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

Fall 2015

Dan Negrut
Vilas Associate Professor
Department of Mechanical Engineering
Department of Electrical and Computer Engineering
University of Wisconsin, Madison

September 2, 2015

© Dan Negrut, 2015
ECE/ME/EMA/CS759 UW-Madison
“Educating does not consist in filling jars, but in lighting fires.”
Michel de Montaigne (1533-1592)
• **Purpose of today’s lecture**
  - Get a 30,000 perspective on this class and understand whether this is a class worth taking

• **What we will cover today**
  - Course logistics
  - Brief overview of syllabus
  - Motivation and central themes of this course
Acknowledgements

- Students helping with this class
  - Ang Li [grader/TA]
  - Colin Vanden Heuvel [takes care of hardware and software]
  - Hammad Mazhar [help with CUDA & thrust]

- NVIDIA, AMD & US Army ARO:
  - Financial support to build Euler, the CPU/GPU cluster used in this class
Instructor: Dan Negrut

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

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

- MSC.Software
  - Product Development Engineer 1998-2005

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

- Division of Mathematics and Computer Science, Argonne National Laboratory

- 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](http://sbel.wisc.edu))
I’m a Mechanical Engineer

- No formal training in CS or ECE
- Last time I took a course in programming was 1987 and I used punch cards
- Learned this stuff because I needed it in my work
Good to know…

- **Time**: 2:30-3:45 PM Mo & Wd & Fr
- **Location**: 1610EH
- **My Office**: 4150ME
- **Phone**: 608 265 6124
- **E-Mail**: negrut@wisc.edu
- **Course Webpage**: [http://sbel.wisc.edu/Courses/ME964/2015/index.htm](http://sbel.wisc.edu/Courses/ME964/2015/index.htm)
- **Grades reported at**: [learnuw.wisc.edu](http://learnuw.wisc.edu)
ME 759 Fall 2013

- Office Hours:
  - Tu 09:00 – 10:00 AM
  - Th 09:00 – 10:00 AM

- Call or email to arrange for meetings outside office hours
No textbook is required, but there are some recommended ones:

**Highly recommended**

- NVIDIA CUDA C Programming Guide V5.5, 2013
- Jason Sanders and Edward Kandrot: CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010 (on reserve, Wendt Lib.)
- Peter Pacheco: An Introduction to Parallel Programming, Morgan Kaufmann, 2011
- B. Kernighan and D. Ritchie, The C Programming Language
- B. Stroustrup, The C++ Programming Language, Third Edition
References [Cntd.]

- **Further reading**
  - D. Negrut, Primer: Elements of Processor Architecture. The Hardware/Software Interplay, link on class website
  - Wen-mei W. Hwu (editor), GPU Gems 4, 2011, Addison Wesley
  - Rob Farber: CUDA Application Design and Development, Morgan Kaufmann 2011
  - H. Nguyen (editor), GPU Gems 3, Addison Wesley, 2007 (on reserve, Wendt Lib.)
  - Peter Pacheco: Parallel Programming with MPI, Morgan Kaufmann, 1996
  - Michael J. Quinn: Parallel Programming in C with MPI and OpenMP, McGraw Hill, 2003
Course Related Information

- This course is offered on an accelerated track
  - Three lectures per week, each 75 minutes long

- Why accelerated track?
  - It’ll give us more than one month to work on a meaningful Final Project
    - More on this later

- Last lecture: November 16
  - 28 lectures plus one midterm exam for a grand total of 29 meetings
    - Just like a regular semester yet compressed in 2.5 months

- After Nov. 16:
  - I will still have office hours
  - Homework will continue to be assigned past Nov. 16
    - A total of 11 assignments, in line with a “non-accelerated” and normal course
Course Related Information

- I used to provide handouts but this is not feasible anymore
  - Class size too large, 92 people registered – we’ll go through a forest this semester
  - A PDF with the slides used will be emailed to you 15 mins before the beginning of each lecture

- Lecture material (PDF and audio) will be made available online at class website

- 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/ME964/2015/](http://sbel.wisc.edu/Courses/ME964/2015/)
The 964 Issue

- Class first taught in 2008
- Called ME964
- 900-level classes are experimental, need to change to 700 format
- Now cross-listed in ME, ECE, EMA, and CS as 759
- All old websites, links, forum, etc. – still reference the 964 number
- Apologies for any confusion this might cause
Grading

- Homework 40%
- 1st Midterm Exam 15%
- 2nd Midterm Exam 15%
- Final Project 25%
- Course Participation 5%

- Total 100%

NOTE:
- Score related questions (homework/exam) must be raised prior to next class after the homework/exam is returned.
Homework Policies

- There will be 11 HWs assigned
  - No late HW accepted
    - HW due at 11:59 PM on the due day

- The assignments with two lowest scores will be dropped when computing final score

- Homework and projects should be handed in using Learn@UW dropbox
  - There will be a window when you can submit your homework

- This class is hard because of the assignments. Very time consuming
Speaking of Homework…

