavé, Shams Imam, Maggie Tang, Bing Xue
Lectures:	Herzstein Hall 212	**Lecture times:**	MWF 1:00 - 1:50pm
Labs:	Symonds II	**Lab times:**	Monday, 4:00 - 5:30pm (Section A01, Staff: Yunming, Kumud, Wenxuan, Maggie)
			Wednesday, 4:30 - 6:00pm (Section A02, Staff: Rishi, Kyle, Max, Bing)

Course Objectives

The goal of COMP 322 is to introduce you to the fundamentals of parallel programming and parallel algorithms, using 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 model that you may encounter in the future, and also prepare you for studying advanced topics related to parallelism and concurrency in more advanced courses such as COMP 422.

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.

Course Overview

COMP 322 provides the student with a comprehensive introduction to the building blocks of parallel software, which includes the following concepts:

- Primitive constructs for task creation & termination, synchronization, task and data distribution
- Abstract models: parallel computations, computation graphs, Flynn's taxonomy (instruction vs. data parallelism), PRAM model
- Parallel algorithms for data structures that include arrays, lists, strings, trees, graphs, and key-value pairs
- Common parallel programming patterns including task parallelism, pipeline parallelism, data parallelism, divide-and-conquer parallelism, map-reduce, concurrent event processing including graphical user interfaces.

These concepts will be introduced in three modules:

1. *Deterministic Shared-Memory 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 arrays.
2. *Nondeterministic Shared-Memory Parallelism and 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. *Distributed-Memory Parallelism and Locality*: memory hierarchies, cache affinity, data movement, message-passing (MPI), communication overheads (bandwidth, latency), MapReduce, accelerators, GPGPUs, CUDA, OpenCL, energy efficiency, resilience.

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 221 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:

- Module 1 handout (*Deterministic Shared-Memory Parallelism*)
- Module 2 handout (*Nondeterministic Shared-Memory Parallelism and Concurrency*)
- Module 3 handout (*Distributed-Memory Parallelism 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:

- 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 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

lec35-slides

Week	Day	Date (2014)	Topic	Reading	Videos	In-class Worksheets	Slides	Code Examples	Work Assigned	Work Due
1	Mon	Jan 13	Lecture 1: The What and Why of Parallel Programming, Task Creation and Termination (Async, Finish)	Module 1: Sections 0.1, 0.2, 1.1	Topic 1.1 Lecture, Topic 1.1 Demonstration	worksheet1	lec1-slides	Demo File: ReciprocalArraySum.java	Topic 1.1 Lecture Quiz, Topic 1.1 Demo Quiz	
	Wed	Jan 15	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	Demo File: Search.java	Topic 1.2 Lecture Quiz , Topic 1.2 Demo Quiz , Topic 1.3 Lecture Quiz , Topic 1.3 Demo Quiz	
	Fri	Jan 17	Lecture 3: , Abstract Performance Metrics, Multiprocessor Scheduling	Module 1: Section 1.4	Topic 1.4 Lecture, Topic 1.4 Demonstration	worksheet3	lec3-slides	Worksheet File: Search.java Homework 1 Files: QuicksortUtil.java , QuicksortSeq.java , QuicksortPar.java	Homework 1, Topic 1.4 Lecture Quiz , Topic 1.4 Demo Quiz, Topic 1.6 Lecture Quiz , Topic 1.6 Demo Quiz	
2	Mon	Jan 20	No lecture, School Holiday (Martin Luther King, Jr. Day)							
	Wed	Jan 22	Lecture 4: Parallel Speedup and Amdahl's Law	Module 1: Section 1.5	Topic 1.5 Lecture, Topic 1.5 Demonstration	worksheet4	lec4-slides	Demo File: VectorAdd.java	Topic 1.5 Lecture Quiz , Topic 1.5 Demo Quiz	
	Fri	Jan 24	No lecture (inclement weather)							All 12 lecture & demo quizzes in Unit 1 are due by 5pm CST today
3	Mon	Jan 27	Lecture 5: Future Tasks, Functional Parallelism	Module 1: Section 2.1	Topic 2.1 Lecture , Topic 2.1 Demonstration	worksheet5	lec5-slides	Demo File(s): ReciprocalArraySumFutures.java, BinaryTreesSeq.java, BinaryTrees.java		
	Wed	Jan 29	Lecture 6: Finish Accumulators	Module 1: Section 2.3	Topic 2.3 Lecture , Topic 2.3 Demonstration	worksheet6	lec6-slides	Demo File: Nqueens.java Worksheet5.java, nqueens.java		

	Fri	Jan 31	Lecture 7: 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		lec7-slides	Demo File: ReciprocalArraySum.java		Homework 1
4	Mon	Feb 03	Lecture 8: Map Reduce	Module 1: Section 2.4	Topic 2.4 Lecture , Topic 2.4 Demonstration	worksheet8	lec8-slides	Demo File(s): WordCount.java, words.txt Worksheet Files: WordCount.java , words.txt Homework 2 Files: GeneralizedReduce.java, GeneralizedReduceApp.java, SumReduction.java, TestSumReduction.java	Homework 2	
	Wed	Feb 05	Lecture 9: Memoization	Module 1: Section 2.2	Topic 2.2 Lecture , Topic 2.2 Demonstration	worksheet9	lec9-slides	Demo File: PascalsTriangleWithFuture.java Worksheet File: PascalsTriangleMemoized.java Worksheet Solution: PascalsTriangleMemoizedSolution.java		
	Fri	Feb 07	Lecture 10: Abstract vs. Real Performance			worksheet10	lec10-slides			
5	Mon	Feb 10	Lecture 11: Loop-Level Parallelism, Parallel Matrix Multiplication		Topic 3.1 Lecture, Topic 3.1 Demonstration, Topic 3.2 Lecture , Topic 3.2 Demonstration	worksheet11	lec11-slides	Demo File: ForallWithIterable.java, VectorAddForall.java, MatrixMultiplicationMetrics.java		
	Wed	Feb 12	Lecture 12: Iteration Grouping (Chunking), Barrier Synchronization		Topic 3.3 Lecture , Topic 3.3 Demonstration , Topic 3.4 Lecture , Topic 3.4 Demonstration	worksheet12	lec12-slides	Demo File: MatrixMultiplicationPerformance.java, BarrierInForall.java		
	Fri	Feb 14	Lecture 13: Iterative Averaging Revisited		Topic 3.5 Lecture , Topic 3.5 Demonstration , Topic 3.6 Lecture , Topic 3.6 Demonstration	worksheet13	lec13-slides	Demo File: OneDimAveragingGrouped.java, OneDimAveragingBarrier.java Worksheet File: OneDimAveragingBarrier.java		
6	Mon	Feb 17	Lecture 14: Data-Driven Tasks and Data-Driven Futures		Topic 4.5 Lecture , Topic 4.5 Demonstration	worksheet14	lec14-slides	Demo File: DataDrivenFutures4.java		Homework 2
	Wed	Feb 19	Lecture 15: Review of Module-1 HJ-lib API's			worksheet15	lec15-slides	Homework 3 Files: SeqScoring.java	Homework 3	
	Fri	Feb 21	Lecture 16: Point-to-point Synchronization with Phasers		Topic 4.2 Lecture , Topic 4.2 Demonstration	worksheet16	lec16-slides	Demo File: Phaser3Asyncs.java		
7	Mon	Feb 24	Lecture 17: Phasers (contd), Signal Statement, Fuzzy Barriers		Topic 4.1 Lecture , Topic 4.1 Demonstration	worksheet17	lec17-slides	Demo File: PhaserSignal.java		
	Wed	Feb 26	Lecture 18: Midterm Summary, Take-home Exam 1 distributed				lec18-slides		Exam 1	
	F	Feb 28	No Lecture (Exam 1 due by 4pm today)							Exam 1
-	M-F	Feb 28-Mar 09	Spring Break							
8	Mon	Mar 10	Lecture 19: Critical sections, Isolated construct, Parallel Spanning Tree algorithm		Topic 5.1 Lecture, Topic 5.1 Demonstration, Topic 5.2 Lecture, Topic 5.2 Demonstration, Topic 5.3 Lecture, Topic 5.3 Demonstration	worksheet19	lec19-slides			
	Wed	Mar 12	Lecture 20: Speculative parallelization of isolated constructs (Guest lecture by Prof. Swarat Chaudhuri)			worksheet20	lec20-slides			Homework 3

