ME 964 – Spring 2011
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/ME964/2011/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.: Arman Pazouki and Toby Heyn
Grader: Arman Pazouki

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:
- 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/3_2_prod/toolkit/docs/CUDA_C_Programming_Guide.pdf*, download from NVIDIA website, November 2010
- 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: *Parallel Programming with MPI*, Morgan Kaufmann, 1996 (on reserve at Wendt Library)
- 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 Newton, a cluster with 48 CPU cores and 24 GPU cards. Each student will receive an individual account on Newton that will be used for GPU computing, MPI-enabled parallel computing and possibly OpenMP multi-core computing. Newton runs Windows HPC Server 2008 R2.

Course Workload: The course will have one midterm exam, one midterm project, and one final project. The last three to four weeks of the class will be entirely dedicated to work related to the final project. The students will have the option to work on actual code they use in their research and use their ME964 accounts to speed up critical components of the code. All projects will be individual projects.

Course Structure: For this class, the Microsoft PowerPoint notes used in class will be posted online at http://sbel.wisc.edu/Courses/ME964/2011/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 is expected to be emailed to me964uw@gmail.com no later than the midnight of the day when it is due. 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.

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 1 one topic (out of four possible Midterm Project topics) that you want to work on. The Midterm Project software and technical report is due on April 13 at 11:59 PM. All students are expected to work on the project individually. Projects will be uploaded on the course website. Note that a progress report on your Midterm Project is due on 03/29. It should run about two pages long and contain an overview of the algorithm that you plan to implement. Specifically, it should include a flow diagram, data structures that you plan to use, discuss how your algorithm maps upon the underlying SIMD architecture, possible limiting factors that work against your solution implementation (for instance, if all threads executing a kernel need to synchronize, or to perform atomic operations, etc.). Please indicate the use of any third party CUDA libraries such as thrust, for instance. The use of existing libraries is encouraged as long as they don't completely solve your problem.

Final Project: There will be one final project. The projects will be individual and the students will choose a project topic after consulting with the instructor. Ideally, the topic of the project would be related to your research. Alternatively, you can continue to work on and augment your Midterm Project. A two-page or shorter PDF doc describing your final project is due on 04/06 at 11:59 PM. You will make a 5 minute presentation in front of the class describing the final project on 04/07 or 04/12.
No homework will be assigned at the end of the semester to allow you to concentrate on your final project. Each student will make a presentation, about ten minutes long, regarding his/her final project at the time/date of the final exam (May 10, 12:25 PM). I will probably come up with a couple of additional time slots during the finals' week for students to report on their work. For all students the final project report is due on May 9, at 11:59 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 3650ME, 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 2011 – **Tentative** Syllabus

Date	Title	Lecturer	HW Assigned	Observations
01/18	Syllabus related issues ME964 course overview	Negrut		
01/20	Overview of C language	Negrut		
01/25	Issues with Sequential Computing. The memory, ILP, and power walls	Negrut	HW1[1] (due 02/01) Read Chapter 5 of Brian W. Kernighan and Dennis M. Ritchie "The C Programming Language" book	HW: C programming-related
01/27	High Performance Computing: Why, and why now?	Negrut		
02/01	GPU Computing and CUDA Intro Building CUDA apps under Visual Studio 2008	Negrut/Seidl	HW2[2] (due 02/10) Read the articles of Manferdelli and Amdhal's (link on class website)	HW: getting started with CUDA, understanding thread/block index issues.
02/03	Accessing Newton CUDA Execution Model and Memory Spaces	Negrut/Seidl		
02/08	CUDA API	Negrut	HW3[1] (due 02/17) Read 2005 article of Dongarra et al., overview of HPC (link on class website)	HW: CUDA matrix addition
02/10	CUDA API wrap-up GPU Memory Spaces	Negrut		
02/15	GPU Compute Core & Scheduling Issues	Negrut	HW4[2] (due 02/22) Read article on GPU computing evolution of Nickolls & Dally (link on class website)	HW: CUDA tiled matrix multiplication Simple vector reduction Kernel Call Overhead
02/17	CUDA Execution Scheduling Issues and Efficient Memory Transactions	Negrut		
02/22	CUDA Shared Memory Access. Bank conflicts	Negrut	HW5[2] (due 03/01) Read 2008 article of Volkov and Demmel on GPU benchmarking (link on class website)	HW: CUDA matrix 2D convolution
02/24	Overview Midterm Project: **Option 1**: Discrete Element Method on the GPU **Option 2**: Collision Detection on the GPU	Heyn (1) and Pazouki (2)		Students choose a Midterm Project[3] (out of four options). Project due 04/13 at 11:59 PM. Intermediate progress report due on 03/29.
03/01	Overview Midterm Project: **Option 3**: Finite Element Analysis on the GPU	Suresh (3) and Negrut (4)		Students choose a Midterm Project (out of four options). Project due 04/13 at 11:59 PM.

