

# EECS 594 – 002

## Scientific Computing for Engineers

---

Web page for the course:

<http://www.cs.utk.edu/~dongarra/WEB-PAGES/cs594-2008.htm>

EECS 594 – 002  
Wednesday's 1:30 – 4:30

---

- ◆ **Scientific Computing for Engineers**
- ◆ **Spring 2008 - 3 credits**
  - ↳ Jack Dongarra
  - ↳ with help from:
    - » George Bosilca
    - » Karl Ferlinger
    - » Jakub Kurzak
    - » Stan Tomov
- ◆ **Class will meet in Room C233, Claxton Building**

2

## To Get Hold of Us

---

- ◆ **Email: [dongarra@cs.utk.edu](mailto:dongarra@cs.utk.edu)**
  - ↳ **Room: 413, Claxton**
  - ↳ **Phone: 974-8295**
- ◆ **Office hours:**
  - ↳ **Wednesday 11:00 - 1:00, or by appointment**
  
- ◆ **TA: Gwang Son, [son@eecs.utk.edu](mailto:son@eecs.utk.edu)**
- ◆ **Rm 349, Claxton Complex, 974-3760**
  - ↳ **OH: 10am-11am TR, or by request**

3

## Four Major Aspects Of The Course:

---

1. Start with current trends in high-end computing systems and environments, and continue with a practical short description on parallel programming with MPI, OpenMP, and pthreads.
2. Deal with solvers: both iterative for the solution of sparse problems of part II, and direct for dense matrix problems. Algorithmic and practical implementation aspects will be covered.
3. Illustrate the modeling of problems from physics and engineering in terms of partial differential equations (PDEs), and their numerical discretization using finite difference, finite element, and spectral approximation.
4. Various software tools will be surveyed and used. This will include PETSc, Sca/LAPACK, MATLAB, and some tools and techniques for scientific debugging and performance analysis.

4

## Grades Based on:

---

- ◆ 30% on weekly homework  
(the lowest homework grade will be dropped)
- ◆ 30% on a written report and presentation  
(20 pages circa.)
- ◆ 30% on a final exam (2 hours)
- ◆ 10% on class participation.

5

## Homework

---

- ◆ Usually weekly
- ◆ Lowest grade will be dropped
- ◆ Must be turned in on time (no late homework)
- ◆ Don't copy someone else's homework.
- ◆ Sometimes problems, sometimes programming assignment, sometimes requiring running a program to find the solution.

6

## Homework (continued)

---

- ◆ We expect an analysis and detailed discussion of the results of your efforts.
  - ⌞ The program itself is not very interesting.
- ◆ Programming in C or Fortran.
- ◆ Will go over the assignments the week they are due.
- ◆ See class web page weekly for details.

7

## Using the SInRG Clusters

---

### Battlecat:

- ◆ 8x Intel Core 2 Duo 2.13 GHz
  - ⌞ 8 Nodes
  - ⌞ 2 GB RAM
  - ⌞ Gig-E

### Grig:

- ◆ 128x Intel Xeon 3.2 GHz
  - ⌞ 64 Nodes
  - ⌞ 4G RAM
  - ⌞ Myrinet 2000

### Frodo:

- ◆ 128x AMD Opteron 1.4 GHz
  - ⌞ 64 Nodes
  - ⌞ 4G RAM
  - ⌞ Myrinet 2000

### MSC:

- ◆ 96x AMD Opteron 265 (Dual Core) 1.8GHz Processors
  - ⌞ 24 Nodes
  - ⌞ 4G RAM
  - ⌞ Silverstorm 10Gb/s Infiniband NIC
  - ⌞ Mellanox 20Gb/s Infiniband NIC
  - ⌞ Myricom Myrinet 10G NIC(16 of these)

8

## Build a Cluster

---

- ◆ Form subgroups
- ◆ Each subgroup will get a cluster to put together and experiment with.
  - ↳ Intel dual core based with a GPU from Nvidia
- ◆ Put software on and run experiments

9

## Project

---

- ◆ Topic of general interest to the course.
- ◆ The idea is to read three or four papers from the literature (references will be provided)
- ◆ Implement the application on the cluster you build
- ◆ Synthesize them in terms of a report (~20 pages)
- ◆ Present your report to class (~30 mins)
- ◆ New ideas and extensions are welcome, as well as implementation prototype if needed.

10

## Remarks

---

- ◆ Hope for very interactive course
- ◆ Willing to accept suggestions for changes in content and/or form

11

## Final Exam

---

- ◆ In class
- ◆ Will cover the material presented in the course
- ◆ ~2 hours

12

## Material



### ◆ Book:

↖ [The Sourcebook of Parallel Computing](#),  
Edited by Jack Dongarra, Ian Foster,  
Geoffrey Fox, William Gropp,  
Ken Kennedy, Linda Torczon, Andy White,  
2002, 760 pages, ISBN 1-55860-871-0,  
Morgan Kaufmann Publishers.

