DO NOT TYPE IN THIS BOX

Bulletin # : _____

Academic Year : _____

FLORIDA INTERNATIONAL UNIVERSITY
UNIVERSITY CURRICULUM COMMITTEE
Proposal for a New Course

1. **School/College** Engineering and Computing

   Div./Dept. in Which Taught School of Computing and Information Sciences

2. | Alpha Prefix | 1st Digit | Last 3 Digits | "C"-lec-lab "L"-Lab | Cr. Hrs. |
   |---|---|---|---|---|
   | CIS | 4 | | | 3 |

   CIP Code ***(Leave this blank):*** _____

3. **Grading Method (select one):** ☒ Graded ☐ Pass/Fail

4a. **Course Title** Applied Parallel Computing

b. **Abbreviated course Title** ***(for computer class schedules, transcripts)*** Applied Parallel Comp.

LIMITED TO 25 Characters *(including spaces)*

5. **Statewide Course Numbering Subject Matter Area** CIS (Computing and Information Sciences)

6. **Catalog Description/Major Topics** *(not to exceed 200 characters including spaces)*

College of Medicine and College of Law: Attach description not exceeding 1,000 characters including spaces.

This course teaches advance undergraduate and graduate students to solve problems from scientific, social and financial domains using parallel computing principles and techniques.

7. **Attach detailed syllabus course outline and course justification on separate page(s).**
8. **Prerequisite(s):** (COP-3530 and (CDA-3102 or CDA-4101 or EEL-4709)) or permission of the instructor
9. **Corequisite(s):** None
10. **Objective(s) of Course:**

Provide students with a solid and comprehensive understanding of parallel computing algorithms focusing on their pivotal role in real-world applications.

11. **Does this course duplicate/overlap other courses at FIU?** ☒ No ☐ Yes
    If yes, please explain: _____
12. **What other closely related department(s) have been consulted about this course?**

None

13. **Is this course used for the assessment of a program or a certificate (if yes, then send a notification to assessment@fiu.edu)? No Yes**

PROPOSAL REQUESTED BY:

Faculty Contact Fahad Saeed — 09 / 10 / 20 19
(Type name) **(Signature)**

fsaeed@fiu.edu — 305-348-3131
(Email address) **(Phone number)**

Chairperson (Dept./Div.) S.S. Iyengar — ____ / ____ / 20 19
(Type name) **(Signature)**

Chairperson (Curr. Comm.) Wei-Chiang Lin — ____ / ____ / 20 19
(Type name) **(Signature)**

College/School Dean John Volakis — ____ / ____ / 20 19
(Type name) **(Signature)**

Submit one original form. Attach one copy of the course justification and course syllabus, course description, objectives, major topics and textbooks.

Applied Parallel Computing
Instructor: Fahad Saeed

Course Justification
This course teaches both graduate and advanced undergraduate students from diverse departments how use parallel computers both efficiently and productively. The second goal is increasingly important since essentially all computers are (becoming) parallel, from supercomputers to laptops and all scientific and societal problem require processing of large data sets. So beyond teaching the basics about parallel computer architectures and programming languages, we emphasize commonly used patterns that appear in essentially all programs that need to run fast. These patterns include both common computations (e.g. linear algebra, graph algorithms, structured grids,) and ways to easily compose these into larger programs. We show how to recognize these patterns in a variety of practical problems, efficient (sometimes optimal) algorithms for implementing them, how to find existing efficient implementations of these patterns when available, and how to compose these patterns into larger applications. All of this is done in the context of parallel computing models that are widely used today: shared memory (OPENMP on your multicore), Distributed-Memory (MPI on a supercomputer) and CPU-Accelerator (CUDA and OPENCL for GPU's), finally Cloud-Computing paradigms (MapReduce and Hadoop). Theoretical analysis techniques will also be studied as part of the design and implementation.

Currently there are virtually no courses on parallel computing or high-performance computing at FIU. There is only one course offered by SCIS as Introduction to Parallel Computing which introduces different parallel computing paradigms and students get some experience in parallel models and architectures. However, the proposed new course will focus more on the application of these parallel computing models, and how to recognize the parallelism that needs to be exploited. This includes overarching goal of writing programs easily that execute efficiently on highly parallel computing systems, use "Dwarfs" to design and evaluate parallel programming models and architectures instead of focusing on traditional benchmarks, and maximizing application efficiency by using programming models that support a wide range of data types and successful models of parallelism: task-level parallelism, word-level parallelism, and bit-level parallelism. Further, the students will be exposed to "parallel programming patterns", relatively short list of basic computing problems that appear repeatedly. There are good ways to solve these problems, and so it is most productive to be able to recognize these "patterns" when they appear and use the best available algorithms and software to implement them.