[1] Graded by Toby Heyn
[2] Graded by Arman Pazouki
[3] Students expected to work individually on the Midterm Project. Use Learn@UW drop-box to provide your project description by 03/03 at 11:59 PM.

	Option 4: Sparse direct solver for linear systems on the GPU (Cholesky)			Intermediate progress report due on 03/29 at 11:59 PM.
03/03	Control Flow in CUDA. Resource Utilization Issues. CUDA Optimization Rules of Thumb	Negrut	HW6[2] (due 03/22) Read 2010 paper written by Intel on debunking GPU performance. Read paper of Vuduc et al. on same topic (link on class website)	HW: CUDA parallel Scan Please drop your midterm project description in Learn@UW by 11:59 PM
03/08	Wrap up, CUDA Optimization Discussion. Parallel Prefix Scan on the GPU	Negrut		
03/10	Multiple Streams in CUDA. Using multiple GPUs.	Negrut	Read 2009 paper (or tech report) of Bell and Garland on Sparse Linear Algebra on the GPU (link on class website)	
03/15	SPRING	BREAK.	NO	CLASS.
03/17	SPRING	BREAK.	NO	CLASS.
03/22	Advanced Optimization Techniques for GPU computing under CUDA	Mazhar		Both HW6 and a two page intermediate report for the Midterm Project are due today at 11:59 PM
03/24	Parallel Programming with MPI	Negrut	Read Chapters 1 through 3 of CUDA Programming Guide 3.2 (link on class website)	
03/29	Parallel Programming with MPI	Negrut		
03/31	Parallel Programming with OpenMP	Negrut	HW7[1] (due 04/07) Read Chapters 4 & 5 of the CUDA Programming Guide 3.2 (link on class website)	HW is related to MPI programming
04/05	Parallel Programming using OpenMP	Negrut		One to two page PDF doc with your proposal for the Final Project due on 04/06 at 11:59 PM. Use Learn@UW FinalProject drop-box to provide the doc.
04/07	Parallel Programming – Algorithm Design Decisions	Negrut	HW8[2] (due 04/14) Browse the Appendices of the CUDA Programming Guide 3.2 (link on class website)	HW is related to OpenMP programming
04/12	Dan is out travelling. Makeup class scheduled for May 03, see 05/03 entry.			
04/14	Final Project[4]: Topic Presentation[5]	Each Student	Read "CUDA C Best Practices Guide" (link	Each student presents her/his final project topic and outlines

[4] Students expected to work individually on the Final Project. Final Project due on Monday, May 9, 11:59 PM.
[5] PDF doc with your proposal for the Final Project due on 04/06 at 11:59 PM.

		on class website)	the context in which topic is relevant
04/19	MIDTERM EXAM	Can bring along the annotated hardcopies of the papers/documents that you had to read. Otherwise, it is a closed-book exam.	Two page Final Project Proposal PDF doc due today at 11:59 PM. Short presentation of results will be made during time slot for Final Exam
04/21	Guest Lecturer: PETSc Library	Matt Knepley U of Chicago (PETSc developer)	PETSc is the most widely used package for HPC on classical supercomputers (IBM Blue Gene, for instance)
04/26	Managing Projects with cMake Debugging CUDA applications	Brian Davis	
04/28	Guest Lecture: 500 Teraflops Heterogeneous Cluster	Ginger Ross of US Air Force Research Lab	Ginger will talk about the US Air Force High Performance Computing assets, including a new cluster, which is a heterogeneous system combining the capabilities of multi-code CPU head-nodes, GPGPUs, and PlayStation3s
05/03	Support for Parallel Computing in MATLAB	Narfi Stefansson of MathWorks	**Part 1**: Summary of support for GPU computing when using MATLAB. Features and implementation. **Part 2**: Covers MATLAB Parallel Computing Toolbox (PCT) and MATLAB Distributed Computing Server (MDCS).
05/05	Supercomputing for the Masses: Killer-Apps, Parallel Mappings, Scalability and Application Lifespan. Course evaluation	Rob Farber	

Final Exam: Tuesday, May 10 – 12:25PM. The two hours for the final exam used for Final Project Presentations