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.