School of Computing and Information Science

Course Title: Parallel Computing Systems **Date:** 10/04/2023

Course Number: CEN 5082

Number of Credits: 3

Subject Area: Computer Architecture	**Subject Area Coordinator:** **email:**
Catalog Description: Advanced course on parallel computing. Students will learn the state-of-the-art parallel architectures and programming methods, including heterogeneous computing systems, parallel programming models, and performance modeling and optimization. Students will also gain deep knowledge on challenges and research opportunities in modern large-scale parallel systems.	
Textbook: None. We will distribute reading materials throughout the semester.	
References: 1） Introduction to Parallel Computing (2nd Edition), authored by Ananth Grama, Anshul Guptam, George Karypis, and Vipin Kumar. Addison-Wesley, 2003 2) Programming Massively Parallel Processors: A Hands-on Approach, Second Edition. By David Kirk and Wen-mei Hwu 3) Parallel Computer Architecture: A Hardware/Software Approach. David E. Culler, Jaswinder Pal Singh and Anoop Gupta.	
Prerequisite Courses: COP-3530 and COT 5432; A knowledge of programming in C/C++, the basics of data structures, and the basics of machine architecture; or permission of the instructor	
Corequisite Courses: None	

Type: **Elective**

Audience: graduate student will be exposed to the course materials listed in the course outline. The graduate students will have work assigned to them such as reading additional materials, research papers and executing a small project in the context of the course.

Prerequisite Topics:

1. C/C++ programming, Computer Architecture, and Data Structures

Course Outcomes:

Students who successfully complete this course will be able to:

1. Describe a selection of concepts, algorithms, and models used in parallel computing for solving complex real-world problems.
    - Provide a solid background on the pertinent computer science, mathematical, and electrical engineering concepts that make up the foundations of the discipline of electrical engineering and computer science, as well as their closely associated fields.
    - By the end of this course, students will gain deep knowledge of parallel architecture designs and recent trends of parallel architectures.

- To expand the research of computer science to nontraditional areas by continually seeking to incorporate new methodologies and research findings to our graduate curriculum.

2. Identify a class of parallel computing techniques and algorithms that might be applied to a specific task for a given parallel architecture.
   - By the end of this course, students will have hands-on experience with a series of popular parallel programming models.
   - Students will also be conversant with performance analysis and modeling of parallel programs.
   - Explain the basic understanding of sources of parallelism and locality and how to identify enough parallelism and reduction in data movement costs for given parallel architectures.
   - To provide students with a knowledge to correctly apply the laws of nature to the creative formulation and solution of engineering problems through the use of analytical, computational, and experimental techniques.
3. Identify, understand, and apply the concept of programming "patterns" and how to identify the "7 basic dwarfs" of high-performance computing and how these dwarfs can be parallelized for big data problems in manufacturing, medicine, scientific, and financial domains.
   - By the end of this course, students will be able to explain the major research challenges in the current parallel computing research.
4. Identify and explain the metrics for measuring performance and be able to identify bottlenecks in parallel algorithms and architectures and be able to describe dynamic and static load balancing techniques for a given architecture.
   - The course also emphasizes the importance of performance measurement, and students will become adept at identifying both bottlenecks in parallel algorithms and architectures and understanding appropriate load balancing techniques.

Outline:

Topic	***Hours (Total: 37.5 hours = 15 weeks * 2 lectures/week * 1.25 hrs/lecture)***	**Outcome**
1. Parallel programming models including CUDA, OpenCL, OpenMP, MPI, UPC, Cilk+, TBB, domain specific programming model	**8**	***1,2***
2. Architecture for parallel systems, including vector and SIMD, memory coherence, consistency, and synchronization, FPGA	**8**	***1,2,3***
3. Performance models and optimization, including classic parallel performance models, profiling and analysis tools	**4**	***1,2,3***
4. Sources of parallelism and locality in simulation and Algorithms	**4**	***1,2,3,4***
5. Emerging research topics in large-scale parallel systems, like scientific machine learning, machine learning for system optimization	**6**	***1,2,3,4***