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 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 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

a. [10] Explain the difference between strong scaling and weak scaling in parallel computing. Give an example of a scenario where each would be beneficial.

b. [10] Describe Amdahl's Law and its implications for parallel program performance. How does the fraction of the program that cannot be parallelized affect the potential speedup?

c. [10] What are the primary challenges in achieving good load balancing in a parallel program? Discuss at least three different challenges and possible approaches to mitigating them.

Problem 2

Consider a parallel program that sorts a large array of integers.

a. [10] Describe a suitable parallel sorting algorithm (e.g., merge sort, quicksort) and explain how it would be implemented using threads or processes.

b. [10] Discuss the factors that would influence the choice between using threads or processes for this parallel sorting algorithm. Consider aspects like communication overhead, memory management, and scalability.

c. [10] Analyze the scalability of the chosen algorithm. How would the runtime scale with the number of processors or threads, assuming a fixed input size? What are the limitations to scalability?

Problem 3

The following DAG represents tasks and their dependencies:

(Imagine a DAG here with 6 nodes: A, B, C, D, E, F. A has no dependencies. B depends on A. C depends on A. D depends on B and C. E depends on D. F depends on D. Execution times are: A=2, B=3, C=4, D=1, E=2, F=3)

Execution times (in milliseconds) for each task on two different CPU types are:

Task	CPU Type X	CPU Type Y
A	2	3
B	3	2
C	4	3
D	1	2
E	2	1
F	3	2

a. [10] Schedule the tasks on two CPU Type X processors to minimize execution time. Show your schedule and calculate the total execution time.

b. [10] Schedule the tasks on three CPU Type Y processors to minimize execution time. Show your schedule and calculate the total execution time.

c. [10] Compare the efficiency of using CPU Type X versus CPU Type Y for this specific DAG. Discuss factors that contributed to the performance differences.

Problem 4

Consider the following OpenMP code snippet:

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

int main() {
  int i, n = 1000;
  double sum = 0.0;

  #pragma omp parallel for reduction(+:sum)
  for (i = 0; i < n; i++) {
    sum += i * i;
  }
  printf("Sum: %f\n", sum);
  return 0;
}
```

a. [8] Explain the role of the `reduction(+:sum)` clause in this OpenMP code. What does it accomplish?

b. [7] How would the performance of this code change if the `reduction` clause were removed?

c. [10] Discuss how the number of threads used by OpenMP would affect the runtime of this code. Would you expect linear speedup? Why or why not?

Problem 5

a. [10] Explain the concept of a race condition in parallel programming. Provide a simple code example demonstrating a race condition.

b. [10] Describe at least three common techniques or mechanisms used to prevent or mitigate race conditions in parallel programs.

Problem 6

Consider a distributed-memory system with four nodes. We want to perform a matrix multiplication of two 1000x1000 matrices.

a. [10] Describe a strategy for distributing the matrices and the computation across the four nodes to achieve parallel execution.

b. [10] What are the major communication challenges in this parallel matrix multiplication? How would you address these challenges?

Problem 7

a. [10] Explain the difference between cache coherence and cache consistency. What are the implications for parallel program performance?

b. [10] Describe a cache coherence protocol (e.g., MESI) and how it ensures that multiple processors have consistent views of shared memory.