ECE/ME/EMA/CS 759 – Fall 2013
High Performance Computing for Engineering Applications

Time: 8:00 – 9:15 Monday-Wednesday-Friday
Location: 2121ME

Instructor: Dan Negrut
Office: 2035ME
Phone: 608 890 0914
E-Mail: negrut@engr.wisc.edu
Course Page: http://sbel.wisc.edu/Courses/ME964/2013
Grades Page: http://learnuw.wisc.edu
Forum: http://sbel.wisc.edu/Forum/viewforum.php?f=15
Grader: Ang Li (ali28@wisc.edu)

Office Hours:
Monday, 2 – 3:30 PM
Friday, 12:30 – 2 PM
Other times by appointment (please call or email to arrange)

Prerequisites: C Programming

Recommended Texts (see also lecture handout for other recommendations):
R. Bryant and D. O'Hallaron, *Computer Systems: A Programmer's Perspective*, Prentice Hall, 2nd Edition, 2011
D. Negrut, *Primer: Elements of Processor Architecture. The Hardware/Software Interplay*, available on the course website.
Jason Sanders and Edward Kandrot: *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional, 2010
GPU Programming Guide, version 5.0
David B. Kirk and Wen-mei W. Hwu: *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, 2010
H. Nguyen (ed.), *GPU Gems 3*, Addison Wesley, 2007
Peter Pacheco: *An Introduction to Parallel Programming*, Morgan Kaufmann, 2011
T. Mattson, et al.: *Patterns for Parallel Programming*, Addison Wesley, 2005
Michael J. Quinn: *Parallel Programming in C with MPI and OpenMP*, McGraw Hill, 2003

Course Objectives: The course will (a) provide an overview of existing High-Performance Computing (HPC) software and hardware, (b) present basic software design patterns for high performance parallel computing, (c) introduce CUDA for parallel computing on the Graphics Processing Unit (GPU), (d) introduce the OpenMP solution to enabling parallelism across multiple CPU cores, and (e) introduce the Message Passing Interface (MPI) standard and its MVAPICH2 implementation for leveraging parallelism on a CPU cluster. The approach is hands-on, the students are expected to use the lecture information, a series of assignments and a final project to emerge at the end of the class with parallel programing knowledge that can be immediately applied to their research projects.

Hardware Used: The course is designed to leverage Euler, a cluster with about 1,200 CPU cores and 56 GPU cards. Each student will receive an individual account on Euler that will be used for GPU computing, MPI-enabled parallel computing and possibly OpenMP multi-core computing. Euler runs Scientific Linux.

Course Workload: The course will have one midterm exam, one midterm project, and one final project. All projects will be either individual or two-member team projects.

Course Structure: For this class, the Microsoft PowerPoint notes used in class will be posted online at http://sbel.wisc.edu/Courses/ME964/2013. Grades in ECE/ME/EMA/CS759 will be based on your performance on homework, midterm exam, one midterm project, one final project, and course participation. All homework and exam scores will be maintained on the Learn@UW course website. This will allow you to monitor your performance and see aggregate scores for the rest of the class, which can give you a continuous idea of your performance in relation to the rest of the class. Should you have questions about your score, please contact the instructor. Policies regarding grading and turning in your homework:

1. *Score-related questions about assignments, midterm exam, and midterm project must be raised prior to the next class period after receiving the score.*
2. *If homework that you turned in appears not to be graded (missing) on the Learn@UW course website please point that out to me within one week after the return of the corresponding set of graded assignments. It is a good practice to save your homework so that I will be able to update the grade to give you full credit for your work.*
3. *The two assignments with the lowest scores will be dropped when computing the final homework average*

Final Grade Policy: The final grade will be computed using the following weights:

Homework	=	**40%**
Midterm Exam	=	**15%**
Midterm Project	=	**15%**
Final Project	=	**25%**
Course Participation	=	**5%**

Homework: Assignments will be handed out roughly on a weekly basis for the first part of the class. The homework solution will be returned using the Learn@UW drop-box system. Homework solutions should be *neat, well organized, and well commented*. Your score for each assignment will be between 0-100. No late homework will be accepted. The homework with the two lowest scores will be dropped when computing final score. You can use this mechanism to basically not turn in two assignments.

