Syllabus for Masters of Technology

Computer Engineering

Subject Code: 01CO0304

Subject Name: Parallel Algorithms

M. Tech. Year - II

Objective: Parallel computing has become mainstream and very affordable today. This is mainly because hardware costs have come down rapidly. Processing voluminous datasets is highly computation intensive. Parallel computing has been fruitfully employed in numerous application domains to process large datasets and handle other time-consuming operations of interest. As a result, unprecedented advances have been made in such areas as biology, scientific computing, modeling and simulations, and so forth. Hence the objective of this course is to introduce parallel algorithms and compare it with its sequential equivalent.

Credits Earned: 4 Credits

Course Outcomes:

At the end of the course, students will be able to:

- Gain basic understanding of fundamental concepts in parallel computing.
- Be able to identify and leverage common parallel computing patterns.
- To know about parallel computing model like PRAM, LMCC etc.
- Analyze the computational complexity of parallel algorithms
- Be able to properly assess efficiency and scalability of a parallel algorithm/application

Teaching and Examination Scheme

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial/ Practical Marks		Total Marks
Theory	Tutorial	Practical		ESE (E)	IA	CSE	Viva (V)	Term work (TW)	
3	0	2	4	50	30	20	25	25	150

Content:

Sr. No.	Topics	Contact Hours
1	Computers Models of computation, Need for parallelism, Complexity measure for parallel algorithms, parallel computational models such as PRAM, LMCC, Hypercube, Cube Connected Cycle, Butterfly, Perfect Shuffle Computers, Tree model, Pyramid model, Fully Connected model, PRAM, CREW, EREW models, simulation of one model from another one, Expressing parallel algorithms	**5**
2	Parallel combinatorial algorithms: permutations with and without repetitions combinations, derangements	**4**
3	Maximum/Minimum, Median, Kth Largest/Smallest element, Matrix Vector Multiplication, Matrix-Matrix Multiplication, Parallel discrete event simulation, Image dithering, Dense LU factorization	**6**
4	Parallel sorting algorithms: Hyper quick sort, Merge sort, Bitonic merge Sort, odd even transposition, Enumeration sort (sorting on the CRCW model, CREW model and EREW model)	**11**
5	Parallel searching algorithms: Searching on a sorted sequence (EREW,CREW,CRCW), Searching on a random sequence (EREW, CREW, CRCW, Tree and Mesh) Sequential selection algorithm, Parallel selection algorithm (EREW parallel solution)	**11**
6	Parallel graph algorithms: parallel graph search &, tree traversal algorithms, Graph coloring, Minimal spanning tree, Shortest path algorithm	**11**
	Total Hours	**48**

Reference Books:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introductionto Parallel Computing, Second Edition, Addison Wesley, 2003. ISBN: 0-201-64865.
2. S. Akl. Design and Analysis of Parallel Algorithms, Prentice Hall Inc, 1992.
3. Michael Quinn, Parallel Computing Theory and Practice, McGraw Hill,Second Edition, 1994.
4. F.T.Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes , MK Publishers, San Mateo California, 1992..

5. Wilkinson, M.Allen, Parallel Programming Techniques and Applications using networked workstations and parallel computers, Prentice Hall, 1999
6. Joseph Jaja. An Introduction to Parallel Algorithms, Addison Wesley, 1992.
7. H. Sparkias and A. Gibbon. Lecture notes on Parallel Computation, Cambridge University Press, 1993.
8. K. Hwang and F. A. Briggs.Computer Architecture and Parallel Processing, McGraw Hill Inc., 1985.

List of Practicals:

1. Compare the speedup of the parallel implementation of Quick sort using MPI (On a cluster of 5 Nodes) and OpenMP (Shared Memory Implementation on multicore machine).
2. Compare the speedup of the parallel implementation of Merge sort using MPI (On a cluster of 5 Nodes) and OpenMP (Shared Memory Implementation on multicore machine).
3. Compare the speedup of the parallel implementation of Bitonic sort using MPI (On a cluster of 5 Nodes) and OpenMP (Shared Memory Implementation on multicore machine).
4. Compare the speedup of the parallel implementation of Odd-Even transposition sort using MPI (On a cluster of 5 Nodes) and OpenMP (Shared Memory Implementation on multicore machine).
5. Implement parallel list ranking algorithm on MPI cluster of 5 Nodes and find it speedup over the sequential implementation.\
6. Give parallel implementation (OpenMP/MPI) of Awerbuch-Shiloach algorithm for finding the connected components of a graph.