RAMAKRISHNA MISSION VIDYAMANDIRA

NEP Syllabus B.Sc. Computer Science Honours

Semester-VI

Course Code: 6CMSMJC3

Course Type: Major Course

Course Outcome:

- Define and explain the need, evolution, architectures, and classifications of high-performing parallel computing systems.
- Illustrate and analyze the factors affecting instruction-level parallelism such as resource constraints, data dependencies, and branch delays in pipeline execution.
- Design parallel algorithms for searching, sorting, and matrix operations using appropriate computational models and evaluate performance metrics.
- Develop and execute message-passing, shared-memory, and heterogeneous parallel programs using MPI, OpenMP, and CUDA respectively.
- Compare multicore CPU and GPU execution models to select suitable architectures for different classes of parallel applications.

6CMSMJC3: High Performance Parallel Computing

Credit: 3 **Marks: 50**

Introduction to High Performance Parallel Computing: Need for high-speed computing. Need for parallel computers, Evolution of supercomputing, Modern parallel computers, Need for writing parallel programs, Challenges in parallel programming, Data parallelism, task parallelism, Comparison between data parallelism and task parallelism. **[4L]**

Architecture of Parallel Computers: Evolution of Parallel Computing Architecture, Flynn's classification, Coupling between processing elements, Classification based on mode of accessing memory, Vector processors, Superscalar processors, Very Long Instruction Word (VLIW) processor, Interconnection networks: Networks to interconnect processors to memory or computers to computers, Direct interconnection of computers. **[4L]**

Instruction Level Parallel Processing: Pipelining of processing elements, Delays in pipeline execution: Delay due to resource constraints, data dependency and branch instructions, Hardware modifications and Software methods to reduce delay due to branches. **[4L]**

Multicore Processors: Motivation for Multicore Systems, Fundamentals of Multicore Architecture: homogeneous vs heterogeneous cores, Memory hierarchy and shared cache organization, Multithreading in Multicore Systems, Case Study: Intel x86 multicore processor. **[6L]**

Introduction to GPU: GPU Architecture Basics: Single Instruction, Multiple Threads (SIMT) architecture, GPU Execution Model, Differences in execution model CPU vs GPU, Memory Organization in GPUs, GPU Programming Concepts, Applications of GPUs, Case Study: NVIDIA CUDA architecture. **[6L]**

Parallel Algorithms: Models of computation: The Random Access Machine (RAM), The Parallel Random Access Machine (PRAM), Parallel algorithm design for searching, sorting and matrix operations, analysis of parallel algorithms: Running time, Number of processors, Cost. **[8L]**

Parallel Programming: Different programming paradigms, Message passing programming with Message Passing Interface (MPI): Point-to-point communication, Collective communication, Shared memory programming with Open Multi-Processing (OpenMP): OpenMP basics and execution model, Synchronization and coordination, Heterogeneous programming with Compute Unified Device Architecture (CUDA): CUDA programming model, GPU memory model and data management, Optimization strategies. **[8L]**

Performance Evaluation of Parallel Computers: Basics of performance evaluation: Performance metrics, Performance measures and benchmarks, Sources of parallel overhead: Load imbalance, Synchronization, Parallel balance point, Speedup performance laws: Amdahl’s law, Gustafson’s law, Sun and Ni’s law. **[5L]**

6CMSMJC3 (Practical): High Performance Parallel Computing Laboratory

Credit: 1 Marks: 25

Implementation of searching, sorting, matrix operations and different other case studies using message passing programming with Message Passing Interface (MPI), shared memory programming with Open Multi-Processing (OpenMP), heterogeneous programming with Compute Unified Device Architecture (CUDA).

Access to HPC platforms**.** **[30L]**

Recommended Books

1. Introduction to Parallel Computing by Ananth Grama, George Karypis, Vipin Kumar, and Anshul Gupta, Addison Wesley.
2. Advanced Computer Architecture: Parallelism, Scalability, Programmability by Kai Hwang, Naresh Jotwani, McGrawHill.
3. Computer Architecture - A Quantitative Approach by John L. Hennessy and David A. Patterson, Sterling Book Centre.
4. An Introduction to Parallel Programming by Subodh Kumar, Cambridge University Press.
5. An Introduction to Parallel Programming by Peter S. Pacheco, Morgan Kaufmann.
6. Parallel Programming in C with MPI and OpenMP by M J Quinn, McGrawHill.
7. Programming Massively Parallel Processors: A Hands-on Approach by David Kirk and Wen mei Hwu, Morgan Kaufmann.