General Information

Instructor: Matthew Fluet
E-mail: mtf@cs.rit.edu
Office hours: W 3:00pm – 5:00pm; GOL(70)-3555
                 R 9:00am – 11:00am; GOL(70)-3555
                 F 1:00pm – 3:00pm; GOL(70)-3555
                 or by appointment

Lectures: Section 01    TR 4:00pm – 5:50pm; GOL(70)-2690

Website: www.cs.rit.edu/~mtf/teaching/20102/pcl
           mycourses.rit.edu

Course Description

Parallel Computing is the study of the hardware and software issues in parallel computing. Topics include an introduction to the basic concepts, parallel architectures and network topologies, parallel algorithms, parallel metrics, parallel languages, granularity, applications, parallel programming design and debugging. Students will become familiar with various types of parallel architectures and programming environments. Programming projects will be required.

Prerequisites

- 4003-440 (Operating Systems I) (for 4003-531)
- 4003-713 (Operating Systems) (for 4005-735)
- or permission of instructor

Course Goals

The objectives of this course are to provide students with the fundamentals of parallel computing and to introduce them to the myriad of ways in which parallel computing has been realized. Programming assignments will provide experience in utilizing these fundamental paradigms.

Text Books

Required:
Title: Building Parallel Programs: SMPs, Clusters, and Java
Author: Alan Kaminsky
Publisher: Cengage Course Technology
ISBN: 978-1423901983
Home page: http://www.cs.rit.edu/~ark/bpp/
Topics

1. Introduction
   (a) Why parallel computing
   (b) Hardware Issues
   (c) Software Issues
   (d) Performance Issues
2. Parallel Languages
   (a) Issues
   (b) Parallel Java Library
   (c) Parallaxis
   (d) Linda
   (e) MPI
   (f) Parallel Fortran Issues
   (g) OpenMP
3. Parallel Metrics
   (a) Speedup
   (b) Efficiency
   (c) Cost
   (d) Amdahl’s Law
4. Network Topologies
   (a) Topologies
   (b) Why the hypercube has been popular
   (c) Diameter/Degree
   (d) Grey codes
   (e) Dynamically configurable networks
   (f) Related to issues with shared memory
5. Parallel Algorithms/Application
   (a) Designing Parallel Algorithms
      i. Summation/Max/Min
   (b) From SIMD to MIMD
      i. Odd-Even Transposition Sort
      ii. Parallel Prefix
   (c) 1D Wave Equation
   (d) Matrix Multiplication
   (e) Parallelization of iterative techniques
   (f) Parallel Sorting
   (g) Parallel Graphics
   (h) FFTs
6. Heterogeneous, cluster and grid computing
7. Future Parallel Computers?
   (a) Quantum Computing
   (b) DNA Computing
Outcomes

- The student will explain the hardware architectures used for building parallel computers, including shared memory multiprocessors (SMPs) and clusters. (Evaluation: Exams)
- The student will explain the software architectures and middleware standards used for programming parallel computers, including SMPs and clusters. (Evaluation: Exams)
- The student will calculate metrics of a parallel program such as running time, speedup, and sizeup, given the elemental computation, synchronization, and communication times. (Evaluation: Exams, homeworks)
- The student will calculate metrics of a parallel program such as running time, speedup, and sizeup from experimental data. (Evaluation: Exams, homeworks, project)
- The student will explain common parallel programming patterns and choose the patterns best suited for solving particular problems. (Evaluation: Exams, homeworks, project)
- The student will develop programs for an SMP parallel computer using the appropriate parallel programming patterns and middleware. (Evaluation: Homeworks, project)
- The student will develop programs for a cluster parallel computer using the appropriate parallel programming patterns and middleware. (Evaluation: Homeworks, project)

Grades, Exams, and Assignments

Grades will be assigned based on the following grading scheme:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance &amp; Participation</td>
<td>10.0%</td>
</tr>
<tr>
<td>Homeworks/Programming Assignments</td>
<td>25.0%</td>
</tr>
<tr>
<td>Term Team Project</td>
<td>25.0%</td>
</tr>
<tr>
<td>Mid-term Exam</td>
<td>20.0%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20.0%</td>
</tr>
</tbody>
</table>

