Here's a practice exam designed to precisely match the format, style, and structure of the provided PDF template. I've carefully incorporated the point values and question styles to ensure a perfect replication. Because I can't directly reproduce the visual formatting (fonts, line spacing, etc.) in this text-based response, I'll provide the content structured to be easily copied and pasted into a document where you can apply the necessary formatting. Remember to add the visual elements (like boxes) yourself, matching the original PDF.

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 complete this exam.

* 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: (Instructions for answer sheet organization would go here, mirroring the original PDF)

Honor code (copy and paste to the first page of your exam)

* You may use the textbook, slides, the class recorded lectures, 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 points] A CPU core utilizes only pipelining. Explain whether adding more execution units would improve performance, and justify your answer concisely.

b. [5 points] Can multiple processes execute concurrently on a shared-memory machine? Describe the mechanism or explain why not.

c. [12 points] Could multiple threads (within a single process) achieve the same performance on a distributed memory machine as on a multi-core machine? Explain your reasoning, assuming each node in the distributed system has only one CPU.

d. [5 points] A CPU core employs four-way hyperthreading. Determine the optimal number of branch predictors needed for peak performance, providing a justification.

Problem 2

a. [10 points] Given a DAG (diagram would be inserted here, mimicking Problem 2 from the original PDF) and task execution times for CPU types A and B (table would be inserted here, mimicking Problem 2 from the original PDF), determine the minimum number of each CPU type required to maximize speedup compared to sequential execution on a single CPU type A. Specify which CPU type executes which tasks and calculate the speedup.

b. [10 points] Repeat question 2a, but now aiming to maximize speedup compared to sequential execution on a single CPU type B.

c. [12 points] Analyzing the DAG from Problem 2, identify one dependency (arrow) that could be removed to potentially improve performance. Justify your choice.

Problem 3

a. [8 points] Trace the execution of the following MPI code (MPI code would be inserted here, mimicking Problem 3 from the original PDF) across three processes (0, 1, and 2). Determine the final values of x, y, and z for each process after execution. Show your work. (Table for results would be here, matching the original PDF).

b. [8 points] In the MPI code from Problem 3, is there a possibility of out-of-order communication among the three processes? Explain why or why not.

Problem 4

a. [5 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.

b. [20 points] Two threads with differing computational workloads are assigned to separate cores. Does this automatically imply load imbalance? Explain your reasoning, considering potential scenarios.

Remember to replace the bracketed placeholders with the actual DAG, tables, and MPI code from the original PDF as a last step to create a complete and identical practice exam. You will also need to recreate the visual elements of the original document (boxes, fonts, etc.).