Parallel Computing Practice Exam

Generated on June 26, 2025
Difficulty: Medium

---

1. Consider a parallel program designed to compute the Mandelbrot set using a grid-based approach. Describe two different strategies for distributing the workload across multiple processors, discussing the advantages and disadvantages of each approach in terms of load balancing and communication overhead. Include considerations for data dependencies between grid cells.

2. A parallel algorithm uses a master-worker paradigm to process a large dataset. The dataset consists of 1000 independent tasks, each requiring 10 seconds of computation time on a single processor. Assuming we have 10 processors available, calculate the ideal minimum runtime assuming perfect load balancing. If the actual runtime is 120 seconds, identify potential sources of overhead contributing to the difference.

3. Explain the concept of Amdahl's Law and how it relates to the potential speedup achievable through parallelization. Provide a specific example of a program with a sequential component and calculate the maximum possible speedup achievable with an arbitrarily large number of processors, assuming the sequential component takes 20% of the total execution time.

4. Compare and contrast two different synchronization primitives: mutexes and semaphores. Describe scenarios where each is best suited and illustrate their usage with a simple example showing how they prevent race conditions in a concurrent program accessing a shared resource (e.g., a counter).

5. Describe the challenges associated with debugging parallel programs compared to sequential programs. Discuss at least three specific debugging challenges and explain techniques or tools that can help mitigate these challenges, explaining how they aid in identifying and resolving concurrency-related issues such as deadlocks or race conditions.