11B1WCI834: Parallel Processing

Course Credit: 3
Semester: VIII

Introduction

Parallel programming developed as a means of improving performance and efficiency in a parallel program, the processing is broken up into parts, each of which can be executed concurrently. The instructions from each part run simultaneously on different CPUs. These CPUs can exist on a single machine, or they can be CPUs in a set of computers connected via a network. Parallel programming is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process. The objective of the subject is to provide the knowledge and basic applications of parallel processing concepts, parallel environments and architectures, parallel algorithms and parallel programming. The format of the course will be lecture-discussions, assignments. Students are strongly encouraged to participate actively in class discussions.

Course Objectives (Post-conditions)

Knowledge objectives:

At the conclusion of the course, following learning objectives are expected to be achieved:

1. Describe different parallel processing architectures based on relationships between processing elements, instruction sequence, memory and interconnected network
2. Identify algorithms, which require parallelization as part of system design or performance enhancement
3. Classify shared and distributed memory parallel systems according to their properties and usage models
4. Design and develop parallel algorithms for shared and distributed memory models
5. Evaluate the performance of parallel algorithms designed based on shared and distributed memory models as well as against serial based algorithm designs

Application objectives:

The homework portions of the course are intended to help you apply your understanding,

1. Ability to acquire and apply fundamental principles of science and engineering
2. Ability to apply parallel programming constructs to make software execution parallel
3. Ability to think about different phenomenon in nature for their occurrence in parallel.

Expected Student Background (Preconditions)

Students are expected to have a solid grasp of the fundamentals of computer system, including a basic understanding of the operation of the computer, especially CPU. In addition, students are expected to know application development environment and programming concepts. Assembly programming ability will be helpful, as we will be looking to understand architecture of contemporary computers.

Topics Outline:

S NO	Topics	Hrs
1	Parallel computer architecture, principles of scalable performance and multiprocessors architecture.Parallel Computer Models: The state of computing, Multiprocessors and multicomputer, Multi vector and SIMD computers, PRAM and VLSI Models, Architectural Development tracks	6
2	Network Properties ,Conditions of parallelism, program partition and scheduling ,Program Flow Mechanisms, System Interconnect Architecture, Interconnected networks.	6
3	Principles of scalable Performance, performance Metrics and Measures, Parallel processing Applications , Speedup Performances Laws, Scalability and approaches	6
4	Basic concepts of pipelining, Arithmetic pipelines, Instruction pipelines Hazards in a pipeline: structural, data, and control hazards Overview of hazard resolution techniques, Dynamic instruction scheduling Branch prediction techniques, and Instruction-level parallelism using software approaches.	6
5	Main memories, Cache memory design and implementation, Virtual memory design and implementation, Secondary memory technology, RAID	6
6	Centralized vs. distributed shared memory ,Interconnection topologies, Multiprocessor architecture, Multiprocessor system interconnects, cache coherence and synchronization Mechanism, Three generations of Multicomputer, Message Passage Mechanism using OPEN MP and MPI	6
7	Parallel Processing with GPU (CUDA), Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: (Examples: IBM CELL BE, Nvidia Tesla GPU), Memory hierarchy and transaction specific memory design, Thread Organization	6
	Total	42

References

1. Parallel Programming in C with MPI and OpenMP by M J Quinn
2. Advanced Computer Architecture, Parallelism, Scalability, Programmability, Kai Hwang, Mc Graw Hill International Editions, Computer Science Series.
3. Parallel Programming in OpenMP , by Rohit Chandra (Author), Leo Dagum (Author), Dror Maydan
4. Introduction to Parallel Computing by Ananth Grama, George Karypis, Vipin Kumar, and Anshul Gupta.

Evaluation Scheme:

1.	Mid Term Exam (Viva and Written Exam)	20
2.	End term Exam (Viva and Written Exam)	30
3.	Lab Records	5
4.	Regular Assessment (Quality and quantity of experiment performed, Learning laboratory skills, Attendance etc.)	30
5.	Project	15
	Total	**100**