

# 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 (2 pm EST).
- You have up to 24 hours to submit on Brightspace (i.e., until March 15th, 2 pm 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 in the rectangle at the bottom of this page.
  - In your answer sheet, answer one problem per page. The exam has seven main problems; each one must be answered on 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, Email, 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

- a. [10] Explain the concept of Amdahl's Law. How does it relate to the potential speedup achievable through parallelization?
- b. [10] What are the key differences between OpenMP and MPI? In what scenarios would you choose one over the other?
- c. [10] Describe a scenario where false sharing could significantly degrade performance in a parallel program. How could you mitigate this issue?
- d. [6] What is a race condition? Give a simple code example illustrating a race condition.

## Problem 2

Consider a parallel program that sorts an array of 10 million integers. The program uses 8 threads. The time taken to sort the array sequentially is 10 seconds. Assume the overhead of parallelization is negligible.

- a. [10] If the parallel program achieves perfect linear speedup, what is the runtime of the parallel program?
- b. [10] If the parallel program exhibits only 50

## Problem 3

The following code calculates the sum of elements in an array.

```
#include <omp.h>
#include <stdio.h>

int main() {
    int arr[100];
    int sum = 0;
    #pragma omp parallel for
    for (int i = 0; i < 100; i++) {
        arr[i] = i;
        sum += arr[i];
    }
    printf("Sum: %d\n", sum);
    return 0;
}
```

- a. [15] Explain why this code is incorrect. Create a table showing the potential values of 'sum' depending on the number of threads and the order of execution.
- b. [5] Correct the code to accurately calculate the sum in parallel using OpenMP.

## Problem 4

Consider a system with 4 cores. We have a task graph represented by a Directed Acyclic Graph (DAG) where each node represents a task and each edge represents a dependency between tasks.

- a. [10] Draw a sample DAG with at least 5 nodes and specify the dependencies.
- b. [10] Propose a scheduling algorithm to execute the tasks on the 4 cores, minimizing the overall execution time. Explain your choices.

## Problem 5

- a. [10] Explain the concept of load balancing in parallel computing. Why is it important?
- b. [10] Describe a technique for dynamically load balancing tasks in a parallel program.

## Problem 6

Consider a parallel program that performs matrix multiplication.

- a. [10] Describe a naive parallel implementation of matrix multiplication. What are its limitations?
- b. [10] Describe an optimized parallel implementation that addresses the limitations of the naive approach.

## Problem 7

- a. [5] What is a deadlock in concurrent programming? Give a simple example.
- b. [5] Explain one strategy to prevent deadlocks.