CS 594 – 004
Scientific Computing for Engineers


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: Wes Alvaro <a@utk.edu>
- Rm 351, Claxton Complex, 974-
  > OH: Wednesday’s 10am-12am or by request

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. Put together a cluster and experiment.
2. Deal with numerical linear algebra solvers: both direct dense methods and direct and iterative methods for the solution of sparse 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.

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.

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.
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.

Using the SInRG Clusters

- Grig:
  - 128x Intel Xeon 3.2 GHz
  - 64 Nodes
  - 4G RAM
  - Myrinet 2000

- Mordor:
  - AMD Opteron 8356 2.46GHz
  - 8 nodes (118 cores; 16 cores total)
  - 32G RAM
  - GigE
  - 2x Myricon 10G PCI-E Cards

- Battlecat:
  - 8x Intel Core 2 Duo 2.13 GHz
  - 8 Nodes
  - 2 GB RAM
  - Gig-E

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 and AMD
- Put software on and run experiments

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 (~10-15 pages)
- Present your report to class (~30 mins)
- New ideas and extensions are welcome, as well as implementation prototype if needed.

Remarks

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

Final Exam

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

- **Book:**  
  The Sourcebook of Parallel Computing,  
  Edited by Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, Andy White,  
- 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:  

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, Ian 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

Important Place for Software

- **Netlib** - software repository  
  Go to http://www.netlib.org/

What will we be doing?

- Learning about:  
  - High-Performance Computing.  
  - Parallel Computing  
  - Performance Analysis  
  - Computational techniques  
  - Tools to aid parallel computing,  
  - Developing programs in C or Fortran using MPI and perhaps OpenMP.

Outline of the Course

2. January 19: Dense Linear Algebra  
3. January 26: Parallel programming paradigms and their performances  
4. February 2: Message Passing (MPI) the basics  
5. February 9: Message Passing Interface (MPI)  
6. February 16: MPI I/O  
7. February 23: Cell and GPU Programming  
8. March 2: Performance Analysis Tools  
9. March 9: OpenMP and hybrid MPI/OpenMP programming  
10. March 16 – Spring Break  
11. March 30: Projection and its importance in scientific computing  
12. April 6: Sparse matrices and optimized parallel implementations  
13. April 13: Iterative Methods in Linear Algebra  
14. April 20: Iterative Methods in Linear Algebra (Continued)  
15. April 27: A look at PageRank  
16. May 4: Class Final

What you should get out of the course

In depth understanding of:  
- Why parallel computing is 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 techniques.
Background

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

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 clusters.
- Cluster of PC's: