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 six main problems, each one must be answered in a separate page.
- This exam consists of 6 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 the same process, be executed on a shared memory machine and get the same performance as when executed on a single core? 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 register files do we typically need to get the best performance? Justify.

Problem 2

Consider a task represented by the following Directed Acyclic Graph (DAG):

Task A: 10 seconds Task B: 5 seconds, depends on A Task C: 8 seconds, depends on A Task D: 3 seconds, depends on B and C Task E: 7 seconds, depends on D

Task	Execution Time (seconds)	Dependencies
A	10	-
B	5	A
C	8	A
D	3	B, C
E	7	D

a. [10] What is the minimum execution time of this task sequence if only one CPU is available?

b. [10] Suppose we have two identical CPUs. What is the minimum execution time and how would you schedule the tasks?

c. [10] Suppose we have three identical CPUs. What is the minimum execution time and how would you schedule the tasks?

Problem 3

Consider the following MPI code snippet:

c++ include ¡mpi.h¿ include ¡iostream¿

int main(int argc, char** argv) MPI$_{I}nit(argc, argv); intrank, size; MPI_{C}omm_{r}ank(MPI_{C}OMM_{W}ORLD$

int x = rank; int y = 0; int z = 0;

MPI$_{S}end(x, 1, MPI_{I}NT, (rank+1)MPI_{R}ecv(y, 1, MPI_{I}NT, (rank+size-1)z = x + y;$

std::cout ¡¡ "Process " ¡¡ rank ¡¡ ": x = " ¡¡ x ¡¡ ", y = " ¡¡ y ¡¡ ", z = " ¡¡ z ¡¡ std::endl; MPI$_{F}inalize(); return0;$

a. [9 points] What will be the values of x, y, and z for each of the 3 processes after executing the above code with ‘mpiexec -n 3‘?

Process	x	y	z
P0			
P1			
P2			

b. [5] Is it guaranteed that all processes will print their output in numerical order of their rank? Explain.

c. [5] What will happen if we execute the above code with: ‘mpiexec –n 4‘

d. [5] What will happen if we remove the modulo operator (‘

Problem 4

a. [10] Explain the concept of Amdahl’s Law and its implications for parallel computing scalability.

b. [10] Explain the concept of Gustafson’s Law and how it differs from Amdahl’s Law.

Problem 5

Consider a parallel program that sorts an array of 1 million integers. The program divides the array into chunks and sorts each chunk independently using a parallel merge sort.

a. [10] What are the potential sources of overhead in this parallel program (e.g., communication, synchronization, load imbalance)?

b. [10] How could you mitigate or reduce the overhead identified in part (a)?

Problem 6

a. [10] Describe a scenario where using threads instead of processes would be advantageous. Justify your answer.

b. [10] Describe a scenario where using processes instead of threads would be advantageous. Justify your answer.