COSC 594 – 006

3 credit hours

Scientific Computing for Engineers

Web page for the course:

http://bit.ly/cs594-2018

CS 594
Wednesday's 1:30 – 4:30

- **Scientific Computing for Engineers**
- **Spring 2017 - 3 credits**
  - Jack Dongarra
  - with help from:
    - George Bosilca
    - Mark Gates
    - Azzam Haidar
    - Jakub Kurzak
    - Heike Jagode
    - Piotr Luszczek
    - Stan Tomov
- **Class will meet in Room C-233, Claxton Building**

To Get Hold of Us

- Email: dongarra@icl.utk.edu
  - Room: 203, Claxton
  - Phone: 974-8295
- Office hours:
  - Wednesday 11:00 - 1:00, or by appointment

- TA: Qinglei Cao, qcao3@vols.utk.edu
- TA's Office : Claxton

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:

- 40% on weekly assignments (the lowest grade will be dropped)
- 40% on a written report and presentation (20 pages circa.)
- 20% on a final exam (2 hours) & on class participation.

Homework

- Usually weekly
- Lowest grade will be dropped
- Must be turned in on time (no late assignments)
- Don't copy someone else's work.
- Sometimes problems, sometimes programming assignments, 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.

Computer Accounts

- For much of the class computing you can use one of our set of computer clusters or systems you have access to.
- If you have an account in the Department you have access to the clusters.

Using the various computer systems from NICS

- **UTK's Beacon**

Beacon

- 48 compute nodes
  - Node: 2 8-core Intel Xeon E5-2670 processors & 4 Intel Xeon Phi™ coprocessors

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

- 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://bit.ly/cs594-2018

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 OpenMP.

Outline of the Course

1. January 10th: Introduction to Class & Parallel programming paradigms and their performances
2. January 17th: Architectures and POSIX threads
3. January 24th: Introduction to High Performance Computing
4. January 31st: MPI
5. February 7th: MPI Part 2
6. February 14th: OpenMP
7. February 21th: Accelerators
8. February 28th: Performance Modeling
10. March 7th: Dense Linear Algebra Discretization of PDEs

    March 14th: Spring Break
11. March 21th: Projection and Its Importance in Scientific Computing
12. March 28th: Deep Learning
13. April 4th: Discretization of PDEs and Parallel Solvers
12. April 11th: Sparse Matrices and Optimized Parallel Implementations
13. April 18th: Iterative Methods in Linear Algebra Part 1
14. April 25th: Iterative Methods in Linear Algebra Part 2
15. May 3rd: Final and reports

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.