COMP322-Spring-2013

COMP 322: Fundamentals of Parallel Programming (Spring 2013)

Instructor:	Prof. Vivek Sarkar, DH 3131	**Graduate TA:**	Kumud Bhandari
	Please send all emails to comp322-staff at rice dot edu	**Graduate TA:**	Deepak Majeti
Assistant:	Sherry Nassar, sherry.nassar@rice.edu, DH 3137	**Graduate TA:**	Sriraj Paul
		Graduate TA:	Rishi Surendran
		Undergrad TA:	Annirudh Prasad
Cross-listing:	ELEC 323	**Undergrad TA:**	Yunming Zhang
		HJ consultants:	Vincent Cavé, Max Grossman, Shams Imam
Lectures:	Herzstein Hall 212	**Lecture times:**	MWF 1:00 - 1:50pm
Labs:	Ryon 102	**Lab times:**	Tuesday, 4:00 - 5:15pm (Section 3, TAs: Kumud Bhandari, Yunming Zhang)
			Wednesday, 3:30 - 4:50pm (Section 2, TAs: Deepak Majeti, Sriraj Paul)
			Thursday, 4:00 - 5:15pm (Section 1, TAs: Annirudh Prasad, Rishi Surendran)

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 get a strong grasp of parallel programming foundations, the classes and homeworks will place equal emphasis on advancing both theoretical and practical knowledge. The programming component of the course work will initially use a simple parallel extension to the Java language called Habanero-Java (HJ), developed in the Habanero Multicore Software Research project at Rice University. Later in the course, we will introduce you to some real-world parallel programming models including Java Concurrency, .Net Task Parallel Library, MapReduce, CUDA and MPI. The use of Java will be confined to a subset of the Java language that should also be accessible to C programmers --- advanced Java features (e.g., wildcards in generics) will not be used. 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; any parallel programming primitives should be easily recognizable based on the primitives 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 four 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.
2. *Nondeterministic Shared-Memory Parallelism and Concurrency:* critical sections, atomicity, isolation, high level data races, nondeterminism, linearizability, liveness/progress guarantees, actors, request-response parallelism
3. *Distributed-Memory Parallelism and Locality*: memory hierarchies, cache affinity, false sharing, message-passing (MPI), communication overheads (bandwidth, latency), MapReduce, systolic arrays, accelerators, GPGPUs.
4. *Current Practice — today's Parallel Programming Models and Challenges:* Java Concurrency, locks, condition variables, semaphores, memory consistency models, comparison of parallel programming models (.Net Task Parallel Library, OpenMP, CUDA, OpenCL); energy efficiency, data movement, resilience.

Prerequisite

The prerequisite course requirement is COMP 215 or equivalent. This course 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*)
- Module 4 handout (*Current Practice — today's Parallel Programming Models and Challenges*)

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

- 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)

Quiz Schedule

- Lab quizzes are usually published on Tuesday each week there is a lab, and are due by that Friday night.
- Lecture quizzes are usually published on Saturday each week and are due by the following Tuesday night.
    - Exception: combined lecture quiz for Week 5 and Week 6 will be assigned on Thursday, Feb 14, and due by Sunday, Feb 17, night

Lecture Schedule

Week	Day	Date (2013)	Topic	Reading	Slides	Audio (Panopto)	Code Examples	Homework Assigned	Homework Due
1	Mon	Jan 7	Lecture 1: The What and Why of Parallel Programming	Module 1: Sections1.1, 1.2, 2.1, 2.2	lec1-slides	lec1-audio	ArraySum0.hj		
	Wed	Jan 9	Lecture 2: Async-Finish Parallel Programming, Data & Control Flow with Async Tasks, Computation Graphs	Module 1: Sections 1.3, 3.1, 3.2	lec2-slides	lec2-audio		HW1, quicksort.hj	
	Fri	Jan 11	Lecture 3: Computation Graphs (contd), Parallel Speedup, Strong Scaling, Abstract Performance Metrics	Module 1: Sections 3.1, 3.2, 3.3	lec3-slides		ArraySum1.hj		
2	Mon	Jan 14	Lecture 4: Abstract Performance Metrics (contd), Parallel Efficiency, Amdahl's Law, Weak Scaling	Module 1: Sections 3.3, 3.4	lec4-slides	lec4-audio	Search2.hj		
	Wed	Jan 16	Lecture 5: Data Races, Determinism, Memory Models	Module 1: Chapter 4	lec5-slides				
	Fri	Jan 18	Lecture 6: Data races (contd), Futures --- Tasks with Return Values	Module 1: Chapter 4, Section 5.1, 5.2	lec6-slides	lec6-audio			
3	Mon	Jan 21	No lecture, School Holiday (Martin Luther King, Jr. Day)						
	Wed	Jan 23	No lecture, Reading Assignment on Futures: Chapter 5 of Module 1 handout	Module 1: Chapter 5				HW2, GeneralizedReduce.hj, GeneralizedReduceApp.hj, SumReduction.hj, TestSumReduction.hj	HW1 (due by 5pm on Jan 23rd)
	Fri	Jan 25	Lecture 7: Futures (contd), Parallel Design Patterns, Finish Accumulators	Module 1: Chapter 5, Chapter 6	lec7-slides				

