ME964
High Performance Computing for Engineering Applications

Spring 2011

Dan Negrut
Assistant Professor
Department of Mechanical Engineering
University of Wisconsin, Madison

January 18, 2011

"I think there is a world market for maybe five computers."
T. J. Watson, chairman of IBM, 1943.

Today's Game Plan

- Course logistics
- Brief overview of syllabus
- Motivation and central themes of this class
- Start quick overview of C programming language

Instructor: Dan Negrut

- Polytechnic Institute of Bucharest, Romania
  - B.S. – Aerospace Engineering (1992)

- The University of Iowa
  - Ph.D. – Mechanical Engineering (1998)

- MSC.Software
  - Product Development Engineer 1998-2005

- The University of Michigan
  - Adjunct Assistant Professor, Dept. of Mathematics (2004)

- Division of Mathematics and Computer Science, Argonne National Laboratory
  - Visiting Scientist 2004-2005, 2006, 2010

- The University of Wisconsin-Madison, Joined in Nov. 2005
  - Research Focus: Computational Dynamics (Dynamics of Multi-body Systems)
  - Established the Simulation-Based Engineering Lab (http://sbel.wisc.edu)

Good to know…

Time	9:30 Tu & Th
Location	1163ME
Office	2035ME
Phone	608 890-0914
E-Mail	negrut@engr.wisc.edu
Course Webpage	http://sbel.wisc.edu/Courses/ME964/2011/index.htm
Grades reported at:	learnuw.wisc.edu
ME964 Forum:	http://sbel.wisc.edu/Forum/index.php?board=3.0

ME 964 Spring 2011

- Office Hours:
  - Monday 2 – 4 PM
  - Wednesday 2 – 4 PM

- Call or email to arrange for meetings outside office hours

- Walk-ins are fine as long as they are in the afternoon

- TAs:
  - Arman Pazouki
  - Toby Heyn

Text

- No textbook is required, but there are some recommended ones:

- **Highly recommended, useful in this class**
- NVIDIA CUDA C Programming Guide V3.2, 2010 :
  - http://developer.download.nvidia.com/compute/cuda/3_2/toolkit/docs/CUDA_C_Programming_Guide.pdf
- Jason Sanders and Edward Kandrot: CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010 (on reserve, Wendt Lib.)
- David B. Kirk and Wen-mei W. Hwu: Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, 2010 (on reserve, Wendt Lib.)
- Peter Pacheco: Parallel Programming with MPI, Morgan Kaufmann, 1996 (on reserve, Wendt Lib.)
- B. Kernighan and D. Ritchie, The C Programming Language
- B. Stroustrup, The C++ Programming Language, Third Edition

- **Further reading**
- H. Nguyen (ed.), GPU Gems 3, Addison Wesley, 2007 (on reserve, Wendt Lib.)
- 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 Related Information

- Handouts will be printed out and provided before each lecture
- Lecture slides (PPT and PDF) will be made available online at class website
- Video streaming of class anticipated to be available on the same day at
  - 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
- Grades will be maintained online at Learn@UW
- Syllabus will be updated as we go
  - It will contain info about
    - Topics we cover
    - Homework assignments
  - Available at the course website
    - http://sbel.wisc.edu/Courses/ME964/2011/index.htm

Grading

Homework	40%
Midterm Exam	10%
Midterm Project	20%
Final Project	25%
Course Participation	5%
Total	100%

NOTE:

- Score related questions (homeworks/exams/labs) must be raised prior to next class after the homeworks/exams/lab is returned.