Attendance & Participation

Students are required to attend and expected to participate in class. Participation means being an engaged student: asking and answering questions, not simply attending class. The use of cell phones and audio players is prohibited during class. If you must take a phone call, please leave the classroom immediately and do not return until you have ended the phone call. The use of a laptop (or notebook or netbook) computer is permitted during class only for the purpose of taking notes. Persistent use of a laptop for other activities will result in 0 credit for your Attendance & Participation grade. Assigned readings should be completed before the lecture section. You are responsible for the material in assigned readings, whether covered during lecture or not.
Term Team Project

Students will undertake a term project in teams of 2 or 3 students. The term project will consist of proposing a topic for investigation, developing a parallel program and measuring its performance characteristics, submitting the developed software and a written report, and giving a presentation and demo during class. Students taking the course for graduate credit will be expected to include a more significant research component than those taking the course for undergraduate credit.

For more information, see `project.pdf/project.html`. Also, see below for important project due dates.

Mid-term Exam

There will be one mid-term exam; see below for date.
The mid-term exam must be taken at its scheduled time. Make-up mid-term exams will not be administered, unless exceptional circumstances have been discussed with the instructor in advance of the exam date and/or other arrangements have been made.

Final Exam

There will be a final exam; see below for the date. The final will be comprehensive and will cover material from the entire course, including readings, lectures, and assignments.
The final exam must be taken at its scheduled time. Any exam conflicts must be reported to the instructor by the end of Week 6 (see the RIT Final Examination Policies).

Late Policy

Assignments that are submitted electronically (most assignments) will generally be due at 11:59PM on the due date. Assignments that are submitted in person will generally be due at the beginning of a class period.

Assignments are to be submitted on time. However, to accommodate the occasional difficulty with meeting an assignment due date, each student begins the term with five “extension tokens.” By spending an extension token, you will receive a 24-hour extension on a single assignment. To spend an extension token, you must e-mail the instructor before the assignment is due; you cannot spend an extension token after an assignment’s due date has passed. You may spend at most two extension tokens on a single assignment (and you may spend the second extension token at any time before the first extension expires). After spending five extension tokens, late assignments will not be accepted.

Regrading

After a graded exam or assignment has been returned, you have one week to bring any questions about grading to the instructor’s attention. No grade adjustments will be made after this time.
Important Dates

December 6 (Mon.): Project team formation due
December 13 (Mon.): Project proposal due
January 11 (Tue.): Mid-term Exam (in class, 110min)
February 14 (Mon.): Project software and report due
February 15 & 17 (Tue. & Thu.): Project presentations
February 17 (Thu.): Project presentation materials due
February 24 (Thu.): Final Exam (12:30pm – 2:30pm; GOL(70)-2690)

Academic Integrity

As with all courses, the RIT Honor Code and the RIT Academic Honesty Policy apply. See the Department of Computer Science’s statement on academic integrity for more details.

In this course, work submitted for homework and programming assignments must be your own work (i.e., written or programmed by you alone, unless explicitly stated otherwise) and must include acknowledgments of any collaborators or sources (other than course text books or handouts) used to produce your submission.

Obviously, work submitted for the term team project will be a joint effort. Nonetheless, all members of a team should make significant contributions to all submitted work.

You are encouraged to discuss course material with other students. Discussion of assignments is also allowed, but sharing solutions or code is not allowed.

Disclaimer

I reserve the right to make any changes to the syllabus as I deem necessary throughout the course. Minor changes, such as assignment due dates, will be announced orally during class and posted on the course mailing list and home page. Major changes, such as grading percentages, will additionally be provided in writing.

Acknowledgements

Portions of this course material based upon instances of this course offered by Alan Kaminsky (RIT) and Minseok Kwon (RIT).