

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 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

- [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.
- [10] Can several processes, belonging to different users, be executed on a shared memory machine concurrently without interference? 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 decoders do we need to get the best performance? Justify.

Problem 2

A directed acyclic graph (DAG) represents the dependencies between tasks in a parallel computation. The tasks are represented by nodes, and the dependencies are represented by edges. Consider a DAG with the following tasks and dependencies: Task A depends on nothing. Task B depends on A. Task C depends on A. Task D depends on B and C. Task E depends on D.

- [10] Draw the DAG representing these tasks and dependencies.
- [10] What is the minimum number of time steps required to execute this DAG on a machine with three processors? Show your scheduling.

Problem 3

Consider the following C code snippet:

```
c include <pthread.h> include <stdio.h>
int sum = 0; pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
void *worker(void *arg) { int i, myid = *(int *)arg; for (i = 0; i < 1000; i++)
pthread_mutex_lock(&mutex); sum += myid; pthread_mutex_unlock(&mutex); return NULL;
int main() { pthread_t t1, t2; int id1 = 1, id2 = 2; pthread_create(&t1, NULL, worker, &id1); pthread_create(&t2, NU
return 0;
```

a. [15] What will be the output of this program? Explain. b. [5] Is there a race condition? Explain.

Problem 4

A program performs matrix multiplication of two NxN matrices.

a. [10] Describe a parallel algorithm to perform this computation, suitable for a shared-memory multiprocessor system. Be specific about data partitioning and synchronization. b. [10] Analyze the scalability of your algorithm. What factors limit its performance as the number of processors increases?

Problem 5

a. [10] Explain the concept of Amdahl's Law. How does it relate to the scalability of parallel programs? b. [10] Explain the concept of Gustafson's Law. How does it differ from Amdahl's Law?

Problem 6

Consider a system with 8 cores. A program has a sequential portion that takes 20 seconds to execute and a parallelizable portion that takes 60 seconds to execute sequentially.

a. [8] Using Amdahl's Law, calculate the speedup achieved if the parallelizable portion is perfectly parallelized on the 8 cores. b. [7] Using Gustafson's Law, calculate the scaled speedup if the problem size is increased to maintain a constant execution time on a single core.

Problem 7

a. [5] What is false sharing? How does it affect performance in a shared-memory system? b. [5] Describe techniques to mitigate false sharing.