Turning in your assignment should boil down to uploading at Learn@UW a zipped (tar.gz) file. Upon unzipping, in the top directory, which should be called HW07 (for the seventh assignment, for instance), one should be able to find a **`hw07ReadMe.pdf`** file with your text, a `makefile` that when run (by typing "`make homework`") will generate an executable called **`hw07.exe`**. The grader will run this executable and expect that all results you report in `hw07ReadMe.pdf` will be output in a file or to the screen.

Midterm Exam: There will be one "closed-books" midterm exam that will cover the entire material discussed in the course. You can bring along annotated copies of the documents that you have been asked to read (reading assignments). There will be no need for a computer for this test. The best way to prepare for exams is to participate in class, learn the fundamental concepts, and work on the assignments diligently. The exam will be scored on a scale of 0 to 100. The midterm is scheduled for November 8. A review session is planned for the evening of November 7.

Midterm Project: You will have to select by 11:59 PM on October 23 the topic of your Midterm Project. The Midterm Project software and technical report is due on November 15, at 11:59 PM. All students are expected to work on the project individually or in two-person teams. The topic of this project will be the solution of either dense banded or fully banded linear systems in CUDA. You will have the option to choose one or the other.
Your project should include your CUDA code, profiling results, weak scaling analysis, and a comparison with either a sequential or a parallel state-of-the-art-solution available commercially or as open source.

Final Project: The projects will be individual or generated by two-student teams. The topic of the project could be related to your research. Alternatively, you can continue to work on and augment your Midterm Project by moving from a dense solver to a sparse solver in CUDA, OpenMP, or MPI. Either way, a two-page or shorter PDF doc describing your final project is due on November 15 at 11:59 PM. Once I get a chance to read your final project proposals I might provide feedback and adjust the topic you suggested.
Each student or two-student team will make a presentation, about 30 minutes long, regarding his/her final project during the finals week. I will come up with additional time slots during the finals' week for students/teams to report on their work. For all students the final project report is due at 11:59 PM on the date the Office of Registrar lists the ECE/ME/EMA/CS759 final exam as taking place.

Course Participation: You are expected to participate actively in class discussions and to pose questions. Beyond this and in order to earn the 5% assigned to this category you will have to post at least five answers by the end of the semesters to the questions posted on the ECE/ME/EMA/CS759 forum (http://sbel.wisc.edu/Forum/viewforum.php?f=15).

Disability requests: I must hear from anyone who has a disability that may require some modification of seating, testing or other class requirements so that appropriate arrangements may be made. Please see me after class or during my office hours.

Complaints: If you have a complaint regarding the course and if you are unsatisfied with the response of the instructor, then you should contact the Chair of the Department of Mechanical Engineering. The Chair's office is in 3107ME, and an appointment to see the Chair can be made by contacting the Department Office at 608 263-5372.

Letter Grades: Final letter grades will be based on the total score accumulated on homework and exams throughout the semester using the following scale:

Score	Grade
≥92	A
86-91	AB
78-85	B
70-77	BC
60-69	C
50-59	D

ECE/ME/EMA/CS 759 High Performance Computing for Engineering Applications

Fall 2013– Syllabus [subject to change]