4	Mon	Jan 28	Lecture 8: Parallel N-Queens, Parallel Prefix Sum (Array Reductions with Associative Operators)	Module 1: Chapter 7	lec8-slides	lec8-audio			
	Wed	Jan 30	Lecture 9: Abstract vs. Real Performance	Module 1: Chapter 9	lec9-slides	lec9-audio			
	Fri	Feb 1	Lecture 10: Abstract vs. Real Performance (contd), seq clause	Module 1: Chapter 9	lec10-slides	lec10-audio			
5	Mon	Feb 04	Lecture 11: Forasync Loops, Forasync Chunking	Module 1: Sections 8.1, 9.4	lec11-slides				
	Wed	Feb 06	Lecture 12: Forall Loops, Barrier Synchronization	Module 1: Sections 10.1, 10.2, 10.4	lec12-slides				HW2 (due by 5pm on Feb 7th)
	Fri	Feb 08	Lecture 13: Forall and Barriers, Dataflow Computing, Data-Driven Tasks	Module 1: Chapters 10, 11	lec13-slides	lec13-audio		HW3, SeqScoring.hj, X.txt, Y.txt, BigSeq.zip, UsefulParScoring.hj, SparseParScoring.hj	
6	Mon	Feb 11	Lecture 14: Recap of HJ constructs, Point-to-point Synchronization, Pipeline Parallelism, Introduction to Phasers	Module 1: Sections 12.1, 12.2	lec14-slides	lec14-audio			
	Wed	Feb 13	Lecture 15: Point-to-point Synchronization with Phasers	Module 1: Section 12.3	lec15-slides				
	Fri	Feb 15	Lecture 16: Phaser Accumulators, Bounded Phasers, Summary of Barriers and Phasers	Module 1: Chapter 12	lec16-slides	lec16-audio			
7	Mon	Feb 18	Lecture 17: Midterm Summary		lec17-slides	lec17-audio			
	Wed	Feb 20	Lecture 18: Midterm Summary (contd), Take-home Exam 1 distributed		lec18-slides				
	F	Feb 22	No Lecture (Exam 1 due by 5pm today)						HW3 (due by 11:55pm on Feb 24th)
-	M-F	Feb 25-Mar 01	Spring Break						
8	Mon	Mar 04	Lecture 19: Critical sections, Isolated statement, Atomic variables	Module 2: Chapters 1, 2, 4, 6	lec19-slides	lec19-audio			
	Wed	Mar 06	Lecture 20: Parallel Spanning Tree algorithm, Monitors, Java Concurrent Collections	Module 2: Chapters 3, 7	lec20-slides	lec20-audio		HW4, hw_4.zip	
	Fri	Mar 08	Lecture 21: Actors	Module 2: Chapter 8	lec21-slides	lec21-audio			
9	Mon	Mar 11	Lecture 22: Actors (contd), Linearizability of Concurrent Objects	Module 2: Chapters 8, 9	lec22-slides	lec22-audio			
	Wed	Mar 13	Lecture 23: Linearizability of Concurrent Objects (contd)	Module 2: Chapters 9, 10	lec23-slides	lec23-audio			
	Fri	Mar 15	Lecture 24: Safety and Liveness Properties, Intro to Java Threads	Module 2: Chapters 11, 12	lec24-slides	lec24-audio			
10	Mon	Mar 18	Lecture 25: Java Threads (contd), Java synchronized statement	Module 2: Chapters 12, 13, 14	lec25-slides				
	Wed	Mar 20	Lecture 26: Java synchronized statement (contd), advanced locking	Module 2: Chapter 14	lec26-slides				
	Fri	Mar 22	Lecture 27: Speculative parallelization of isolated blocks (Guest lecture by Prof. Swarat Chaudhuri)		lec27-slides				HW4 (due by 11:55pm on March 22nd)
11	Mon	Mar 25	Lecture 28: Java Executors and Synchronizers		lec28-slides	lec28-audio			
	Wed	Mar 27	Lecture 29: Dining Philosophers Problem		lec29-slides				
-	Fri	Mar 29	Midterm Recess						
12	Mon	Apr 01	Lecture 30: Task Affinity with Places		lec30-slides	lec30-audio		HW5, hw_5.zip	
	Wed	Apr 03	Lecture 31: More on Actors: Places, Dining Philosophers (Guest lecture by Shams Imam)		lec31-slides		DiningPhilosopher.hj		
	Fri	Apr 05	Lecture 32: Message Passing Interface (MPI)		lec32-slides	lec32-audio			
13	Mon	Apr 08	Lecture 33: Message Passing Interface (MPI, contd)		lec33-slides				
	Wed	Apr 10	Lecture 34: Message Passing Interface (MPI, contd)		lec34-slides				
	Fri	Apr 12	Lecture 35: Cloud Computing, Map Reduce		lec35-slides				HW5 (due by 11:55pm on Sunday, April 14th)

