ADVANCED PARALLEL COMPUTATION
CS 5234

I. CATALOG DESCRIPTION

Survey of leading high-end computing systems and their programming environments. Advanced models of parallel computation. Mapping of parallel algorithms to architectures. Performance programming and tools for performance optimization on parallel systems. Execution environments and system software for large-scale parallel computing. Case studies of parallel applications. Pre: 4234 (3H, 3C). Graduate standing.

Course Number: 5234

ADP Title: Advanced Parallel Computation

II. LEARNING OBJECTIVES

Having successfully completed this course, the student will be able to:

- Describe the design and implementation of high-performance parallel computing systems.
- Use rigorous methods for mapping parallel algorithms to parallel architectures.
- Design, implement, and optimize large-scale parallel programs.
- Develop system software components to support the execution of parallel applications on modern parallel architectures.

III. JUSTIFICATION

High-end parallel computing systems lie at the forefront of scientific discovery. Understanding parallel computing systems and their hardware/software interface from the perspectives of the algorithm designer and the programmer is a valuable skill for graduate students with research interests in computer systems, scientific computation, and bioinformatics. This course builds this important skill by teaching parallel computation at the intersection between architectures, algorithms, and system software. The course complements the undergraduate parallel computation course (CS4234) by exposing students to the organization and programming interfaces of leading supercomputers, detailed models of parallel computation on distributed and shared-memory systems, and advanced development, analysis, and optimization methodologies for parallel programs. The course also strengthens the important skill of performance-oriented programming, which complements the knowledge acquired by students in algorithms and computer organization. The course is appropriately placed at the 5000 level because it requires a maturity and depth of computer science knowledge that is provided by an undergraduate degree in computer science or a related discipline, e.g., computer engineering.

IV. PREREQUISITES

Graduate standing together with the material taught in CS4234 (Parallel Computation), or equivalent material, constitutes the necessary background for this course. In particular, the students should be familiar with two models of parallel systems (distributed memory and shared memory), basic analysis of parallel algorithms, the concepts of speedup, scalability, and efficiency, and at least one parallel programming interface. It is also assumed that students are comfortable with significant programming assignments involving system software development in C and parallel algorithm development, as are required in CS4234.

V. TEXTS AND SPECIAL TEACHING AIDS:

The following textbooks are recommended reading for this course. No single textbook is available that covers this rapidly evolving field. The course will use material from these books, as well as material from journal articles and conference papers that present parallel architectures, models of parallel computation, and case studies.

- Bertsekas, Dimitri and John N. Tsitsiklis. PARALLEL AND DISTRIBUTED COMPUTATION: NUMERICAL METHODS. Nashua NH:Athena Scientific, 1997, 718.
- Culler David, Jaswinder Pal Singh, and Anoop Gupta. PARALLEL COMPUTER ARCHITECTURE: A HARDWARE/SOFTWARE APPROACH. San Francisco CA: Morgan Kaufmann, 1998, 1056.
- Dongarra, Jack, William Gropp, Ken Kennedy, Linda Torczon, and Andy White, eds. THE SOURCEBOOK OF PARALLEL COMPUTING. San Francisco CA: Morgan Kaufmann, 2002, 842.
- Hwang, Kai and Zhiwei Xu. SCALABLE PARALLEL COMPUTING: TECHNOLOGY, ARCHITECTURE, PROGRAMMING. New York NY: McGraw-Hill, 1998, 832.
- Jordan, Harry F. and Ghita Alaghband. FUNDAMENTALS OF PARALLEL PROCESSING. Upper Saddle NJ:Prentice-Hall, 2003, 536.

VI. SYLLABUS:

		Percent of course
1.	Background and overview	5
2.	Supercomputing technologies	20
3.	Analytic models of parallel computation	20
4.	Mapping parallel algorithms to architectures	15
5.	Advanced programming tools	20
6.	Application case studies	10
7.	System software case studies	10
	Total	**100**