- First Assignment goes out today

- Available on the class website (which will be up later today)

- Due in one week
  - Wd, September 9, at 11:59 PM
  - Upload your zipped homework directory at Learn@UW

- Assignment concerns general C programming warm up
1st Midterm Exam

- Accounts for 15% of final grade
- Scheduled during regular class hours
- Scheduled on 10/09/2015
  - Review offered the evening before, 7:15-9:15 PM, location TBA
- Doesn’t require use of a computer (it’s a pen and paper exam)
- It’s an “open books” exam
- Covers the material discussed in class up to that point
2nd Midterm Exam
[in lieu of final exam]

● Accounts for 15% of final grade

● Two hours long, scheduled for the evening of 11/23
  ○ Starts at 7:15 PM
  ○ Review offered the same day during regular lecture (2:30-3:45 PM)

● Doesn’t require use of a computer (it’s a pen and paper exam)

● It’s an “open books” exam

● Covers the entire material discussed in class
● There will be no final exam but rather a Final Project

● The Final Project is due on at 11:59 PM on the day the final exam was scheduled

● Each student/team will present the project in a 15 minute time slot

● Presentation time slots will be posted in doodle for you to choose a convenient one
Final Project

- Final Project (accounts for 25% of final grade):
  - It is an individual project or produced by student team
  - 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 topic is ok (shifting focus on sparse systems)
  - Work on Final Project starts on Nov. 13 after submitting project proposal
Final Project

- Default choices for Final Project

- In case you don’t have any research topic that you could use as a vehicle for this project I’ll provide at least two default choices

- Project 1: A benchmarking analysis to compare different libraries on different architectures
  - Very much in the vein of a MS thesis I completed with a 2013 ME759 student:

- Project 2: An infrastructure for heterogeneous parallel computing
  - Generate a software capability that can handle large complex problems by leveraging GPU computing and/or multi-core computing and/or distributed memory computing
  - Used in simulating the time evolution of multi-physics applications
    - Examples:
      - Meniscus simulation
      - Robotics
      - Granular dynamics
      - Polymer processing
      - CFD
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=15.0
    - Forum meant to serve as a quick way to answer some of your questions by instructor and other 759 colleagues
    - You should get an email with login info shortly (hopefully tomorrow)
  - I should know your name (from office hours or asking/answering questions in class)
Scores and Grades

<table>
<thead>
<tr>
<th>Score</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>92-100</td>
<td>A</td>
</tr>
<tr>
<td>86-91</td>
<td>AB</td>
</tr>
<tr>
<td>78-85</td>
<td>B</td>
</tr>
<tr>
<td>70-77</td>
<td>BC</td>
</tr>
<tr>
<td>60-69</td>
<td>C</td>
</tr>
<tr>
<td>50-59</td>
<td>D</td>
</tr>
</tbody>
</table>

- Grading will **not** 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
Prerequisites

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

- Familiarity with C is needed
  - You can probably be fine if you are a friend of Java

- Decent programming skills are necessary
  - Understanding pointers
  - Being able to wrestle with a compile error on your own
  - You have used a debugger
  - You have heard of a profiler
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

- Copy/paste 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 unless explicitly asked to do so
A Word on Hardware…

- The course designed to leverage a dedicated CPU/GPU cluster called Euler.

- Each student receives an individual account that will be used for:
  - GPU computing
  - OpenMP multi-core computing
  - MPI-enabled parallel computing

- Advice: if possible, do all the programming on a local machine. Move to Euler for “production” runs.
  - Better yet, do the homework on your laptop or desktop.
  - Caveat: make sure your code runs on Euler since there’s where the TA will run your code.
ME759
Heterogeneous Cluster
Euler, from another angle…
ME759
Heterogeneous Cluster

- More than 50,000 GPU scalar processors
- More than 1,200 CPU cores
- Fast Mellanox Infiniband Interconnect (QDR), 40Gb/sec
- About 2.7 TB of RAM
- More than 30 Tflops Double Precision
Latest Addition to Euler

- Eight GPU node configuration
- About 50,000 billion operations per second
- Cumulative bandwidth: 2.4 TB/second
  - My phone, on a good day: 10 MB/second
- Can be programmed w/ CUDA
  - We’ll learn about it in two weeks
A Word on Software…

- We will use Linux as our operating system of choice
  - Euler runs Linux

- We’ll use the following versions of libraries/releases:
  - CUDA: 7.0
  - MPI: 2.0
  - OpenMP: 3.0

- Reliance on makefiles generated with CMake, a build utility tool
  - Scripts will be available to you in order to facilitate compile/link/debug/profile process

- We will use a suite of debugging and profiling tools
  - gdb: debugger under Linux
  - cuda-gdb: debugger for CUDA applications running on the GPU
  - NVIDIA Profiler: Nsight

- Most of these tools are embedded in Eclipse
  - OK to work under Windows, yet make sure your code compiles/runs on Euler before submitting
Staying in Touch…

- Please do not email me unless you have a personal problem
  - Examples:
    - Good: Schedule a one-on-one 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

- The forum is **very** useful
Course Emphasis

- There are multiple choices when it comes to implementing parallelism
  - PThreads, Intel’s TBB, OpenMP, MPI, Ct, Cilk, CUDA, etc.