There is a crucial need to a course that provides an application oriented parallel computing paradigm and lays the theoretical and experimental foundations of these parallel algorithms. This upper undergraduate and graduate level course will thoroughly equip students with the tools of parallel and distributed computing from the application perspective and students will be equipped with strong theoretical tools that can be applied to a wide spectrum of applications and domains. The course will also allow students to measuring performance and finding bottlenecks in supercomputers and Load balancing techniques, both dynamic and static will be explored in application specific context.

Several senior positions in industry as well as national labs **require** such solid background of high-performance computing (including NVIDIA, Amazon, Intel) as a complement to data science and machine-learning knowledge. As the required preliminaries and mathematical concepts are covered in the class, it also can benefit non-CS students in the College of Engineering and Computing who are interested in applying parallel computing tools and models in their research for several computing and engineering applications.

School of Computing and Information Science

Course Title: Applied Parallel Computing **Date:** 10/18/2019

Course Number: CIS 4XXX

Number of Credits: 3

Subject Area: Computer Applications	**Subject Area Coordinator:** **email:**
Catalog Description: This course teaches advance undergraduate and graduate students to solve problems from scientific, social and financial domains using parallel computing principles and techniques.	
Textbook: None. We will distribute reading materials throughout the semester.	
References: 1) Petascale Computing: Algorithms and Applications, Edited by David A. Bader, Chapman & Hall/CRC Computational Science Series, 2007 2) Big Data: Algorithms, Analytics, and Applications; ISBN 9781482240559 3) Multicore Computing: Algorithms, Architectures, and Applications (Chapman & Hall/CRC Computer and Information Science Series) 1st Edition 4) Techniques and Environments for Big Data Analysis: Parallel, Cloud, and Grid Computing (Studies in Big Data) 1st ed. 2016 Edition 5) Networking for Big Data (Chapman & Hall/CRC Big Data Series) Hardcover August 3, 2015	
Prerequisite Courses: (COP-3530 and (CDA-3102 or CDA-4101 or EEL-4709)) or permission of the instructor	
Corequisite Courses: None	

Type: **Elective (Systems group)**

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

Prerequisite Topics:

- Calculus, Basic Programming, 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.
2. Identify a class of parallel computing techniques and algorithms that might be applied to a specific task for a given parallel architecture
3. 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

4. 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
5. 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.

Outline:

Topic	***Hours (Total: 37.5 hours = 15 weeks * 2 lectures/week * 1.25 hrs/lecture)***	**Outcome**
1. Introduction: Why Parallel Computing?	**1**	***1,2***
2. Single processor machines: Memory hierarchies and processor features	**1**	***1,2,3***
3. Introduction to parallel machines and programming models	**1**	***1,2,3***
4. Sources of parallelism and locality in simulation and Algorithms: part 1	**1**	***1,2,3,4***
5. Sources of parallelism and locality in simulation and Algorithms: part 2	**1**	***1,2,3,4***
6. Shared memory machines and programming: OpenMP and Threads	**1**	***1,2,3,4,5***
7. Distributed memory machines and programming in MPI	**2**	***1,2,3,4,5***
8. Partitioned Global Address Space Programming with Unified Parallel C UPC	**1**	***1,2,3,4,5***
9. GPUs, and programming with CUDA and OpenCL	**1**	***1,2,3,4,5***
10. Performance and Debugging Tools	**1**	***3,4,5***
11. Dense Linear Algebra Algorithms: Part 1	**1**	***1,2,3,4,5***
12. Dense Linear Algebra: Part 2	**1**	***1,2,3,4,5***
13. Graph Partitioning: Part 1	**1**	***1,4,5,***
14. Graph Partitioning: Part 2	**1**	***1,4,5***
15. Sparse-Matrix-Vector-Multiply: Part 1 and 2	**2**	***1,3,4,5***
16. Particle (N-Body) methods: Efficient Data Race Detection for Distributed Memory Parallel Programs	**1**	***1,2,3,4,5***
17. Structured grids and multigrid	**1**	***4,5***
18. Cloud computing paradigms with MapReduce and Hadoop	**1**	***3,4,5***
19. Patterns of Parallel Programming	**2**	***4,5***
20. Structured grids	**1**	***4,5***
21. Exascale Computing	**1**	***1,2,3,4,5***
22. Parallel Graph Algorithms	**1**	***1,2,3,4,5***

23. Parallel Climate Modeling	**1**	***1,2,3,4,5***
24. Parallel Fast Fourier Transform (FFT)	**1**	***1,2,3,4,5***
25. Static and Dynamic Load Balancing	**1**	***1,2,3,4,5***
26. Parallel Algorithms and Architectures for Machine-Learning and Deep-Learning Solutions	**2**	***1,2,3,4,5***