# Parallel Computing Practice Exam

Generated on June 26, 2025
Difficulty: Medium

---

**1. Consider a parallel algorithm for matrix multiplication using a distributed memory system with p processors. Describe how you would distribute the matrices amongst the processors to minimize communication overhead and maximize parallel efficiency. Discuss the trade-offs involved in choosing different data distribution strategies, such as block-cyclic versus block distribution, considering the impact of matrix dimensions and the number of processors.**

**2. Explain the concept of Amdahl's Law and its implications for the speedup achievable through parallelization. Provide a numerical example demonstrating how a small portion of a program that cannot be parallelized can significantly limit the overall speedup, even with a large number of processors.**

**3. Compare and contrast the performance characteristics of OpenMP and MPI for parallel programming. Give specific examples of programming scenarios where one approach would be significantly more advantageous than the other, justifying your choices based on the underlying programming models and communication mechanisms.**

**4. A parallel program is designed to perform a complex numerical simulation. Profiling reveals that a significant portion of the execution time is spent in a specific function that involves accessing a shared data structure. Describe several strategies to improve the performance of this function, focusing on reducing contention and improving cache coherence. Explain the potential drawbacks of each strategy.**

**5. You are tasked with implementing a parallel algorithm to sort a large dataset using a message-passing paradigm. Describe a suitable algorithm, such as a parallel merge sort or a parallel quicksort, outlining the steps involved in data partitioning, sorting sub-arrays, and merging the results. Explain how load balancing is addressed in your chosen algorithm and how you would handle potential bottlenecks related to communication and synchronization.**