CSCI-UA.0480-051: Parallel Computing Midterm Exam (March 14th, 2024)

Total: 100 points
Important Notes- READ BEFORE SOLVING THE EXAM

- If you perceive any ambiguity in any of the questions, state your assumptions clearly and solve the problem based on your assumptions. We will grade both your solutions and your assumptions.
- This exam is take-home.
- The exam is posted on Brightspace, at the beginning of the March 14th lecture (2pm EST).
- You have up to 24 hours to submit on Brightspace (i.e. till March 15th 2pm EST), in the same way as you submit an assignment. However, unlike assignments, you can only submit once.
- Your answers must be very focused. You may be penalized for giving wrong answers and for putting irrelevant information in your answers.
- Your answer sheet must be organized as follows:
  - The very first page of your answer must contain only:
    - Your Last Name
    - Your First Name
    - Your NetID
    - Copy and paste the honor code shown below.
  - In your answer sheet, answer one problem per page. The exam has seven main problems, each one must be answered in a separate page.
- This exam consists of 7 problems, with a total of 100 points.
- Your answers can be typed or written by hand (but with clear handwriting). It is up to you. But you must upload one pdf file containing all your answers.

Honor code (copy and paste to the first page of your exam)

- You may use the textbook, slides, the class recorded lectures, the information in the discussion forums of the class on Brightspace, and any notes you have. But you may not use the internet.
- You may NOT use communication tools to collaborate with other humans. This includes but is not limited to Google-Chat, Messenger, E-mail, etc.
- You cannot use LLMs such as chatGPT, Gemini, Bard, etc.
- Do not try to search for answers on the internet, it will show in your answer, and you will earn an immediate grade of 0.
- Anyone found sharing answers, communicating with another student, searching the internet, or using prohibited tools (as mentioned above) during the exam period will earn an immediate grade of 0.
- "I understand the ground rules and agree to abide by them. I will not share answers or assist another student during this exam, nor will I seek assistance from another student or attempt to view their answers."

Problem 1 (10 points)

Explain the difference between cache coherence and memory consistency in a multi-core system. Provide examples of situations where one might be violated while the other holds.

Problem 2 (15 points)

Describe three different approaches to handling false sharing in shared memory programming. For each approach, briefly explain how it works and its advantages and disadvantages.

Problem 3 (15 points)

Consider a parallel program that sorts an array of integers. Describe how you would implement this using:

(a) (5 points) A divide-and-conquer approach using MPI.

(b) (5 points) A parallel merge sort algorithm using OpenMP.

(c) (5 points) Compare the scalability of the two approaches. Which would be better suited for a very large number of cores and a massive dataset? Justify your answer.

Problem 4 (15 points)

A program uses a single producer and multiple consumer threads to process data. Describe how you would implement a thread-safe bounded buffer using appropriate synchronization primitives (e.g., semaphores, mutexes, condition variables). Explain the role of each primitive in ensuring thread safety.

Problem 5 (15 points)

Suppose you are given a task graph where each node represents a task and the edges represent dependencies between tasks. Explain how you would use a topological sort to schedule the tasks for parallel execution, and discuss any challenges you might encounter.

Problem 6 (15 points)

You have a program that needs to perform a large number of independent calculations. Compare and contrast the use of MPI and OpenMP for parallelizing this program. Discuss the strengths and weaknesses of each approach for this type of problem.

Problem 7 (15 points)

Explain the concept of Amdahl's Law and its implications for the scalability of parallel programs. Give an example of a program with a high serial fraction and discuss why it would be difficult to achieve significant speedup using parallelization.