Math 4370/6370
Parallel Scientific Computing

Daniel R. Reynolds
Spring 2015
reynolds@smu.edu
Parallel computing historically focused on state-of-the-art engineering and scientific applications. However, video game consoles, laptops, desktops and even cell phones have transitioned toward chips with multiple processors as well.

Course Goals:

- Introduce the foundations of parallel architectures and algorithm design.
- Investigate the relative benefits and weaknesses among different parallel approaches for various algorithms.
- Learn to use OpenMP and MPI for shared and distributed parallel algorithms.
- Learn to run programs on SMU and NSF research clusters.

Students are expected to have some experience with program design and implementation, and should feel comfortable with the Unix/Linux command line.

While familiarity with compiled languages (e.g. C, C++, Fortran, etc.) is desired, significant prior programming experience is a necessity for this class.
Course Information

Webpage:  http://faculty.smu.edu/reynolds/math6370

More detailed course information is on the syllabus (website above).

Required:  Pacheco, Introduction to Parallel Programming, Morgan Kaufmann, 2011.

Recommended Books:


Supporting Materials (online, free) are listed on the syllabus.
Lectures will be posted on the course web page, along with external readings for each topic.

Readings from Pacheco are required. Readings from additional sources are recommended.

You are expected to read through the lecture slides prior to each class, and to do the accompanying readings as we progress through the course.
This will be a very hands-on course. Homework and projects will use Linux workstations, clusters and supercomputers. Do the Linux tutorial if you haven't done so already, and get comfortable with the command line.

A Linux virtual machine has been posted on the course web page; you can use this on Windows/OS X systems to set up a usable development environment.

All students should have accounts on:
- The ManeFrame cluster.
- The NSF XSEDE portal (email your username to me).
- If your ManeFrame account is not set up yet, let me know and I will request one for you.
Programming

You will be expected to submit programming assignments for both homework and projects.

These assignments may be written in any one of the languages \{C, C++, Fortran77/90/95/2003\}, since all of these support OpenMP and MPI.

For students without significant experience in any of these, I will briefly cover C++, but I strongly recommend that you do extra tutorials and outside reading. I have lined up an online “Zybook” for these students.

All programs will be written within the Mercurial or Git revision control systems.

- Set up an account on Bitbucket (I recommend using your SMU email).
- You'll do all coursework in "private" repositories of your own making.
- In each repository, you must grant me (drreynolds) read/write access.
- I will download your homework and project codes when grading.
Grades in this course will be determined based on homework (30%), labs (20%) and projects (50%).

Homework will consist of small programs and written exercises, and will be assigned periodically throughout the semester.

Labs are self-directed, and follow instructions laid out on the class web page.

Projects:
- Pick during first 2 weeks of the semester.
- Will be due in phases, corresponding to each portion of the class.
- At end of semester, a final technical presentation will be due.

All work in this course **must be completed on your own**, although discussion of approaches/strategy with other students is encouraged, and I'll provide coding/debugging help during office hours.