# Parallel Computing Practice Exam

Generated on June 26, 2025
Difficulty: Medium

**1. Describe the difference between shared-memory and distributed-memory parallel architectures. Provide an example application that would be better suited for each architecture and justify your choices, focusing on the communication overhead and data access patterns.**

**2. Consider a parallel algorithm for matrix multiplication using a 4x4 matrix and 4 processors. Illustrate how the workload would be divided among the processors using a suitable parallel programming paradigm (e.g., master-slave, data parallelism). Detail the steps involved in the computation and identify potential bottlenecks.**

**3. Explain Amdahl's Law and its implications for the speedup achievable through parallelization. Provide a concrete example of a program with a sequential portion and a parallelizable portion, and calculate the maximum speedup achievable with an infinite number of processors, assuming the sequential portion takes 20% of the total execution time.**

**4. Compare and contrast two different synchronization primitives (e.g., mutexes, semaphores, barriers) used in parallel programming. Describe a scenario where one primitive would be more suitable than the other, clearly explaining the reasons behind your choice and the potential consequences of using the less suitable alternative.**

**5. A parallel program is designed to sort a large dataset using a merge-sort algorithm. Discuss the challenges in efficiently parallelizing this algorithm, including the optimal number of processors and potential data partitioning strategies. Describe how load balancing would be addressed to ensure even distribution of work across the processors and prevent performance degradation.**