ECE 6397: Parallel Algorithms for GPUs and Heterogeneous Systems

General Information
David Mayerich, Assistant Professor  Room: AH 204
Office: Engineering Building 2, W330  Time: 2:30pm – 4:00pm, MW
Email: mayerich@uh.edu  Office: 9:00am – 10:00am, MTW

Description
Limitations in single-threaded processors are forcing new paradigms in software and algorithm development in order to process ever-increasing data sizes. Research and industry applications often require massively parallel systems for simulation, data processing, and data analysis. Several architectures, including nVidia’s CUDA and Intel’s Xeon Phi, provide highly parallel performance at low cost. However, algorithms optimized for massively parallel systems require new design and programming strategies. In this course, we will focus on the design and development of algorithms that take advantage of highly parallel co-processors, such as the nVidia GPU and Xeon Phi, in order to solve research related problems. This course will include an overview of data parallel architectures and principles in programming massively parallel systems. Topics covered will focus on three primary topics:

1. Understanding data-parallel hardware in order to develop efficient algorithms
2. Designing numerical methods optimized for data parallel architectures
3. Leveraging data-parallel hardware to process large data sets common in modern Big Data applications

Textbook
The hardware and languages surrounding GPU programming are rapidly changing, however I will recommend the following, which provides a more general look at heterogeneous and parallel programming:

"Programming Massively Parallel Processors: A Hands-on Approach" by David Kirk (nVidia) and Wen-mei Hwu (UIUC)

Course Topics
1. Parallel Programming
   a. GPU Architecture
   b. SIMD/SPMD Paradigms
2. CUDA Programming Language
   a. Program structure
   b. Global memory and data transfers
   c. Threads and kernel functions
3. Profiling C/C++ and CUDA Code
4. Optimization Strategies
   a. Managing memory bandwidth
   b. Numerical representation
   c. Thread divergence
   d. Algorithm design
5. GPU Libraries
   a. cuFFT, cuBLAS
   b. Thrust
   c. GPU Support in MATLAB and Python
6. Applications
   a. Optimization
   b. Numerical Methods
   c. Big Data streaming

Grading
5% Homework Assignments  Grade Point Scale:
25% Programming Assignments  A = 93-100  A- = 90-92
20% Exam 1  B+ = 87-89  B = 83-86  B- = 80-82
20% Exam 2  C+ = 77-79  C = 73-76  C- = 70-72
30% Final Project  D+ = 67-69  D = 63-66  D- = 60-62  F < 60
Homework
There will be regular homework assignments, which will be collected, checked for completion, and randomly graded. Failure to do the work will result in your assignments being selected for more frequent grading. These homework assignments will be based on material from the book or your own notes. Students are encouraged to work together on the homework assignments, however I expect you to turn in actual (not copied) work. **Answers to homework assignments must be legible (typing is preferred) and include your name and the date it was assigned in the upper-left corner. Some assignments will require drawing figures, which should be done neatly by hand. No late homework will be accepted,** because we will generally discuss the assignment in class on the due date. You may hand it in before the due date if you expect to be absent.

Programming Assignments
Programming projects must be written in the language specified (C/C++ and CUDA) and submitted in a single plain-text file via Blackboard. The naming convention will be specified in a handout for each assignment. C/C++ code will be compiled on a CUDA cluster is available to ECE students. Please arrange for an account so that you can verify your code. Visual Studio is also available for free through Microsoft’s website and can be used to create code on systems with an nVidia GPU. **Please note the following when submitting code:**

1) **If you develop on Visual Studio, test your code on Unix. This is a good way to find bugs you wouldn’t otherwise and I will be running your code on the cluster.**

2) **Use the input/output methods described in the assignment. Don’t add other stuff that will make your program difficult to execute (pauses, user input, etc). All input will be through command-line arguments.**

3) **If your code doesn’t work, have your program output a description of what you’ve completed and why it doesn’t work. Please don’t make me waste time trying to get code to compile if it didn’t work when you handed it in. If you tell me clearly what you’ve done, I can more easily grade what you have.**

Final Projects
Each student will select a final project after the first Mid-term Exam. The instructor will provide several suggested projects before the deadline, however students can propose their own. Projects proposed by students can be used as part of their ongoing research. Projects must focus primarily on GPU computing, relying on a significant amount of parallelism and requiring a significant amount of CUDA code. Students will be expected to give (1) a written report and (2) final 10-minute presentation describing their research results, profiling their code, and discussing speedup and implementation. **The code and written report is due the last day of class. Presentations will be given during the last week of class and overflow into the Final Exam date. Students who present during the last week of class will not be required to attend the final presentation date.**

Examinations
**Exam Schedule:** Mid-term 1: Mar. 6 | Mid-term 2: Apr. 24 | Final Exam: Presentations, May 8 2pm – 5pm

Exams will be based on homework assignments, class notes, and my own notes. My slides and any relevant notes will be made available before the exam, providing at least one week for review.

Exams and quizzes are closed book, closed notes, unless otherwise announced. All work must be done on the examination forms provided for that purpose. The seats for exams will be randomly assigned. All of these regulations are designed to reduce the possibility of cheating, so that all students will be graded as fairly as possible. **No makeup examinations will be given.** If you have a medical emergency you should call your instructor as soon as possible, preferably before the examination. Medical documentation will be required in all such cases.
Academic Honesty Policy
Students in this course are expected to follow the Academic Honesty Policy of the University of Houston. It is your responsibility to know and follow this policy.

Religious Holy Days
Students whose religious beliefs prohibit class attendance on designated dates or attendance at scheduled exams may request an excused absence. To do this, you are strongly encouraged to request the excused absence, in writing, by Wednesday, February 11, 2015. Please submit this written request to your instructor to allow the instructor to make appropriate arrangements. For more information, see the Student Handbook.

Students with Disabilities
Students with recognized disabilities will be provided reasonable accommodations, appropriate to the course, upon documentation of the disability with a Student Accommodation Form from the Center for Students with Disabilities. To receive these accommodations, you must request the specific accommodations, by submitting them to the instructor in writing, by Wednesday, February 11, 2015. Students who fail to submit a written request will not be considered for accommodations. For more information, see the Student Handbook.