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 ten main problems, each one must be answered on a separate page.
- This exam consists of 10 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

a. [10] Suppose we have a core with only superscalar execution (i.e., no pipelining or hyperthreading). Will this core benefit from having a larger instruction cache? Justify your answer in 1-2 lines.

b. [10] Can a single process be executed on a distributed memory machine? If yes, explain how, in 1-2 lines. If not, explain why not.

c. [10] Can several threads, belonging to different processes, be executed on a shared memory machine and get the same performance as when executed on a distributed memory machine? If yes, explain how, in 1-2 lines. If not, explain why not.

d. [6] If we have an eight-way superscalar core, how many floating-point units (FPUs) would ideally be needed to maximize performance for floating-point intensive tasks? Justify.

Problem 2

Consider a parallel program represented by the following directed acyclic graph (DAG):

(Imagine a simple DAG here with nodes A, B, C, D, E, where A has edges to B and C, B has an edge to D, C has an edge to D, and D has an edge to E. Execution times for each node are: A=2, B=3, C=4, D=1, E=5)

a. [10] What is the critical path length of this DAG?

b. [10] What is the minimum execution time if we have unlimited processors?

Problem 3

Suppose we have a simple program that calculates the sum of numbers in an array of size 1000. We use 10 threads to parallelize this.

a. [10] Describe a strategy to distribute the work among the 10 threads to minimize load imbalance.

b. [10] If the threads finish at different times, what synchronization mechanism could be used to combine the partial sums correctly?

Problem 4

Explain the difference between Amdahl's Law and Gustafson's Law. [20 points]

Problem 5

a. [8] Describe a scenario where using OpenMP would be more efficient than MPI.

b. [7] Describe a scenario where using MPI would be more efficient than OpenMP.

c. [5] Name one advantage and one disadvantage of using hybrid parallel programming (combining MPI and OpenMP).

Problem 6

Consider the following OpenMP code snippet:

```
#pragma omp parallel for
for (int i = 0; i < 1000; i++) {
  // Some computation involving array[i]
}
```

a. [8] What does the `#pragma omp parallel for` directive do?

b. [7] How would you modify this code to ensure that each thread has a private copy of a variable `local`$_{s}um$?

c. [5] How would you modify this code to ensure that only one thread updates a shared variable global$_{s}um$?

Problem 7

Explain false sharing and how it can affect performance in shared memory programming. [20 points]

Problem 8

Design a simple algorithm to perform matrix multiplication using MPI. Describe the data distribution and communication steps. [20 points]

Problem 9

a. [10] What is a race condition? Give an example.
b. [10] Explain how mutexes can prevent race conditions.

Problem 10

Describe three different approaches to handling load imbalance in parallel programs. [20 points]