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 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

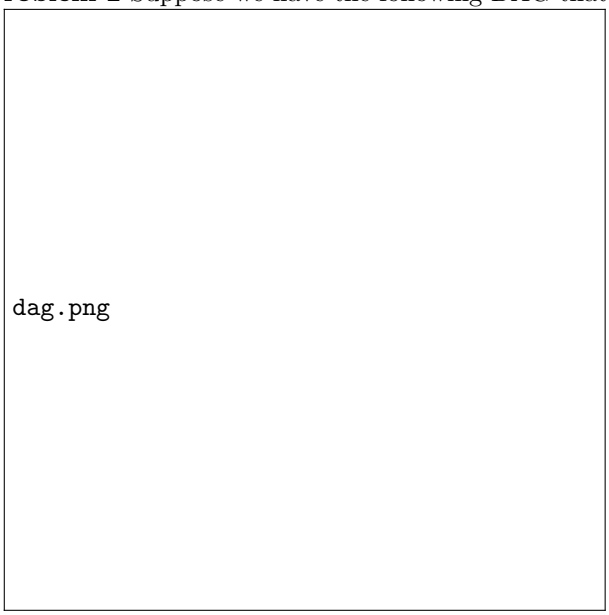
10 Suppose we have a core with only superscalar execution (i.e. no pipelining or hyperthreading). Will this core benefit from having a larger instruction cache? Justify your answer in 1-2 lines.

10 Can several threads be executed on a distributed memory machine? If yes explain how, in 1-2 lines. If not, explain why not. Assume each node has multiple CPUs.

10 Can a single process utilize multiple cores on a shared memory machine? If yes, explain how, in 1-2 lines. If not, explain why not.

6 If we have an eight-way superscalar core, how many instruction fetch units would be ideal for best performance? Justify.

Problem 2 Suppose we have the following DAG that represents different tasks and their dependencies.



The following table shows the execution time of each task if we execute it on a core of type A and if we execute it on core of type B. Each CPU type is optimized for some type of operations. That is, one type of CPU is not always faster than the other type for all tasks. You can ignore communication overhead among tasks.

Task	CPU A (ms)	CPU B (ms)
T1	5	10
T2	10	5
T3	8	12
T4	12	8
T5	7	9
T6	9	7

10 What is the minimum number of CPUs of each type that we need to get the highest speedup over sequential execution on CPU of type A? Show which CPU will execute which task(s) and calculate the final speedup.

10 Repeat the problem above but using CPU of type B.

10 Suppose we discover a way to parallelize T3 and T4, removing the dependency between them. How would this change the optimal scheduling and speedup?

Problem 3 Suppose that MPI_COMM_WORLD consists of the four processes 0, 1, 2, and 3, and suppose the following code is executed after MPI has been initialized (`my_rank` contains the rank of the executing process):

```
int x, y, z;
switch(my_rank) {
```

```
case 0:
x=1; y=2; z=3;
MPI_Send(&x, 1, MPI_INT, 1, 1, MPI_COMM_WORLD);
MPI_Recv(&y, 1, MPI_INT, 3, 2, MPI_COMM_WORLD, &status);
break;
case 1:
x=4; y=5; z=6;
MPI_Recv(&x, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, &status);
MPI_Send(&y, 1, MPI_INT, 2, 3, MPI_COMM_WORLD);
break;
case 2:
x=7; y=8; z=9;
MPI_Recv(&y, 1, MPI_INT, 1, 3, MPI_COMM_WORLD, &status);
MPI_Send(&x, 1, MPI_INT, 3, 4, MPI_COMM_WORLD);
break;
case 3:
x=10; y=11; z=12;
MPI_Recv(&x, 1, MPI_INT, 2, 4, MPI_COMM_WORLD, &status);
MPI_Send(&z, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
break;
}
```

9 What will be the values of x, y, and z for each of the 4 processes after executing the above code? Assume MPI$_I nit has been called.$

	x	y	z
P0			
P1			
P2			
P3			

5 Is there a deadlock in this code? Explain why or why not.

5 What will happen if we remove the 'break;' statements from the 'switch' cases?

5 Suppose process 0 crashes before sending its message. How would this affect the other processes?

Problem 4

5 If we have an application with significant data dependencies between tasks. Does it have good scalability (i.e. as we keep increasing the number of cores, do we see speedup)? Explain.

5 If we have two threads that perform the same type of computation but one has significantly more data to process than the other. Does this necessarily mean we have load imbalance? Explain.

Problem 5 Explain the difference between OpenMP and MPI in terms of:

5 Programming model (shared vs. distributed memory)

5 Communication mechanisms

5 Scalability characteristics

5 Ease of use and programming complexity.

Problem 6 Consider a parallel program that sorts a large array of numbers.

10 Describe a parallel sorting algorithm suitable for this task, explaining how it would be implemented using a suitable parallel programming model (e.g., MPI or OpenMP).

10 Discuss the factors that affect the performance of your chosen algorithm, including the size of the array, the number of processors, and communication overhead.