- Course focuses on parallelism enabled by
  - The Graphics Processing Unit (GPU), mostly aimed at fine grain level parallelism
  - OpenMP standard, aimed both at fine and coarse level parallelism
  - Message Passing Interface (MPI) standard, aimed at coarse grain parallelism

- This is not going to be a hard course but it’ll be a very busy course
  - You’ll easily understand all the material that we’ll cover (no rocket science)
  - The assignments are going to be time consuming
    - Writing software is time consuming
    - Writing parallel computing software adds insult to injury
Course Objectives

- Get familiar with today’s software and hardware for parallel computing
- Help you recognize applications/problems that can draw on HPC
- Help you gain basic skills that will help you map these applications onto a parallel computing hardware/software stack
  - Write code, build, link, run, debug, profile
- Introduce basic software design patterns for parallel computing
- Change your mind set when it comes to writing software for scientific computing
Course Objectives

[Cntd.]

- 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

- 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
  - Focus is not on how to design parallel computing hardware or instruction architecture sets for parallel computing
High Performance Computing for Engineering Applications

Why This Title?

- Computer Science: ISA, Limits to Instruction Level Parallelism and Multithreading, Speculative Execution, Pipelining, Memory Hierarchy, Memory Models, 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?
  - Lots of microarchitecture issues

- 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 in Science/Engineering 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 four more for embedded, covers all 41 EEMBC benchmarks
  8. Search/Sort
  9. Filter
  10. Combinational logic
  11. Finite State Machine
Who Will Be the Typical 759 Student?

- 92 students enrolled coming from several UW-Madison departments
  - Computer Science, Electrical and Computer Engineering, Engineering Mechanics, Mechanical Engineering, Math, Physics, and Engineering Physics

- I don’t know whether there is a typical CS/ECE/ME/EP759 student
  - Fair to say that ME/EMA/EP students will have to work harder

- The course assumes a decent level of programming experience of a typical Engineer
Auditing the Course

- Why auditing?
  - Augments your experience with this class
  - You get an account on the CPU/GPU cluster
  - You will be added to the email list
  - Can post questions on the forum

- How to register for auditing:
  - In order to audit a course, a student must first enroll in the course as usual. Then the student must request to audit the course online. (There is a tutorial available through the Office of the Registrar.) Finally, the student must save & print the form. Once they have obtained the necessary signatures, the form should be turned in to the Academic Dean in the Grad School at 217 Bascom. The Grad School offers more information on Auditing Courses in their Academic Policies and Procedures.

Tutorial website: [http://www.registrar.wisc.edu/isis_helpdocs/enrollment_demos/V90CourseChangeRequest/V90CourseChangeRequest.htm](http://www.registrar.wisc.edu/isis_helpdocs/enrollment_demos/V90CourseChangeRequest/V90CourseChangeRequest.htm)

Auditing Courses: [http://www.grad.wisc.edu/education/acadpolicy/guidelines.html#13](http://www.grad.wisc.edu/education/acadpolicy/guidelines.html#13)
Scalability

- Last time I taught this course: 37 students
- Students registered as of now: 93
- Support for TA/Grading stayed constant
- It was challenging before, even more so now
- The only way forward: we’ll have to work together on this
  - The forum for this class is critical
Overview of Material Covered

[Fall 2015]

- Basic concepts related to sequential computing
- General considerations in relation to trends in the chip industry
- Overview of parallel computation paradigms and supporting hardware/software
- GPU computing and the CUDA programming model
- GPU parallel computing using the thrust template library
- OpenMP programming
- MPI programming
- Heterogeneous parallel computing with CUDA and/or OpenMP and/or MPI
At the Beginning of the Road…

- Teaching the class for the fifth time
  - Rough edges remain
  - Hard to keep pace with the rate of change in this discipline
  - There will be issues that I don’t know and/or don’t understand
    - I’ll point those out
  - There might be questions that you ask for which I don’t have an answer for
    - I’ll follow up on these and get back with you (on the Forum)

- Please ask questions (be curious)

- Historically speaking, students have complained about the work load
  - I will send out later today an email with the 2013 student evaluations to get a better picture
    - Before buckling up for the long run you’ll get a feeling about what you sign up for
My Advice to You [is simple]

- If you can, innovate – do something remarkable, something amazing, something unique
  - I bet you'll enjoy it :-)

48
Iancu de Hunedoara
[Regent of Hungay, defeated in 1456 Mehmed, the conqueror of Constantinople (1453)]

Vlad the Impeller
[born in Transylvania, king of Wallachia]

Sighisoara, Transylvania

Iancu de Hunedoara
[Regent of Hungay, defeated in 1456 Mehmed, the conqueror of Constantinople (1453)]
Siege of Belgrade, 1456