Date	Title	Lecturer	HW Assigned	Reading Assignments Other Observations
09/04	Syllabus related issues. Brief course overview	Negrut		
09/06	Overview of C language	Negrut		Read Chapter 5 of Brian W. Kernighan and Dennis M. Ritchie "The C Programming Language" book
09/09	Overview of C language. Accessing & Using Euler.	Negrut/Seidl	HW1 (due 09/16) C programming-related	
09/11	Overview of Relevant Hardware Issues	Negrut		Additional information available in Negrut's Primer, see link on class website
09/13	Pipelining. Measuring Computer Performance. Memory Aspects.	Negrut		Read the articles of Manferdelli and Amdhal's (link on class website)
09/16	Caches. Virtual Memory.	Negrut	HW2 (due 09/23) C programming-related	Read 2005 article of Dongarra et al. for an overview of HPC (link on class website)
09/18	High Performance Computing: Why, and why now?	Negrut		
09/20	Big Iron HPC.	Negrut/Seidl		
09/23	GPU Computing Intro. The CUDA Programming Model.	Negrut	HW3 (due 09/30) HW: getting started with CUDA, understanding thread/block index issues.	Read article on GPU computing evolution of Nickolls & Dally (link on class website)
09/25	CUDA Execution Configuration. Thread Indexing.	Negrut		
09/27	CUDA Execution Model. CUDA API	Negrut		
09/30	GPU Memory Spaces	Negrut	HW4 (due 10/07) CUDA matrix-vector multiplication & matrix addition	
10/02	GPU Scheduling Issues	Negrut		
10/04	CUDA Scheduling Issues. Global Memory Access Patterns and Implications. Control Flow in CUDA	Negrut		Read 2009 paper (or tech report) of Bell and Garland on Sparse Linear Algebra on the GPU (link on class website).
10/07	CUDA Shared Memory Access. Bank conflicts. Synchronization in CUDA.	Negrut	HW5 (due 10/14) CUDA tiled matrix multiplication Simple vector reduction Kernel Call Overhead	Read Chapters 4 & 5 of the CUDA Programming Guide 5.0 (link on class website)
10/09	Atomic operations in CUDA. Using the CUDA profiler. Example: 1D Stencil Operation	Negrut		Read 2008 article of Volkov and Demmel on GPU benchmarking (link on class website)
10/11	Tiling as a Programing Pattern in CUDA Example: Vector Reduction in CUDA	Negrut		Browse the Appendices of the CUDA Programming Guide 5.0 (link on class website) Read Chapters 1 through 3 of CUDA Programming Guide 5.0 (link on class website)
10/14	CUDA Optimization Issues.	Negrut	HW6 (due 10/21)	Read 2011 GPU Gems 4 paper

	Resource Utilization Issues. Parallel Prefix Scan on the GPU. Using Multiple Streams in CUDA.		CUDA matrix 2D convolution	of Nathan Bell and Jared Hoberock (link on class website)
10/16	Streams, and overlapping data copy with execution.	Negrut		Read 2010 GTC talk of Volkov (links on class website)
10/18	Wrap up, streams. GPU Computing with `thrust`	Negrut		Read “CUDA C Best Practices Guide”.
10/21	Wrap up, GPU computing.	Negrut	HW7 (due 10/28) CUDA parallel reduce operation & cudaMemcpy overhead	
10/23	Parallel Programming with OpenMP: Work Sharing (Sections, Tasks)	Negrut	Decision regarding your Midterm Project topic due with Dan by 11:59 PM.	Read 2010 paper written by Intel on debunking GPU performance. Read paper of Vuduc et al. on same topic (link on class website)
10/25	Parallel Programming with OpenMP: Variable Scoping	Negrut		
10/28	MPI Parallel Programming General Introduction, Point-to-Point Communication	Negrut	HW8 (due 11/04) HW: Parallel scan operation in CUDA.	
10/30	MPI Parallel Programming Point-to-Point communication: Blocking vs. Non-blocking sends	Negrut		
11/01	MPI Parallel Programming Collective Communication	Negrut		
11/04	MPI Parallel Programming: Collective Communication wrap up MPI Derived Datatypes	Negrut	HW9 (due 11/11): Related to `thrust`.	
11/06	MPI Derived Datatypes, wrap-up	Negrut		
11/08	Parallel Programming – Algorithm Design Decisions	Negrut	A review session will be offered the evening of November 7. Room and time TBA. For the exam you can bring along the annotated hardcopies of the papers/documents that you had to read. Otherwise, it is a closed-book exam.	
11/11	NO CLASS		HW10 (due 11/18):Related to OpenMP	
11/13	NO CLASS			
11/15	NO CLASS		**Midterm project due at 11:59 PM**. **Two page Final Project <u>Proposal</u> due at 11:59 PM**	
11/18	NO CLASS		HW11 (due 11/25):Related to MPI	
11/20	NO CLASS			
11/22	NO CLASS			
11/25	Review for Exam	Negrut	Exam is at 7:15 PM. Room TBA.	
11/27	NO CLASS			
	THANKSGIVING HOLIDAY			
12/02	NO CLASS			
12/04	NO CLASS			
12/06	NO CLASS			
12/09	NO CLASS			
12/11	NO CLASS			
12/13	NO CLASS			

This class has no final exam. Final Project Presentations time slots available during finals week. Scheduled through doodle.

Final Project due on Dec. 15 at 11:59 PM (submitted through Learn@UW)