

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 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

- [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.
- [10] Can a single process be executed across multiple nodes in a distributed memory machine? If yes, explain how, in 1-2 lines. If not, explain why not.
- [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. Assume each node of the shared memory machine has one CPU only.
- [6] If we have a two-way superscalar core, how many register files do we need to get the best performance? Justify.

Problem 2

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

Task A: 5 units of computation time. Task B: 3 units of computation time. Depends on A. Task C: 2 units of computation time. Depends on A. Task D: 4 units of computation time. Depends on B and C.

- [10] Draw the DAG.
- [10] What is the minimum execution time if we have only one CPU?
- [10] What is the minimum execution time if we have two CPUs? Show the CPU assignments.

Problem 3

Suppose we have a program that performs matrix multiplication. The matrix is of size 1000x1000. We are using OpenMP.

- a. [8] Write a simple OpenMP code that divides the rows of the matrix amongst available threads and performs the multiplication.
- b. [7] Explain how you would modify the code to improve performance considering cache efficiency.
- c. [10] Discuss potential issues with scalability as we increase the number of threads.

Problem 4

Explain how load balancing affects the performance of a parallel program. Describe different strategies for achieving effective load balancing in a parallel environment, and discuss the challenges involved in dynamically balancing loads.

Problem 5

Suppose that `MPI_COMM_WORLD` consists of four processes (0, 1, 2, 3). Process 0 sends a message to process 2, and process 1 sends a message to process 3. The messages are sent asynchronously using `MPI_Isend` and `MPI_Irecv`.

- a. [10] Write the MPI code that shows this scenario.
- b. [10] Explain potential problems if the program does not handle message ordering properly.

Problem 6

- a. [10] What are the advantages and disadvantages of using threads versus processes for parallel programming?
- b. [10] Describe a scenario where using threads would be preferable to processes, and vice versa.

Problem 7

Consider a parallel program that sorts an array of 1 million integers using a merge sort algorithm.

- a. [10] Describe how you would parallelize this algorithm using OpenMP.
- b. [10] Analyze the scalability of your parallel implementation.

Problem 8

A program uses a mutex to protect a shared variable. Two threads, Thread A and Thread B, access the shared variable.

- a. [10] Describe a scenario where a race condition could occur even with the mutex.
- b. [10] Explain how to avoid this race condition.

Problem 9

- a. [10] Compare and contrast shared memory and distributed memory parallel programming models.
- b. [10] Give examples of programming paradigms and tools used for each model.

Problem 10

Multiple Choice: Which of the following is NOT a common challenge in parallel programming?

- (a) Race conditions (b) Deadlocks (c) Cache coherency (d) Increased sequential execution time