

CS 594 – 31778 Understanding Parallel Computing

Web page for the course:

<http://www.cs.utk.edu/~dongarra/WEB-PAGES/cs594-2005.html>

CS 594 –
Wednesday's 1:30 – 4:00

- ◆ **Understanding Parallel Computing:
From Theory To Practice**
- ◆ **Spring 2003 - 3 credits**
 - ⌞ Jack Dongarra, Professor
- ◆ **Class will meet in Room C211, Claxton Building**

2

To Get Hold of Us

- ◆ **Email: dongarra@cs.utk.edu**
 - ⌞ Room: 413, Claxton
 - ⌞ Phone: 974-8295
- ◆ **Office hours:**
 - ⌞ Wednesday 11:00 - 1:00, or by appointment
- ◆ **TA: Fengguang Song, song@cs.utk.edu**
 - ⌞ 307 Claxton Complex, 974-0517
 - ⌞ OH: 10am-11am TR, or by request

3

Schedule of Topics

- ◆ Introduction
- ◆ **Parallel Programming Models and Machines**
 - ⌞ Shared Memory and Multithreading
 - ⌞ Distributed Memory and Message Passing
 - ⌞ Data parallelism
- ◆ Sources of Parallelism in Simulation
- ◆ **Algorithms and Software Tools (depends on student interest)**
 - ⌞ Dense Linear Algebra
 - ⌞ Partial Differential Equations (PDEs)
 - ⌞ Load balancing, synchronization techniques
 - ⌞ Sparse matrices
 - ⌞ Visualization and monitoring
 - ⌞ Debugging parallel programs
 - ⌞ Sorting and data management
 - ⌞ Metacomputing
- ◆ Applications (including guest lectures)
- ◆ Project Reports

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

- ◆ **Boba Cluster**
 - ⌞ 32 Dell Precision 530s
 - ⌞ Dual Pentium IV Xeon 2.4 GHz Processors
 - ⌞ 512 KB Cache
 - ⌞ 2 GB Ram
 - ⌞ 2 73GB MAXTOR 6L080J4 Disk Drives
 - ⌞ On board 3Com Corporation 3c905C NIC
 - ⌞ Intel e1000 100/1000 NIC
- ◆ **Frodo Cluster**
 - ⌞ 65 dual AMD-Opteron 240 nodes
 - ⌞ 2 GB RAM per node
 - ⌞ Myrinet 2000 interconnect
- ◆ **Neo Cluster**
 - ⌞ 16 Dual 450MHz UltraSPARC-II 64-bit RISC processors with 4MB L2 cache
 - ⌞ 512MB (4x128) ECC SDRAM DIMM memory installed
 - ⌞ SCSI: Integrated dual 40 MB/sec UltraSCSI channels
 - ⌞ 27.3GB (1-18GB, 1-9GB) UltraSCSI 10,000 RPM Hard Drives
 - ⌞ TP Ethernet 10/100BASE-Tt
 - ⌞ SysKconnect Gigabit Ethernet

8

Project

- ◆ Topic of general interest to the course.
- ◆ The idea is to read three or four papers from the literature (references will be provided)
- ◆ 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.

9

Remarks

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

10

Final Exam

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

11

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/cs594-2005.html>



12

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 et al.
 - Sourcebook of Parallel Computing
 - ▮ 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

13

Important Place for Software

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

14

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.**

15

Outline of the Course

1. January 12 - Introduction
2. January 19 - Parallel Architectures
3. January 26 - Parallel Programming, Memory Hierarchy and Cache
4. February 2 ?
5. February 9 - Message Passing
6. February 16 - Message Passing continued
7. February 23 - Dense Linear Algebra
8. March 2 - Dense Linear Algebra continued
9. March 9 - Iterative Linear Algebra Methods
10. March 16 - Iterative Linear Algebra Methods continued
March 23 - Spring Break
11. March 30 - Grid Computing
12. April 6 - Grid Computing continued
13. April 13 - Tools for Debugging
14. April 20 - Tools continued
15. April 27 - Last Class

16

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

17

Background

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

18

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/internal/iclhelp/clusters.html>

19

Homework #1
January 12, 2005
Due: January 26, 2005

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,

$$y = Ax + b$$
$$y_i = x_j + \sum_{k=1}^n A_{ij} x_k, \text{ for } i=1, \dots, n$$

- matrix multiplication

$$C = A * B$$
$$C_{ij} = \sum_{k=1}^n A_{ik} * B_{kj}, \text{ for } i, j=1, \dots, n$$

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. 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, Fangguang Song, gsong@cs.utk.edu, will have a set of timer you can use to measure the execution time of your programs.

You can find out information on various processors at:

20