

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 in 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 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 multicore with the same number of cores? If yes, explain how, in 1-2 lines. If not, explain why not.
- d. [6] If we have a two-way superscalar core, how many instruction decoders do we need to get the best performance? Justify.

Problem 2

Consider a parallel program that sorts an array of 1024 integers. The program uses a divide-and-conquer approach, recursively splitting the array into smaller subarrays until each subarray contains only one element. Then, it merges the sorted subarrays back together. The table below shows the execution time (in milliseconds) for different numbers of processors:

Number of Processors	Execution Time (ms)
1	100
2	55
4	30
8	20
16	15

- a. [10] Plot the speedup and efficiency for this program. b. [10] Explain the observed speedup and efficiency. Why doesn't the speedup increase linearly with the number of processors?

Problem 3

Consider the following OpenMP code:

```
c++ include <omp.h> include <iostream>
int main() { int sum = 0; pragma omp parallel for reduction(+:sum) for (int i = 0; i < 1000; ++i) { sum
+= i; std::cout << "Sum: " << sum << std::endl; } return 0; }
```

- a. [10] Explain the role of the 'reduction(+:sum)' clause. b. [10] What would happen if the 'reduction' clause were removed? Explain the potential problems.

Problem 4

- a. [5] If we have a data-parallel application. Does it have good scalability? Assume the problem size is big enough and communication overhead is negligible.
- b. [5] If we have four threads, each performing the same computation on a different part of the data. Does this necessarily mean we have perfect load balance? Explain.

Problem 5

Describe three different parallel programming models and give an example of when each would be most appropriate. [20 points]

Problem 6

- a. [10] Explain Amdahl's Law and its implications for parallel program design. b. [10] Explain Gustafson's Law and how it differs from Amdahl's Law.

Problem 7

Consider a task graph with the following dependencies: Task A must complete before Tasks B and C can start. Tasks B and C must complete before Task D can start. Tasks B and C are independent of each other.

a. [10] Draw the task graph. b. [10] Determine the minimum execution time if each task takes 10 units of time and we have 3 processors.

Problem 8

Explain the concept of false sharing in shared memory programming and how it can negatively impact performance. Provide a solution to mitigate false sharing. [20 points]

Problem 9

Compare and contrast the concepts of threads and processes. Include a discussion of their creation, management, and memory usage. [20 points]

Problem 10

Write a short pseudocode or a code snippet (in any language of your choice) to demonstrate how you would implement a parallel prefix sum using a tree-based approach. [20 points]