PRERANA EDUCATIONAL AND SOCIAL TRUST®
PES Institute of Technology and Management
NH-206, Sagar Road, Shivamogga-577204

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
[DESIGN]

Parallel Computing [BCS702]

2025-26

QUESTIONS BANK

MODULE 1

1. What are the two main classifications of parallel computers?
2. Define SIMD and MIMD systems.
3. List any four interconnection network types used in parallel computing.
4. What is cache coherence?
5. Identify whether the following is a shared-memory or distributed-memory system (provide examples).
6. Explain the key differences between SIMD and MIMD architectures.
7. Describe the concept of shared-memory and how it differs from distributed-memory systems.
8. Why is cache coherence important in shared-memory systems?
9. Summarize the role of interconnection networks in parallel hardware.
10. Distinguish between thread-level and process-level parallelism.
11. Apply the concept of shared-memory architecture to illustrate how threads communicate in OpenMP.
12. Given a simple problem, demonstrate how MIMD systems can execute it in parallel.
13. Use a diagram to show how an interconnection network supports communication between processors.
14. Illustrate how to coordinate processes in a distributed-memory system using MPI.
15. Apply knowledge of cache coherence to identify performance issues in a multi-threaded program.

MODULE 2

1. Define MIMD (Multiple Instruction, Multiple Data) system.
2. What is speedup in the context of parallel computing?
3. State Amdahl's Law.
4. List the key components of a hybrid system involving CPU and GPU
5. Explain the difference between GPU and CPU architectures.
6. Describe how speedup and efficiency are calculated in MIMD systems.
7. Interpret the meaning of Amdahl's Law and its implication on scalability.
8. Explain how GPU performance can be measured and improved.
9. Compare shared-memory and hybrid computing models.
10. Apply Amdahl's Law to calculate the theoretical speedup when 80% of a task is parallelized over 4 processors.
11. Use timing functions to measure and compare execution time of a sequential vs. parallel MIMD program.
12. Given a real-world task (e.g., image processing), show how to implement it using GPU for performance gain.
13. Apply knowledge of hybrid systems to design a simple CPU-GPU cooperative task.
14. Demonstrate performance scaling by running a parallel code with increasing processor count and plotting speedup

MODULE 3

1. Name any four basic MPI functions used in every MPI program.
2. Define collective communication in MPI.
3. What is the purpose of `MPI_Comm_rank` and `MPI_Comm_size`?
4. List any two MPI collective communication functions.
5. Explain the role of the trapezoidal rule in demonstrating parallel computation with MPI.
6. Describe how MPI handles communication between processes.
7. Explain how MPI-derived datatypes can help in structuring communication.
8. Summarize the difference between point-to-point and collective communication in MPI.
9. Describe the need for performance evaluation in MPI programs
10. Apply `MPI_Bcast` to distribute input data from the root process to all other processes.
11. Implement the trapezoidal rule in MPI to approximate definite integrals in parallel.
12. Use MPI functions to write a parallel program that sorts a set of numbers using a distributed sorting algorithm.
13. Apply MPI file I/O routines to write output from multiple processes into a common file.
14. Evaluate the performance of an MPI-based matrix multiplication program by measuring execution time with increasing process count.