14	Mon	Apr 15	Lecture 36: Partitioned Global Address Space (PGAS) languages (Guest lecture by Prof. John Mellor-Crummey)		lec36-slides			HW6	
	Wed	Apr 17	Lecture 37: Comparison of Parallel Programming Models		lec37-slides				
	Fri	Apr 19	Lecture 38: Course Review, Take-home Exam 2 distributed		lec38-slides				HW6 (due by 11:55pm on April 19th, penalty-free extension till April 26th)
-	Fri	Apr 26	No lectures this week — Exam 2 due by 4pm today						

Lab Schedule

Lab #	Date (2013)	Topic	Handouts	Code Examples
1	Jan 08, 09, 10	Infrastructure setup, Async-Finish Parallel Programming	lab1-handout	HelloWorldError.hj, ReciprocalArraySum.hj
2	Jan 15, 16, 17	Abstract performance metrics with async & finish	lab2-handout	ArraySum1.hj, Search2.hj, ArraySum3.hj
3	Jan 22, 23, 24	Data race detection and repair	lab3-handout	RacyArraySum1.hj, RacyFib.hj, RacyParSearch.hj, RacyFannkuch.hj
4	Jan 29, 30, 31	Futures, Finish Accumulators	lab4-handout	ArraySum2.hj, ArraySum4.hj, binarytrees.hj
5	Feb 05, 06, 07	Real performance, work-sharing and work-stealing runtimes	lab5-handout, linux-tutorial-handout	nqueens.hj, OneDimAveraging.hj
6	Feb 12, 13, 14	Barriers, Data-Driven Futures	lab6-handout	Data-Driven Future Examples: TestAsyncDDF0.hj, TestAsyncDDF2.hj
-	Feb 19, 20, 21	No lab (HW3 due, Exam 1 assigned)		
7	Mar 05, 06, 07	Isolated Statement and Atomic Variables	lab7-handout	spanning_tree_seq.hj
8	Mar 12, 13, 14	Actors	lab8-handout	PiSerial1.hj, PiSerial2.hj, PiUtil.hj, PiActor1.hj, PiActor2.hj, SieveSerial.hj, Sieve.hj, other-actor-examples
9	Mar 19, 20, 21	Java Threads	lab9-handout	nqueens.hj , spanning_tree_atomic.hj
10	Mar 26, 27, 28	Java Locks	lab10-handout	lab10.zip
-	Apr 02, 03, 04	No new lab (extra time to complete Lab 10 due to midterm recess)		
11	Apr 09, 10, 11	Message Passing Interface (MPI)	lab11-handout	lab11.zip
12	Apr 16, 17, 18	Map Reduce	lab12-handout	

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 lecture & lab quizzes (weighted 10% in all), and class participation (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, quizzes 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 and all quizzes are pledged under the Honor Code. They test your individual understanding and knowledge of the material. Collaboration on quizzes and exams is strictly forbidden. Quizzes are open-book and exams are closed-book. Finally, it is also your responsibility to protect your homeworks, quizzes 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.