- ◆ For each lecture a set of slides will be made available in pdf or html.
- ◆ Other reading material will be made available electronically if possible.
- ◆ The web site for the course is:
  - ↖ <http://www.cs.utk.edu/~dongarra/WEB-PAGES/cs594-2008.htm>

13

## Other Sources

- ◆ Will use material from the internet (manuals, papers)
- ◆ Will use a variety of book sources; including
  - ↖ Ian Foster
    - » Designing and Building Parallel Programs
  - ↖ Alices E Koniges
    - » Industrial Strength Parallel Computing
  - ↖ Jack Dongarra, Iain Duff, Danny Sorensen, Henk van der Vorst
    - » Numerical Linear Algebra for High Performance Computers
  - ↖ Ananth Gramma et al.
    - » Introduction to Parallel Computing
  - ↖ Michael Quinn
    - » Parallel Programming
  - ↖ David E. Culler & Jaswinder Pal Singh
    - » Parallel Computer Architecture
  - ↖ George Almasi and Allan Gottlieb
    - » Highly Parallel Computing

14

## Important Place for Software

---

- ◆ **Netlib - software repository**
  - ↳ Go to <http://www.netlib.org/>

15

## What will we be doing?

---

- ◆ **Learning about:**
  - ↳ **High-Performance Computing.**
  - ↳ **Parallel Computing**
  - ↳ **Performance Analysis**
  - ↳ **Computational techniques**
  - ↳ **Tools to aid parallel computing.**
  - ↳ **Developing programs using PVM, MPI, HPF, and perhaps OpenMP.**

16

## Outline of the Course

---

1. January 9 Introduction to High Performance Computing
2. January 16 HPC Architectures
3. January 23 Message Passing
4. January 30 Message Passing continued
5. February 6 Parallel Programming Paradigms
6. February 13 Parallel Programming Paradigms and Performance
7. February 20 Floating Point Arithmetic, Memory Hierarchy & Cache
8. February 27 Dense Linear Algebra
9. March 5 Dense Linear Algebra, part 2 and Cell processor
10. March 12 Performance Analysis Tools;  
March 19 - Spring Break
11. March 26 Projection and its importance in scientific computing
12. April 2 Discretization of PDEs
13. April 9 Sparse matrices
14. April 16 Iterative Methods in Linear Algebra
15. April 23 Iterative Methods in Linear Algebra, part 2
16. April 30 Class Final reports

17

## What you should get out of the course

---

**In depth understanding of:**

- ◆ **When is parallel computing useful?**
- ◆ **Understanding of parallel computing hardware options.**
- ◆ **Overview of programming models (software) and tools.**
- ◆ **Some important parallel applications and the algorithms**
- ◆ **Performance analysis and tuning**

18

## Background

---

- ◆ C and/or Fortran programming
- ◆ Knowledge of parallel programming
- ◆ Some background in numerical computing.

19

## Computer Accounts

---

- ◆ For much of the class computing you can use one of our set of computer clusters. More on this later
- ◆ If you have an account in the Department you have access to the TORC cluster: torc1 through torc8.
- ◆ Cluster of PC's:
  - ↳ <http://icl.cs.utk.edu/iclhelp/> look under clusters

20

CS 594 - Scientific Computing for Engineers

Homework #1  
January 9, 2008  
Due: January 23, 2008

I would like you to implement a version of the following mathematical operations:

- the 2-norm of a vector,

$$\|x\|_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^n x_i^2}$$

- matrix - vector multiplication,

$$y = y + A * x$$

$$y_i = y_i + \sum_{j=1}^n A_{i,j} * x_j, \text{ for } i=1, \dots, n$$

- matrix multiplication

$$C = C + A * B$$

$$C_{i,j} = C_{i,j} + \sum_{k=1}^n A_{i,k} * B_{k,j}, \text{ for } i, j=1, \dots, n$$

The point of this assignment is not to write software, but to look at the performance for each of your implementations and try to explain why you are getting the performance you see and what you could do to increase the performance. You should produce a software implementation for each and run some experiments on various systems, in particular use processors from battlecat, frodo, msc, and grid clusters. I would like to see a report and analysis of your results, perhaps some plots of your performance data for n between say 10 and 1000. Please verify and convince me that you are computing the correct results in each case. Let me know what computers you used and how you are getting the performance results as well.

Our TA, Gwang Son, will have a set of timer you can use to measure the execution time of your programs. See the course webpage for details.

You can find out information on various processors at:

<http://www.cpu-world.com/CPU%20index.html>  
<http://www.cpu-world.com/sspec/index.html>

