Computer Science 4000

Introduction to Distributed, Parallel, and Web-Centric Computing

Spring 2020-21

Call #: 2605
Prerequisites:CS3560 and CS 3610
Class meetings: 10:45 a.m. to 11:40 a.m. Via Teams
Text: P.S. Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufman, 2011.
Recommended Readings: Other papers, texts, and web-sites will be used throughout the semester.

Instructor: David Juedes (pronounced "Yedis"), Chair and Professor
Office: 335 Stocker Center, 593-1567
Office hours: 8:30 a. m.– 9:30 a. m. Monday and Friday (via Teams) or by appointment (send email)
Email:`juedes@ohio.edu`
Grader: TBD
WWW: All lecture notes and assignments will be available via BLACKBOARD.
Grading: Four components, weighted as follows will contribute to the final grade.

Online Quizzes	10 %
Written Homework and Programming Projects	45 %
Midterm Exam	20 %
Final Exam	25 %
	100 %

Final grades will be determined by applying the following scale. 90% and above = A, 80% and above = B, 70% and above = C, 60% and above = D, Below 60% = F. This scale may be adjusted downward at the discretion of the instructor.

Important Dates:

First Day of Class	Monday, January 20th, 2021
Wellness Break	Wednesday, March 3rd, 2021
Midterm Exam	Friday, March 5th, 2021
Last Day to Drop	Friday, April 23rd, 2021
Final Exam	asynchronous — during exam week.

Copyright Notice The original lectures, classroom activities, and all materials associated with this class and developed by the instructor are copyrighted in the name of David Juedes on this date January 20th, 2021, unless otherwise noted.

Late Policy: All online quizzes must be completed by the deadline. All graded homework will have a posted deadline. To receive full credit, you must submit the homework electronically (via blackboard) by the posted deadline. If you turn your assignment in after the deadline, you will lose credit based on the following sliding scale: 0 - 2 days (48 hours) late — 10% deduction, 3 – 7 days late — 25% deduction. Homework that is submitted more than 7 days after the due date will receive no credit.

Attendance Policy: Regular class attendance is *strongly* recommended. However, class attendance is not used in the final determination of grades.[1] Students are required to attend class during the midterm exams and the final exam.

Excused Absences: As documented in the Undergraduate Catalog, the university recognizes certain absences as excused. In the case that you miss class because of an excused absence, please contact the instructor as soon as possible. All work missed because of excused absences must be completed.

Accommodations for Students with Disabilities Any student who suspects s/he may need an accommodation based on the impact of a disability should contact the class instructor privately to discuss the student's specific needs and provide written documentation from the Office of Student Accessibility Services. If the student is not yet registered as a student with a disability, s/he should contact the Office of Student Accessibility Services.

Academic Misconduct It is expected that the work you submit is your own. **Unless otherwise noted, all assignments are individual effort.** Plagiarism and other forms of academic misconduct[2] (e.g., cheating on a test) will be handled within the guidelines of the *Student Handbook*. For additional details, see the handout concerning plagiarism that can be found on Blackboard. Students who commit serious acts of academic misconduct **will receive an F in this class** and the Office of Community Standards and Student Responsibility will be notified. That office may impose additional sanctions on students charged with academic misconduct. Please note, however, that students charged with academic misconduct may appeal any academic sanctions given by this professor through the standard grade appeal process, as documented in the Faculty Handbook.

[1] Attendance is noted by the instructor. Potential employers often ask me whether a student attended class regularly. They do this because they want to know if you will be a *reliable employee*. When asked for a reference, I do mention student attendance to potential employers, if asked. In general, I do not consider class attendance to be optional.

[2] Placing homework solutions on public `git` repositories during this class will be considered academic misconduct.

Course Description

Provides an introduction to distributed, parallel, and web-centric computing, along with associated enabling technologies (e.g., networking). Introduces distributed and parallel models of computation, distributed and parallel computer architectures, multi-core designs, potential speed-up, threading, synchronization, and multi-core programming, parallel and distributed algorithms, sockets and client-server based software, web programming, accessing databases across the web, and web-security.

Course Outcomes

Student Outcome 1: Students will have the ability to analyze a complex computing problem, and to apply principles of computing and other relevant disciplines to identify solutions.

1. Students will be able to identify potential race-conditions in parallel and/or multi-threaded code.
2. Students will able to describe and apply how Amdahl's and Gustafson's Laws affect parallel performance.

Student Outcome 2: Students will have the ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

1. Students will be able to design and implement race-free parallel (multi-threaded) code to solve a computing problem.
2. Students will be able to design and implement parallel (multi-threaded) code to solve a computing problem that efficiently uses all parallel resources.
3. Students will be able to design and implement client-server based distributed software.
4. Students will be able to apply the principles of secure computing in the design of distributed/parallel systems.

In additional, students will be able to analyze the local and global impact of computing on individuals, organizations, and society.

1. Students will be able to describe how various issues related to parallel/distributed computing (e.g., freedom of speech and intellectual property) are shaped in multinational and multicultural environments.
2. Students will be able to analyze the impact of parallel/distributed computing solutions on a global workforce.

Course Topics (not necessarily in this order)

1. Basic Terminology and Concepts (3-5 hours)
    - Parallel vs. Distributed vs. Concurrent vs. Web-Centric Computing
    - Sequential and Parallel Models of Computation (Turing Machine and RAM vs. PRAM, Boolean Circuits)
        - Parallel Memory Models and Parallel Architectures
        - Flynn's Taxonomy
        - Practical Considerations: Caching and Coherence in Multicore
    - Introduction to Networking
        - Naming and address schemes (DNS,IP, URI, etc.)
        - Distributed applications/ Socket APIs.
        - HTTP protocol, TCP/UDP, LANs.
2. Parallel Algorithmic Techniques (8-10 lectures)
    - Divide and Conquer
    - Parallel Prefix Computation
    - Map/Reduce
    - Others, e.g., Cache Oblivious and Communication Avoiding Algorithms
3. Parallel/Distributed Performance (Amdahl's Law and Gustafson's Law) (2-3 hours)
4. Data Dependencies and Critical Paths (2-3 hours)
5. Threading, Synchronization, and Multi-Core Programming (10 hours)
    - Race conditions
    - Critical sections
    - OpenMP
    - PThreads
6. Distributed and Cluster Computing (8-10 hours)
    - OpenMPI
    - Hadoop / Spark
    - Sockets and Client Server Programming
    - Related Security Issues.
7. Advanced Topics/ Web Centric Computing (as time permits)
    - Web-Centric Computing/Web Services, Web-Programming Languages
    - Web-Security Issues