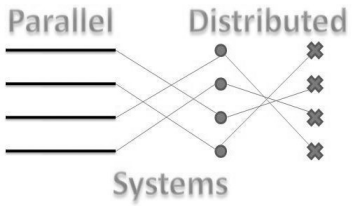
22CS311 PARALLEL AND DISTRIBUTED COMPUTING



Source: http://uceou.edu/PDS/About%20PDS.html

Hours Per Week :

L	T	P	C
2	2	0	3

PREREQUISITE KNOWLEDGE: Fundamental of programming and operating system, computer networking, and computer Organization..

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the challenges faced in constructing parallel and distributed applications, Various implementation techniques, paradigms, architectures and parallel algorithms. And current trends in parallel and distributed computing like Open MP, POSIX Threads, Apache Hadoop (DFS).

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

INTRODUCTION TO PARALLEL COMPUTING

The idea of Parallelism, Power and potential of parallelism, examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instructions streams.

Parallel Architecture: Pipeline architecture, Array processor, Multi-processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, 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).

UNIT-2 **8L+8T+0P=16 Hours**

PARALLEL ALGORITHM DESIGN PRINCIPLES AND PROGRAMMING

Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task- Based Decomposition, Data Parallel Decomposition, Characteristics of task and interaction, Load balancing, Data Management, parallel algorithm models, Sources of overhead in parallel programs, Performance metrics for parallel algorithm implementations, Parallel algorithmic patterns like divide and conquer, Map and Reduce, Specific algorithms like parallel Merge Sort, Parallel graph Algorithms.

PRACTICES:

- Identify Multiple Instruction Single Data, or MISD. How would an MISD system work? Give an example.
- Suppose a shared-memory system uses snooping cache coherence and write-back caches. Also suppose that core 0 has the variable x in its cache, and it executes the assignment x = 5. Finally suppose that core 1 doesn't have x in its cache, and after core 0's update to x, core 1 tries to execute y = x. What value will be assigned to y? Why?
- Consider a simplified version of bucket-sort. You are given an array A of n random integers in the range [1...r] as input. The output data consist of r buckets, such that at the end of the algorithm, Bucket i contains indices of all the elements in A that are equal to i.
    - Describe a decomposition based on partitioning the input data (i.e., the array A) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.
    - Describe a decomposition based on partitioning the output data (i.e., the set of r buckets) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.

SKILLS:

- ✓ *Recognize parallelism in computational problems.*
- ✓ *Know different parallel systems and their classification.*
- ✓ *Design parallel algorithms for different applications.*
- ✓ *Compare replication schemes with respect to performance, availability and consistency concerns.*
- ✓ *Design, implement, and debug distributed systems.*
- ✓ *Implement parallel algorithms using MPI and OpenMP environments.*
- ✓ *Element parallel algorithms using MPI and OpenMP environments 3.0 3.5*
- ✓ *AVERAGE.*

- Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming that it does not take any time to assign work to a process, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

INTRODUCTION TO DISTRIBUTED SYSTEMS

Goals of the Distributed Systems, Relation to parallel systems, synchronous versus asynchronous execution, design issues and challenges, Types of Distributed Systems, Distributed System Models, Hardware and software concepts related to distributed systems, middleware models.

Distributed Computing and Communication design principles: A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- 08 Passing, Consensus, Conditional Actions, Critical Paths, Scalability, and cache coherence in multiprocessor systems, synchronization mechanism.

UNIT-2 **8L+8T+0P=16 Hours**

PARALLEL AND DISTRIBUTED PROGRAMMING FRAMEWORKS

Overview of CUDA, Open MP, POSIX Threads, Apache Hadoop (DFS), and current trends in parallel and distributed computing.

PRACTICES:

- Give five types of hardware resource and five types of data or software resource that can usefully be shared. Give examples of their sharing as it occurs in practice in distributed systems.
- The host computers used in peer-to-peer systems are often simply desktop computers in users' offices or homes. What are the implications of this for the availability and security of any shared data objects that they hold and to what extent can any weaknesses be overcome through the use of replication?
- Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed or delivered too fast for the recipient to handle them, but those that are delivered arrive with the correct contents. Describe the classes of failure exhibited by each service. Classify their failures according to their effects on the properties of validity and integrity. Can service B be described as a reliable communication service?
- Illustrate distributed design through a substantial case study, examining in detail the design of the Google infrastructure, a platform and associated middleware that supports both Google search and a set of associated web services and applications including Google Apps.
- Implementation of the parallel algorithms (on a PC-cluster under Linux platform). The programs will be based on POSIX Thread, MPI programming, Hadoop, Apache Spark etc.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the fundamentals of parallel and parallel computing including architectures and paradigms.	Apply	1	1
2	Analyse the various design principles of parallel algorithms.	Analyse	1	2
3	Learn the intricacies of distributed programming.	Under-stand	2	1
4	Develop and execute basic parallel and distributed applications using basic programming models and tools.	create	2	5

TEXT BOOKS:

1. DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition. George Coulouris. Cambridge University. Jean Dollimore formerly of Queen Mary, University of London.
2. Distributed Systems Principles and Paradigms Andrew S. Tanenbaum Maarten Van Steen, 3rd Edition, 2017.

REFERENCE LINKS:

1. Introduction To Parallel Programming, Peter S. Pacheco University of San Francisco.
2. Introduction To Parallel Processing, M.Sasikumar, Dinesh Shikhare and P. Ravi Prakash, Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley.
3. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
4. Parallel Programming in C with MPI and Open MP Michael J.Quinn, McGrawHill Higher Education.
5. https://hpc.llnl.gov/training/tutorials/introduction-parallel-computing-tutorial.
6. https://www.geeksforgeeks.org/introduction-to-parallel-computing/.
7. https://nptel.ac.in/.
8. https://www.coursera.org/.