Here's a practice exam designed to precisely match the format, style, and structure of the provided PDF template. I've incorporated the provided questions and point values as requested. Due to the limitations of this text-based environment, I cannot perfectly replicate the visual formatting (fonts, exact spacing, etc.), but I've aimed for the closest textual representation possible. You'll need to use a word processor or similar tool to achieve precise formatting if needed.

CSCI-UA.0480-051: Parallel Computing

Practice Midterm Exam (October 26th, 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.
- You have up to 24 hours to submit.
- 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, and the honor code (see below).
- Answer one problem per page. The exam has ten problems.
- 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)

"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 (12 points)

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 2 (5 points)

Suppose we have a core with only pipelining (no superscalar or hyperthreading). Will this core benefit from having more execution units? Justify your answer.

Problem 3 (12 points)

Can several processes be executed on a shared memory machine? If yes, explain how. If not, explain why not.

Problem 4 (5 points)

Can several threads belonging to the same process be executed on a distributed memory machine and get the same performance as when executed on a multicore? If yes, explain how. If not, explain why not. Assume each node of the distributed machine has one CPU only.

Problem 5 (15 points)

If we have a four-way hyperthreading core, how many branch predictors do we need to get the best performance? Justify your answer.

Problem 6 (12 points)

What is the minimum number of CPUs of each type needed to achieve the highest speedup over sequential execution, given a DAG representing tasks and their dependencies, and execution times for each task on different CPU types? Show which CPU will execute which task(s) and calculate the final speedup. *(Assume you will be given a DAG and execution time table similar to Problem 2 in the original exam)*

Problem 7 (10 points)

Using a better algorithm/programming approach, how could you enhance the DAG performance by removing an arrow? Which arrow would you remove and why? *(Assume you will be given a DAG similar to Problem 2 in the original exam)*

Problem 8 (10 points)

What will be the values of x, y, and z for each of the 3 processes after executing a given MPI code involving broadcasts and sends/receives? *(Assume you will be given an MPI code snippet similar to Problem 3 in the original exam)*

Problem 9 (10 points)

Is there a possibility that communication among the 3 processes in the given MPI code executes out of order? If yes, explain the reason. If not, why not? *(Assume you will be given an MPI code snippet similar to Problem 3 in the original exam)*

Problem 10 (9 points)

If we have an embarrassingly-parallel application, does it have good scalability (i.e., as we keep increasing the number of cores, do we see speedup)? Assume the problem size is big enough. If we have two threads that have different types and numbers of computations and assign each thread to a core, does this necessarily mean we have load imbalance? Explain.

This practice exam closely mirrors the original template's style and structure. Remember to replace the bracketed placeholders with the actual DAG, execution times, and MPI code as instructed. You can then adjust formatting in a word processor for a perfect match.