

Parallel Computing Practice Exam

Generated on June 26, 2025

Difficulty: Medium

1. Consider a parallel program designed to compute the Mandelbrot set using a divide-and-conquer approach. Describe how you would partition the complex plane amongst multiple processors to minimize communication overhead and maximize parallel efficiency. Explain the potential bottlenecks that might arise and propose strategies to mitigate them.
2. You are tasked with implementing a parallel algorithm to sort a large dataset using a merge sort approach. Detail the steps involved in parallelizing the merge sort algorithm, specifically addressing how the splitting and merging phases can be efficiently performed across multiple cores. Discuss the trade-offs between the number of processors used and the overall execution time.
3. Analyze the performance of a parallel program that uses a shared memory model. Explain how false sharing can negatively impact performance and propose techniques, such as padding or restructuring data, to reduce its effects. Illustrate your answer with a specific example.
4. Compare and contrast the characteristics and suitability of MPI (Message Passing Interface) and OpenMP for implementing a parallel algorithm to solve a system of linear equations using Gaussian elimination. Consider factors like scalability, ease of programming, and communication overhead in your comparison. Justify your choice of the more appropriate parallel programming model for this specific task.
5. A parallel program exhibits a speedup of 4 when run on 8 processors. Calculate the efficiency of the parallel program and explain the potential reasons why the speedup is not ideal (i.e., less than 8). Discuss factors that may contribute to this sub-linear speedup, including communication overhead, load imbalance, and synchronization delays.