L	T	P	C
4	0	0	4

Course Code: CSE409
Semester: VII

PARALLEL & DISTRIBUTED SYSTEMS

Course Objectives

This course will help the learner to design parallel and distributed algorithms and demonstrate them using CUDA.

UNIT - I — 15 Periods

Heterogeneous Parallel Computing with CUDA: Parallel Computing - Heterogeneous Computing - **CUDA Programming Model:** Timing Your Kernel - Organizing Parallel Threads - **Global Memory** - CUDA memory model - Memory Management - **Shared Memory and Constant Memory:** Shared Memory Allocation - Banks and Access Mode - Configuring the Amount of Shared Memory - Synchronization - Constant Memory - **Streams and Concurrency:** Introducing Streams and Events - **Tuning Instruction-level primitives:** CUDA Instructions

UNIT - II — 15 Periods

Parallel Processing: Introduction - Parallel Processing Terminology - The Sieve of Eratosthenes - **PRAM Algorithms:** Parallel Reduction - Prefix sums - List Ranking - Pre-order Tree Traversal - Merging of two sorted Lists - Graph coloring - **Matrix Multiplication:** Algorithms for processor Arrays - **Sorting:** Enumeration sort - Odd Even transposition sort- Parallel Quick sort - Hyper quick sort

UNIT - III — 15 Periods

Introduction: Design goals - **Types of distributed systems:** High performance distributed computing - Distributed information systems - Pervasive systems - **Architecture:** System architecture - **Communication:** Message-oriented communication - Simple transient messaging with sockets - Advanced transient messaging - Message-oriented persistent communication - **Multicast communication**: Application-level tree-based multicasting - Flooding-based multicasting - Gossip-based data dissemination

UNIT - IV — 15 Periods

Coordination: Clock Synchronization - Logical clocks - **Mutual Exclusion:** Centralized algorithm - Distributed Algorithm - Token-ring algorithm - Decentralized Algorithm - **Election Algorithms**: Bully algorithm - Ring algorithm - Elections in wireless environment and large scale systems - **Fault Tolerance:** Introduction to fault tolerance - Concepts - Failure models - Failure masking by redundancy - **Reliable client server communication:** Point to point communication - RPC semantics in the presence of failures - **Reliable Group Communication:** Atomic multicast - Distributed commit - Recovery

TEXTBOOKS

1. Maarten van Steen and Andrew S. Tanenbaum. *Distributed Systems,* Prentice Hall of India, Third Edition, 2017.
2. John Cheng, Max Grossman and Ty McKercher. *Professional CUDA C programming*, John Wiley & Sons Inc., 2014.
3. Michael J. Quinn. *Parallel Computing Theory and Practice*, McGraw Hill, Second Edition, 2011.

REFERENCES

1. Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General Purpose GPU Programming*, Addison-Wesley, 2011.
2. Andrew S. Tanenbaum. *Distributed Operating System*, Prentice Hall of India, 2006.
3. Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar. *Introduction to Parallel Computing*, Pearson Education, Second Edition, 2003.

ONLINE MATERIALS

1. http://nptel.ac.in/courses/106102114/
2. http://nptel.ac.in/courses/106106107/

LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

Unit I	• Recognize the properties of CUDA enabled device • Design kernel using different types of memory with multiple blocks and multiple threads
Unit II	• Classify the concepts of parallel computing • Distinguish various PRAM algorithms and parallel algorithms for matrix multiplication and sorting
Unit III	• Understand the need and nature of distributed concepts • Evaluate transient and persistence message oriented communication • Understand the multi-cast communication
Unit IV	• Demonstrate clock synchronization with fault tolerance • Formulate group communication • Construct distributed commit and recovery procedures

COURSE LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

- Write programs to explore parallel programming, thread synchronization and atomics using CUDA environment
- Apply PRAM algorithms and parallel algorithms for matrix multiplication and sorting
- Appraise the software and hardware characteristics of distributed systems
- Discuss the concept of system architecture and style of distributed systems
- Demonstrate message oriented communication in distributed environment
- Formulate group communication with clock synchronization, fault tolerant mechanisms, distributed commit and recovery procedures