Math/CS 4777

Introduction to Parallel and Vector Scientific Computing

Fall 2010, M,W,F 12:05 PM - 12:55 PM, Skiles 169

Dr. Lew Lefton, Skiles 159, 404-385-0052
Office Hours Mon. 2-3, Thu. 10-11, and by appointment
llefton@math.gatech.edu
Final Exam is scheduled Dec 13th (Mon) 11:30 AM - 2:20 PM

Your grade will be determined by 60% homework and 40% final project. Each of these will be discussed further in class.

The Textbook for this course is An Introduction to Parallel and Vector Scientific Computation. We will cover the topics in the book but we may also spend some time on other related supplementary topics.

Chapter	Topics
1	Introduction, von Neumann computer, Flynn's Classification, vector pipelines, Hockney's formulas, distributed memory systems, Amdahl's law.
2	Theoretical considerations, DAG, Complexity, Data dependencies
3	Machine Implementations, processes, threads, OpenMP, MPI
4	Floating point numbers, error propagation, basic matrix arithmetic (BLAS)
5,6,7	Direct methods for linear systems, LU, triangular systems, Thompson's algorithm, Cholesky Decompostion, QR, Matrix norms, Givens Rotation
8	Iterative Methods, Jacobi, Gauss Seidel, Fixed point iteration, Relaxation methods, Conjugate gradient
9	Eigenvalues and eigenvectors, power method, Householder Transformations, Hessenberg Form
10-11	Monte Carlo Methods, Random numbers, quadrature, MCMC
NA	Other topics

The prerequisites for the course are some basic linear algebra (Gaussian Elimination, LU, eigenvalues, etc.) and some basic programming experience in a compiled language (e.g. Fortran, C++, C, Java). You will have to write code for some of the homeworks and for the final project. It will likely require more than just Matlab, particularly for the MPI exercises.

HOMEWORK

Homework assignments will appear below as they are assigned.

Due Date	Page number	Exercises
Monday Aug 30, 2010	Page 24 - 26	3, 5, 8, 11, 14, 16

Wednesday Sep 8, 2010 (UPDATE: You may turn this in on Sep. 10 with no penalty)	Page 41 - 43	4, 6, 8, 12, 15, 17
Friday Sep 24, 2010	NA	1. In pseudo code, show how to add the terms of the harmonic series (1 + 1/2 + 1/3 + ...) using 4 threads. Stop adding terms when they become smaller than 10^(-5).
Friday Sep 24, 2010	NA	2. Implement an MPI program to perform matrix-matrix multiplication AB where A and B are both 128x128 random matrices. Run the code for 2, 4 and 8 processors.
Monday Oct 11, 2010	Page 124 - 125	1, 2, 11, 12
Monday Oct 11, 2010	NA	Write preliminary project description following these rough guidelines.
Monday Nov 1, 2010	Page 158 - 161	3, 4, 6, 14, 18
	Page 171	5, 9*
NOTE: p. 171 #9 is postponed to be turned in Dec. 3		* NOTE: this problem should be modified to read: Write a Cholesky algorithm and use it to solve the linear system on page 197. Use the boundary conditions and RHS g from exercise 7, p. 205. Namely: u(x,y)=x^2+y^2, g(x,y)=4.
	Page 184 - 185	3, 9, 10
Friday Dec 3, 2010	Page 204 - 205	1, 4, 8
	Page 227 - 228	1, 5, 8
-->		

PROJECT INFORMATION

The final project comes in three steps. First you will write a proposal. Proposal guidelines will be forthcoming. I suggest you think about a topic from your major. It should be something which can benefit from parallelization. The second step is to write the code and do the actual computation, recording results. The third step is a written and oral presentation of your project to the class.

Here is a copy of the tentative Project Schedule. Please let me know ASAP if you have a conflict or if you want to present at a different time.

PROGRAMMING EXAMPLES

The code in example1.c illustrates basic use of POSIX threads. However, it has a race condition.

The code in example2.c illustrates use of POSIX threads with a mutex call to protect shared memory.

The code in example3.c illustrates the problem with using shared memory and no mutex.

The code in example4.c fixes the problem from example 3.

USEFUL LINKS

Warning! These have not yet been updated for Fall 2010, but some are probably still valid. When surfing through these, be sure to note the date wherever possible. If the page is dated in the early to mid 1990's or before, don't spend too much

time abrsorbing detailed technical information from it. The ideas may be worthwhile, but the technology has probably changed...

- There is a quick start guide to using the cluster computing resources at the School of Math. This short guide includes instructions on getting started, resources available, links to sample programs in MPI, sample Torque/PBS scripts, and information about screen, top, nice, and other useful commands for running multiple jobs on balsa.
- I have now put the instructions for creating ssh tunnels on this website. This short howto includes instructions for both Windows and Linux.
- The Online Documentation page for the HPC group at Georgia Tech has a user guide and tutorial on using the SGI Origin 2000.
- www.openmp.org is currently one of the best supported ways to program shared memory systems. They have support for C, C++ and Fortran. OpenMP uses compiler directives, library functions and environment variables and is supported by many vendors. There are some good openmp examples from Lawrence Livermore National Labs.
- For distributed memory, MPI is a commonly used tool. These are primarily library calls and are well supported. A nice tutorial on MPI is from the Lawrence Livermore National Laboratory. I have posted some sample MPI programs here for you to use as templates.
- A nice discussion of the IEEE Standard 754 for floating point numbers and some observations and pitfalls when comparing floating point numbers may be of interest.
- Lots of Fortran and some C code is available from netlib. These are the folks who bring you the LAPACK, LINPACK, BLAS, ATLAS, etc. libraries.
- For IBM SP2 specific information, there is a very good starting point page here. Lots of links from the parent page too.
- Ian Foster at ANL has an online book. He is also quite active in grid computing efforts. You can read more from the globus project.
- Never forget google. Their directory on parallel computing and their directory on distributed computing are a good source of links.
- The Single UNIX Specification (Version 2) standard for threads. Includes manual pages of all POSIX thread functions.
- Daniel Robbins, President/CEO, Gentoo Technologies, Inc. wrote a nice three part article on threads. The examples 3 and 4 above were taken from these: POSIX threads explained, Part 1, POSIX threads explained, Part 2, and POSIX threads explained, Part 3.
- There is an older paper on parallel algorithms for linear algebra with some nice ideas. In particular, there is a discussion of a block algorithm for LU Factorization.
- I mentioned a webpage which details some disasters attributable to bad numerical computing which is worth looking at, whenever you get the feeling that all these error estimates and convergence results aren't worth the trouble.
- High Performance Fortran (HPF) is useful for data parallelism (not task parallelism). Charles Koelbel has a good tutorial here.
- Another HPF online tutorial is by A. C. Marshall located at http://www.hku.hk/cc/sp2/software/hpf/Course/HTMLSlides.html.
- Yet another one can be found at http://www.liv.ac.uk/HPC/HTMLHPFCourse/HTMLHPFCourseSlides.html
- The Portland Group compilers are available on balsa.math.gatech.edu. There is some documentaion available here. See also /usr/pgi/EXAMPLES on balsa.
- Designing and Building Parallel Programs is an online book on iterative methods by Ian Foster.
- An online guide to the field of numerical optimization can be found at the Optimization Tree.
- There are several good short programs available online from the Numerical Recipies Home Page.
- A nice discussion of the conjugate gradient method "without the agonizing pain" can be found here.
- There is a similar course at Stanford. In particular, I borrowed some notes on iterative methods from there in a recent lecture.
- Random numbers are important for Monte Carlo simulations. One place to start looking is here.