ME 964 – Spring 2012
High Performance Computing for Engineering Applications

Time: 9:30 – 10:45 Tu&Th
Location: 1163ME

Instructor: Dan Negrut
Office: 2035ME
Phone: 608 890 0914
E-Mail: negrut@engr.wisc.edu
Course Page: http://sbel.wisc.edu/Courses/ME964/2012/index.htm
Grades Page: http://learnuw.wisc.edu
Video Streaming:
http://mediasite.engr.wisc.edu/Mediasite/Catalog/pages/catalog.aspx?catalogId=31c0b7c4-3a0f-410b-bacf-0c238380112f&folderId=96ee9eab-32a4-4321-8b45-6eae85c267ef&rootDynamicFolderId=e5b4a945-c68f-45b2-9eb7-b2512f5122cd
Forum: http://sbel.wisc.edu/Forum/viewforum.php?f=3
T.A.: Toby Heyn (heyn@wisc.edu)

Office Hours:
Monday, 2 – 4 PM
Wednesday, 2 – 4 PM
Other times by appointment (please call or email to arrange)

Prerequisites: C Programming

Recommended Texts (see also lecture handout for other recommendations):

Jason Sanders and Edward Kandrot: *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional, 2010 (on reserve at Wendt Library)
GPU Programming Guide: *http://developer.download.nvidia.com/compute/cuda/4_0/toolkit/docs/CUDA_C_Programming_Guide.pdf*, download from NVIDIA website,
David B. Kirk and Wen-mei W. Hwu: *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, 2010 (on reserve at Wendt Library)
H. Nguyen (ed.), *GPU Gems 3*, Addison Wesley, 2007 (on reserve at Wendt Library)
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 is meant to (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 Message Passing Interface (MPI) standard and its MPICH2 implementation for leveraging parallelism on a CPU cluster, and (e) introduce the OpenMP solution to enabling parallelism across multiple CPU cores. 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 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/ME964/2012/index.htm. Grades in ME964 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 instruction. Policies regarding grading and turning in your homework:

1. *Score-related questions about homeworks, 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 homeworks. 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 homework with the lowest score 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	=	**10%**
Midterm Project	=	**20%**
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 Lear@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 lowest score will be dropped when computing final score.

Midterm Exam: There will be one "closed-books" midterm exam that will cover material introduced in the first part of 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 1 to 100. Note that there will be no final exam.

Midterm Project: You will have to select by March 8 the topic of your Midterm Project. The Midterm Project software and technical report is due on April 12 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. Either way, a two-page or shorter PDF doc describing your final project is due on 04/10 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 on May 16, at 11:59 PM. The final exam for this course is scheduled for May 17, 2:45 PM.

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 ME964 forum (http://sbel.wisc.edu/Forum/index.php?board=3.0).

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 3065ME, 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