Homework Policies

- About eight or nine HWs assigned
  - No late HW accepted
    - HW due at 11:59 PM on the day indicated as due date

- Homework with lowest score will be dropped when computing final score

- Homework and projects should be emailed to me964uw2011@gmail.com
  - To get credit for your work the email time-stamp should be prior to the assignment due time/date

Midterm Exam

- One midterm exam
- Scheduled during regular class hours
- Tentatively scheduled on April 21
- Doesn’t require use of a computer (it’s a pen and paper exam)
- It’s a “closed books” exam
  - You can bring annotated copies of the papers that you are asked to read

Midterm Project

- Each one of you will have to select one of four topics by March 1
  - Topic 1: Simplified N-Body problem on the GPU
  - Topic 2: Collision detection on the GPU
  - Topic 3: Finite Element Analysis on the GPU
  - Topic 4: GPU-based parallel solution of sparse large positive definite linear system using Cholesky decomposition

- Topics listed according to their level of difficulty
  - Topics 2, 3 conference paper worth if implemented right
  - Topic 4 journal paper worth if implemented right

- Due on April 13 at 11:59 PM

- Accounts for 20% of final grade

- Project is individual

Final ~~Exam~~ Project

- Scheduled for Tuesday, May 10, 12:25 PM
  - The Final Project is due on May 9, at 11:59 PM
  - Two hour time slot used to have Final Project <u>presentations</u>
  - Additional presentation time slots will very likely be needed during finals' week
  - I will come up with a way for you to select your time slot based on your availability during the finals' week

Final ~~Exam~~ Project

- Final Project (accounts for 25% of final grade):
  - It is an individual project
  - You choose a problem that suites your research or interests
  - You are encouraged to tackle a meaningful problem
    - Attempt to solve a useful problem rather than a problem that you are confident that you can solve
    - Projects that are not successful are ok, provided you aim high enough and demonstrate good work
    - Continuing the Midterm Project is ok for Topics 2, 3, and 4
  - Tentatively,
    - Work on Final Project will start on April 15
    - Presentation of *topic* tentatively scheduled for April 7 and 12

Class Participation

- Accounts for 5% of final grade. To earn the 5%, you must:
  - Contribute at least five meaningful posts on the class Forum
    - Forum is live at: http://sbel.wisc.edu/Forum/index.php?board=3.0
    - Forum meant to serve as a quick way to answer some of your questions by instructor and other ME964 colleagues
    - Your ME964 Forum account is already set up, you should have got an email with login info

Scores and Grades

Score	Grade
92-100	A
86-91	AB
78-85	B
70-77	BC
60-69	C
50-59	D

- Grading will <u>not</u> be done on a curve
- Final score will be rounded to the nearest integer prior to having a letter assigned
  - Example:
    - 85.59 becomes AB
    - 85.27 becomes B

Rules of Engagement

- You are encouraged to discuss assignments with other class students
  - Post and read posts on Forum
- Getting **verbal** advice and suggestions from anybody is fine
- Any copying of non-trivial code is not acceptable
  - Non-trivial = more than a line or so
  - Includes reading someone else's code and then going off to write your own
- Use of third party libraries that directly implement the solution of a HW/Project is not acceptable

Rules of Engagement

- Breaking the rules:
  - Zero points on HW/Exam/Project at first occurrence
  - Automatic F final grade upon second occurrence
- These rules are vague and not meant to police you
- <u>I count on your honesty more than anything else</u>

A Word on Hardware…

- The course is designed to leverage Newton, a cluster with 48 CPU cores and 24 GPU cards
  - CPUs: Intel Xeon 5520, a quadcore chip
  - GPUs: NVIDIA TESLA C1060
    - 240 Scalar Processors each
    - 4 GB global memory on the device

- Each student receives an individual account on Newton to be used for
  - GPU computing
  - MPI-enabled parallel computing
  - OpenMP multi-core computing

- A second 64 CPU core and 32 GPU card cluster available by end of February (we'll call this machine Euler)

- Advice: if possible, do all the programming on a local machine with Developer Studio and CUDA. Move to the cluster for "production" runs

A Word on Software…

- Newton managed with Windows HPC Server 2008 R2

- We will be using Microsoft Developer Studio 2008
  - Already available on Newton
  - You can download for free and install Developer Studio thanks to UW-Madison agreement with Microsoft
    - Follow link: http://msdn07.e-academy.com/elms/Storefront/Home.aspx?campus=uwm_cae

