Brooklyn College  
Department of Computer & Information Science  

CISC 3330 [47] Foundations of Parallel and Distributed Computing  
3 hours; 3 credits

Survey of parallel and distributed hardware fundamentals, including SMP machines, clusters, grids, and networks of workstations. Introduction to the design of parallel algorithms and the analysis of their efficiency. Survey of software issues for parallel and distributed computation including message passing and shared memory, processes and threads, client/server and peer to peer, and issues of synchronization. Complexity considerations and the limits of parallelism.

Objectives  
Parallel and distributed computing is now ubiquitous in both business and scientific computing. The recent introduction of Multiple Core technology, the cost effective construction of Cluster and Grid supercomputers, and the increasing importance of client/server distributed enterprise level applications all point to a new focus on parallel computing. In particular, multiple cores brings the problem of effective use of parallelism into the computing mainstream as never before. Our students need a firm background in these topics to be prepared for the 21st century job market. This course provides the necessary background for parallel and distributed computing. It surveys the principal architectures, software systems and issues, and provides both the theoretical and practical background required for the design of parallel algorithms and their successful implementation as parallel or distributed systems.

- To provide students with the ability to design and analyze parallel algorithms.
- To provide students with a knowledge of the hardware platforms available for parallel computing and how they impact software design.
- To provide students with knowledge of software tools used for parallel and distributed system design.
- To provide students with a practical knowledge of the synchronization issues involved in parallel and distributed programming.

Syllabus  
Week 1  
Introduction. Course outline and grading.  
Models of Parallel and Distributed computation  
SISD and superscalar machines  
SIMD  
MISD  
MIMD
Week 2  MIMD machines - tightly coupled vs loosely coupled parallelism
message passing vs. shared memory.
interconnection networks
example: routing on the hypercube.
SMP machines, clusters, grids, other networks.

Week 3  Introduction to parallel algorithms
summing an array using message passing and shared memory
finding smallest and largest element in an unsorted array, etc.
Review of algorithm analysis.
Performance issues: efficiency and speedup
Introduction to the PRAM (EREW, CREW, and CRCW) - a
theoretical model of shared memory MIMD.

Week 4  Algorithms for the PRAM
searching and sorting
a) with unlimited processing elements
b) with bounded (less than n) number of processors.
c) sorting in the real world - An example for the Hypercube or
other supercomputer.

Weeks 5-6  Models of parallel programs
a) threads
b) processes and interprocess communication
c) Models of distributed computing - client/server, peer to peer
c) examples from programming systems (including virtual
machines, such as Xen, VMware, and mainframe VM)

Week 7  Exam

Weeks 8-9  More advanced algorithms for the PRAM
a) Graph algorithms - connectivity and shortest path.
b) Algorithms taken from chosen area of interest

Weeks 10-11 Issues of synchronization and Critical sections.
a) semaphores for barrier synchronization.
b) monitors
c) readers/writers and producer/consumer problems

Week 12  Exam

Weeks 13  Parallel and distributed data structures.

Weeks 14  Limits of parallelism--complexity issues
Bibliography


