Parallel Computing Practice Exam

Generated on June 26, 2025

Difficulty: Medium

---

1. A parallel program is designed to process a large dataset of 10 million integers, sorted into 1000 equal-sized chunks. Each chunk is processed independently by a separate thread. Describe a potential race condition that could occur if the program needs to calculate the global average of these integers. Explain how you would modify the program to avoid this race condition, detailing the synchronization mechanisms you would employ.

2. Consider the task of multiplying two large matrices, A (1000x500) and B (500x2000), using a parallel algorithm. Explain how you would partition these matrices for optimal performance on a system with 8 processing cores, taking into account data locality and communication overhead. Justify your chosen partitioning strategy.

3. Compare and contrast the performance characteristics of two common parallel programming models: message passing (MPI) and shared memory (OpenMP). Discuss scenarios where one model would be preferable to the other, considering factors like scalability, programming complexity, and hardware architecture.

4. A parallel program experiences significant slowdown as the number of processing cores increases beyond 16, despite the problem size remaining constant. This behavior suggests a scaling bottleneck. Identify at least three potential causes for this slowdown, providing specific reasons for why each could lead to reduced performance with a greater number of cores.

5. You are tasked with implementing a parallel algorithm to find the k-th smallest element in an unsorted array of n elements (where k << n). Describe a parallel algorithm suitable for this task and analyze its time complexity in terms of n and the number of processors p. Consider the limitations and potential inefficiencies of your chosen approach.