- We will be using for GPU computing NVIDIA's CUDA 3.2

- If Euler becomes available, it will run Linux but Linux will not be supported due to limited TA/instructor bandwidth

Staying in Touch…

- Please do not email me unless you have a personal problem
  - Examples:
    - Good: Schedule a meeting outside office hours
    - Bad: Asking me clarifications on Problem 2 of the current assignment (this needs to be on the Forum)
    - Bad: telling me that you can't compile your code (this should also go to the Forum)

- Any course-related question should be posted on the Forum
  - I continuously monitor the Forum
  - If you can answer a Forum post, please do so (counts towards your 5% class participation and helps me as well)
  - Keeps all of us on the same page

Course Objectives

- Introduce student to existing High-Performance Computing (HPC) software and hardware
  - Usually "high-performance" refers to parallel architectures or vector machines; i.e., architectures that have the potential to run much faster compared to your desktop computer

- Help you recognize and appreciate the fact that there are numerous applications/problems that can be solved in a parallel fashion

- Help you gain basic skills that will help you map this applications onto a parallel computing hardware/software stack

- Present basic software design patterns for parallel computing

Course Objectives

- What I'll try to accomplish
  - Provide enough information for you to start writing software that can leverage parallel computing to hopefully reduce the amount of time required by your simulations to complete
  - Emphasis is on GPU computing
- What I will not attempt to do
  - Investigate how to design new parallel computing languages or language features, compilers, how new hardware should be designed, etc.
- To summarize,
  - I'm a Mechanical Engineer, a consumer of parallel computing
  - I'm not a producer of parallel computing

Course Emphasis

- There are multiple choices when it comes to implementing parallelism
  - Pthreads, Intel's TBB, OpenMP, MPI, Ct, Cilk, CUDA, Etc.

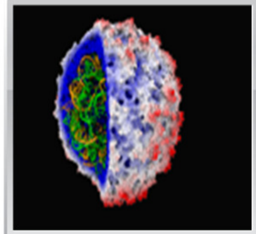
- Emphasis will be on HPC on the Graphics Processing Unit (GPU)
  - GPU computing typically associated with fine grain parallelism
  - Three lectures will be dedicated to the Message Passing Interface (MPI) HPC model, which is aimed at coarse grain parallelism
  - One lecture dedicated to OpenMP

- Why emphasize GPU Computing?
  - There are more than 60 million computers in use today that have a CUDA enabled GPU card
  - GPU computing proved to deliver significant speedups at very affordable prices

GPU Proved Fast in Several Applications

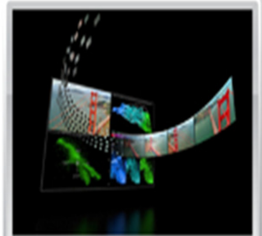
146X

Medical Imaging
U of Utah



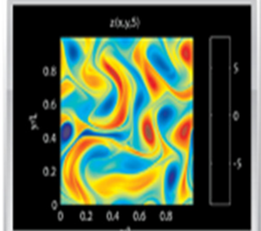
36X

Molecular Dynamics
U of Illinois, Urbana



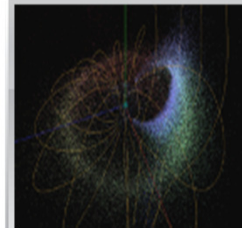
18X

Video Transcoding
Elemental Tech



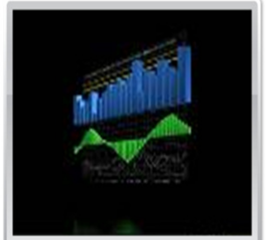
50X

Matlab Computing
AccelerEyes



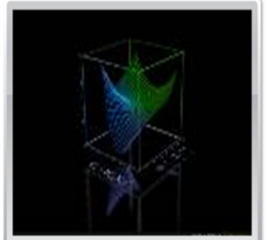
100X

Astrophysics
RIKEN



149X

Financial simulation
Oxford



47X

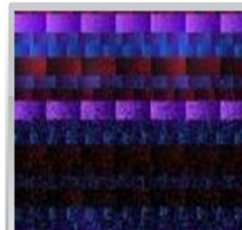
Linear Algebra
Universidad Jaime

20X

3D Ultrasound
Techniscan

130X

Quantum Chemistry
U of Illinois, Urbana



30X

Gene Sequencing
U of Maryland

Who Will Be the ME964 Student?

