MCA502 PARALLEL AND DISTRIBUTED COMPUTING

L	T	P	Cr
3	0	2	4

Course Objective: To learn the concepts of Parallel and Distributed Computing and its implementation for assessment of understanding the course by the students

Parallelism Fundamentals : Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations, Programming Constructs for creating Parallelism, communication, and coordination. Programming errors not found in sequential programming like data races, higher level races, lack of liveness

Parallel Architecture : Architecture of Parallel Computer, Communication Costs, parallel computer structure, architectural classification schemes, Multicore processors, Memory Issues : Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, vector processing, GPU, co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA)

Parallel Decomposition and Parallel Performance: Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task-Based Decomposition, Data Parallel Decomposition, Actors and Reactive Processes, Load balancing, Data Management, Impact of composing multiple concurrent components, Power usage and management. Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance Power Use and Management, Cost-Performance trade-off;

Communication and Coordination: Shared Memory, Consistency, Atomicity, Message-Passing, Consensus, Conditional Actions, Critical Paths, Scalability,cache coherence in multiprocessor systems, synchronization mechanism.

CUDA programming model: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device, API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, Developing kernel function that will be executed by threads in the parallelized part, Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

Parallel Algorithms design, Analysis, and Programming : Parallel Algorithms, Parallel Graph Algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, Naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms

Laboratory Work: To implementparallel programming using CUDA with emphasis on developing applications for processors with many computation cores, mapping computations to parallel hardware, efficient data structures, paradigms for efficient parallel algorithms

Recommended Books

1. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley 2008.
2. A Grama, AGupra, G Karypis, V Kumar. Introduction to Parallel Computing (2nd ed.). Addison Wesley, 2003.
3. B Gaster, L Howes, D Kaeli, P Mistry, and D Schaa. Heterogeneous Computing With Opencl. Morgan Kaufmann and Elsevier, 2011.
4. T Mattson, B Sanders, B MassingillPatterns for Parallel Programming. Addison-Wesley 2004.
5. Quinn, M. J. (2004). Parallel Programming in C with MPI and OpenMP. McGraw-Hill.