ME 964 High Performance Computing for Engineering Applications

Spring 2012 – **Tentative** Syllabus

Date	Title	Lecturer	HW Assigned	Observations
01/24	Syllabus related issues ME964 course overview	Negrut		
01/26	Overview of C language	Negrut	HW1 (due 02/02) Read Chapter 5 of Brian W. Kernighan and Dennis M. Ritchie "The C Programming Language" book	C programming-related
01/31	Overview of C language. Accessing & Using Euler. Source version control issues	Negrut/Seidl		Read Eclipse presentation
02/02	Issues with Sequential Computing. The memory, ILP, and power walls	Negrut	HW2 (due 02/09) Read the articles of Manferdelli and Amdhal's (link on class website)	C programming-related
02/07	High Performance Computing: Why, and why now?	Negrut		
02/09	GPU Computing and CUDA Intro	Negrut	HW3 (due 02/16) Read 2005 article of Dongarra et al., overview of HPC (link on class website)	HW: getting started with CUDA, understanding thread/block index issues.
02/14	CUDA Execution Model. Using CMake to build an executable. Debugging and Profiling on Euler under Eclipse	Negrut/Seidl		
02/16	CUDA Execution Model. CUDA API	Negrut	HW4 (due 02/23) Read article on GPU computing evolution of Nickolls & Dally (link on class website)	HW: CUDA matrix-vector multiplication & matrix addition
02/21	GPU Memory Spaces	Negrut		
02/23	GPU Scheduling Issues	Negrut	HW5 (due 03/01) Read 2008 article of Volkov and Demmel on GPU benchmarking (link on class website)	HW: CUDA tiled matrix multiplication Simple vector reduction Kernel Call Overhead
02/28	Using the CUDA debugger and profiler	Negrut		
03/01	Using the CUDA profiler	Negrut	HW6 (due 03/08) Read 2009 paper (or tech report) of Bell and Garland on Sparse Linear Algebra on the GPU (link on class website).	HW: CUDA matrix 2D convolution
03/06	CUDA Scheduling Issues. Global Memory Access Patterns and Implications. Control Flow in CUDA	Negrut		
03/08	CUDA Shared Memory Access. Bank conflicts. Synchronization in CUDA. Atomic operations in CUDA.	Negrut	HW7 (due 03/15) Read Chapters 4 & 5 of the CUDA Programming Guide 4.0 (link on class website)	HW: CUDA parallel reduce operation & cudaMemcpy overhead. **Midterm project description due in Mercurial by 11:59 PM.** **Due date: 04/12**
03/13	CUDA Optimization Issues. Example: Vector Reduction in CUDA	Negrut		
03/15	Resource Utilization Issues. Parallel Prefix Scan on the GPU.	Negrut	HW8 (due 03/22) Browse the Appendices of the CUDA Programming Guide 4.1 (link on class	HW: Parallel scan operation in CUDA.

			website) Read Chapters 1 through 3 of CUDA Programming Guide 4.1 (link on class website)	
03/20	Multiple Streams in CUDA. Using multiple GPUs.	Negrut	Read 2011 GPU Gems 4 paper of Nathan Bell and Jared Hoberock (link on class website)	
03/22	Streams, and overlapping data copy with execution. GPU Computing with `thrust`	Negrut	HW9 (due 03/29) Read "CUDA C Best Practices Guide". Read 2010 GTC talk of Volkov (links on class website)	HW9: related to `thrust`.
03/27	GPU Computing with `thrust` wrap-up. Quick overview, the Fermi architecture. The CUDA Ecosystem.	Negrut		
03/29	MPI Parallel Programming General Introduction, Point-to-Point Communication	Negrut	HW10 (due 04/12) Read 2010 paper written by Intel on debunking GPU performance. Read paper of Vuduc et al. on same topic (link on class website)	HW10: Related to MPI A two page intermediate report for the Midterm Project is due in **Mercurial** today at 11:59 PM
04/03	**SPRING**	**BREAK**	**NO**	**CLASS**
04/05	**SPRING**	**BREAK**	**NO**	**CLASS**
04/10	MPI Parallel Programming Point-to-Point communication: Blocking vs. Non-blocking sends	Negrut	**Two page Final Project Proposal due today at 11:59 PM**. **Due Date for Final Project: May 16, 11:59 PM**. Use Mercurial to provide the doc.	
04/12	MPI Parallel Programming Collective Communication	Negrut	**Midterm project due today at 11:59 PM**. Use **`Mercurial`**. HW11 (due 04/22) – related to MPI	
04/17	**MIDTERM Exam**	You can bring along the annotated hardcopies of the papers/documents that you had to read. Otherwise, it is a closed-book exam.		
04/19	MPI Parallel Programming: Collective Communication wrap up MPI Derived Datatypes	Negrut	HW12 (due 04/29)	HW related to OpenMP
04/24	MPI Derived Datatypes, wrap-up Parallel Programming with OpenMP – general concepts	Negrut	Final Project related work	
04/26	Parallel Programming with OpenMP: Work Sharing (Sections, Tasks)	Negrut	Final Project related work	
05/01	Parallel Programming with OpenMP: Variable Scoping	Negrut	Final Project related work	
05/03	Parallel Programming – Algorithm Design Decisions	Negrut	Final Project related work	
05/08	CUDA in Medical Physics Visualization	Brian Davis	Final Project related work	
05/10	CUDA and OpenMP in Computational Dynamics	D. Melanz A. Seidl H. Mazhar	Final Project related work	

Final Exam: Thursday, May 17 – 2:45 PM. The two hours set aside for the final exam used for 30 mins. Final Project Presentations. Presentations can be scheduled here: http://www.doodle.com/3b65mca99d6wuety.