- Hard to pinpoint the typical student
- 45 students enrolled coming from 14 UW departments
  - Astronomy (2), Biomedical Engineering (1), Chemical Engineering (2), Chemistry (1), Civil and Environmental Engineering (2), Computer Science (3), Electrical Engineering (8), Engineering Mechanics (1), Business Management and Human Resources (1), Materials Science (1), Mechanical Engineering (13), Medical Physics (4), Nuclear Engineering and Engineering Physics (2), Physics (4)

- Title says "High Performance Computing for Engineering Applications"
- Typical student is from the College of Engineering
- I did not advertise the class with the CS department since the material would probably be boring

Why This Title?

- Computer Science: ISA, Limits to Instruction Level Parallelism and Multithreading, Pipelining, Memory Hierarchy, Memory Transactions, Cache Coherence, etc.
  - Long story short: how should a processor be built?

- Electrical Engineering: how will we build the processor that the CS colleagues have in mind?

- This class: how to use the system built by electrical engineers who implemented the architecture devised by the CS colleagues
  - At the end of the day, in our research we'll be dealing with one of the seven dwarfs…

Phillip Colella's "Seven Dwarfs"

High-end simulation in the physical sciences = 7 numerical methods:

1. Structured Grids (including locally structured grids, e.g. Adaptive Mesh Refinement)
2. Unstructured Grids
3. Fast Fourier Transform
4. Dense Linear Algebra
5. Sparse Linear Algebra
6. Particles
7. Monte Carlo

- If add 4 more for embedded, covers all 41 EEMBC benchmarks
  8. Search/Sort
  9. Filter
  10. Combinational logic
  11. Finite State Machine
- Note: Data sizes (8 bit to 32 bit) and types (integer, character) differ, but algorithms the same

Slide from "Defining Software Requirements for Scientific Computing", Phillip Colella, 2004 . Credit: D. Patterson

Overview of Material Covered

- Quick C Intro
- General considerations vis-à-vis trends in the chip industry
- Overview of parallel computation paradigms and supporting hardware/software
- GPU computing and the CUDA programming model
- Brief intro to MPI and OpenMP programming
- Midterm/Final Project related discussions
- Two or three tentative guest lectures

Overview of the GPU (CUDA) component…

- GPU Computing and CUDA Intro
- CUDA Memory Model
- CUDA Hardware
- GPU Compute Core
- Bank Conflicts
- Control Flow in CUDA
- Parallel Programming - Application Performance
- Parallel Programming - Algorithm Styles

Prerequisites

- This is a high-level graduate class in a very fluid topic

- Familiarity with C is expected

- Good programming skills are necessary
  - Understanding pointers
  - Being able to wrestle with a compile error on your own
  - Having used a debugger
  - Having used a profiler

At the beginning of the road…

- Teaching the class for the second time
  - There will be rough edges
  - There might be questions that I don’t have an answer for
    - I promise I’ll follow up on these and get back with you (on the Forum)

- Please ask questions (be curious)

My Advice to You

- If you can, do something remarkable, innovate, amaze everybody…

Acknowledgements

- Students helping with this class
  - Andrew Seidl
  - Arman Pazouki
  - Toby Heyn
  - Naresh Khude
- College of Engineering - financial support for video recording
- Department of ME – support of one TA
- Microsoft – financial support to develop the course material
- NVIDIA – financial support to build Newton and Euler

End: Discussion of Syllabus

Beginning: Quick Review of C