	Fri	Mar 14	Lecture 21: Read-Write Isolation, Atomic variables		Topic 5.4 Lecture , Topic 5.4 Demonstration , Topic 5.5 Lecture, Topic 5.5 Demonstration, Topic 5.6 Lecture, Topic 5.6 Demonstration	worksheet21	lec21-slides			
9	Mon	Mar 17	Lecture 22: 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	worksheet22	lec22-slides	Homework 4 Files: hw4_files.zip	Homework 4	
	Wed	Mar 19	Lecture 23: 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	worksheet23	lec23-slides			
	Fri	Mar 21	Lecture 24: Monitors, Java Concurrent Collections, Linearizability of Concurrent Objects		Topic 7.4 Lecture	worksheet24	lec24-slides			
10	Mon	Mar 24	Lecture 25: Linearizability (contd), Intro to Java Threads		Topic 7.1 Lecture	worksheet25	lec25-slides			
	Wed	Mar 26	Lecture 26: Java Threads (contd), Java synchronized statement		Topic 7.2 Lecture	worksheet26	lec26-slides			
	Fri	Mar 28	Lecture 27: Java synchronized statement (contd), advanced locking		Topic 7.3 Lecture	worksheet27	lec27-slides			
11	Mon	Mar 31	Lecture 28: Safety and Liveness Properties		Topic 7.5 Lecture	worksheet28	lec28-slides			
	Wed	Apr 02	Lecture 29: Dining Philosophers Problem		Topic 7.6 Lecture	worksheet29	lec29-slides			Homework 4 (due by 11:55pm on April 2nd)
-	Fri	Apr 04	Midterm Recess							
12	Mon	Apr 07	Lecture 30: Message Passing Interface (MPI)			worksheet30	lec30-slides	Homework 5 files: hw5_files.zip	Homework 5	
	Wed	Apr 09	Lecture 31: Partitioned Global Address Space (PGAS) languages (Guest lecture by Prof. John Mellor-Crummey)			worksheet31	lec31-slides			
	Fri	Apr 11	Lecture 32: Message Passing Interface (MPI, contd)			worksheet32	lec32-slides			
13	Mon	Apr 14	Lecture 33: Task Affinity with Places			worksheet33	lec33-slides			
	Wed	Apr 16	Lecture 34: GPU Computing			worksheet34	lec34-slides			
	Fri	Apr 18	Lecture 35: Memory Consistency Models			worksheet35	lec35-slides	Homework 6 (written only)		
14	Mon	Apr 21	Lecture 36: Comparison of Parallel Programming Models			worksheet36	lec36-slides			Homework 5 (due by 11:55pm on Monday, April 21st)
	Wed	Apr 23	NO CLASS (time allocated to work on homeworks)							
	Fri	Apr 25	Lecture 37: Course Review (lectures 19-35), Take-home Exam 2 distributed, Last day of classes				lec37-slides		Exam 2	Homework 6 (due by 11:55pm on April 25th, penalty-free extension till May 2nd)
-	Fri	May 02	Exam 2 due by 4pm today							Exam 2

Lab Schedule

Lab #	Date (2014)	Topic	Handouts	Code Examples
1	Jan 13, 15	Infrastructure setup, Async-Finish Parallel Programming	lab1-handout	HelloWorldError.java, ReciprocalArraySum.java
-	Jan 20, 22	No lab this week — Jan 20 is Martin Luther King, Jr. Day		
2	Jan 27, 29	Abstract performance metrics with async & finish	lab2-handout	ArraySum1.java , ArraySumUtil.java Search2.java , ArraySumLoop.java , ArraySumRecursive.java
3	Feb 03, 05	Futures	lab3-handout	ArraySum2.java, ArraySum4.java, BinaryTrees.java
4	Feb 10, 12	Real Performance from Finish Accumulators and Loop-Level Parallelism	lab4-handout	Nqueens.java, OneDimAveraging.java, Linux/Sugar Tutorial
5	Feb 17, 19	Futures vs. Data-Driven Futures	lab5-handout	MatrixEval.java, test.txt
6	Feb 24, 26	Barriers and Phasers	lab6-handout	OneDimAveraging.java
-	Mar 03, 05	No lab this week — Spring Break		
7	Mar 10, 12	Isolated Statement and Atomic Variables	lab7-handout	spanning_tree_seq.java
8	Mar 17, 19	Actors	lab8-handout	PiSerial1.java PiActor1.java PiSerial2.java PiActor2.java PiUtil.java Sieve .java SieveSerial.java
9	Mar 24, 26	Java Threads	lab9-handout	nqueens_hj.java spanning_tree_atomic_hj.java
10	Mar 31, Apr 02	Java Locks	lab10-handout	lab10.zip
11	Apr 07, 09	Message Passing Interface (MPI)	lab11-handout	lab11.zip
12	Apr 14, 16	Map Reduce	lab12-handout	
-	Apr 21, 23	No lab this week — Last Week of Classes		

Grading, Honor Code Policy, Processes and Procedures

Grading will be based on your performance on six homeworks (weighted 40% in all), two exams (weighted 20% each), weekly lab exercises (weighted 10% in all), and class participation including worksheets, in-class Q&A, Piazza participation, etc (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 CLEAR turn-in system. Homework is worth full credit when turned in on time. A 10% penalty per day will be levied on late homeworks, up to a maximum of 6 days. No submissions will be accepted more than 6 days after the due date.

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.