Course Description

COMP/CS 605 – Scientific Computing

Spring 2013

Instructor. James Otto, jotto@mail.sdsu.edu, 594-3505 (EBA 109).

Intent. COMP/CS 605 provides experience in designing and running parallel programs in a modern academic cluster setting. The course might be aptly called Parallel Programming with MPI and CUDA, as parallel computing is taught through the use of MPI – Message Passing Interface, following Pacheco’s MPI book as a guide, and through programming GPU devices using CUDA. One goal is to make our way through as many parallel programming exercises as time will allow in an attempt to tackle computationally intense problems from both the CPU and GPU side, and to benchmark the results. This is parallel computing by doing, primarily in a Linux cluster environment, using C and Fortran programming languages. A broad goal is to provide students with tools and expertise that will help further the computational aspect of their research efforts.

Translating a mathematical description of a problem into a computer program description is a prerequisite skill for the course, as are: fluency with the objects, language, and methods of linear algebra & undergraduate calculus, program development (using line editors – like vi, and makefiles) in the Unix command environment, and writing C-Language computer programs. (Note – basic programming is not taught in this class – but see COMP 526, where Unix/Linux command line environment is also taught). Here it is assumed that the student is an accomplished C programmer, and that one is able to expertly navigate the Unix command line interface.

Programming portions of this skill set (starting with the Unix environment) are provided in COMP 526, Computational Methods for Scientists, offered Fall semesters.

COMP/CS 605 is “hands-on” with weekly and semi-weekly programming assignments.

Requirements. The ability to program well in C is a requirement for the class. In particular, the NVIDIA CUDA compiler for GPU programming is a C compiler (with extensions). C language or Fortran 90 are supported for use in programming MPI. Familiarity with navigating, working with files, and compiling programs in a Unix or Linux environment is also assumed. Mathematical knowledge at the level of performing matrix operations in linear algebra and derivatives and (multiple) integrals in calculus is required.

The course will utilize C (required) and Fortran 90 (optional) programming languages along with MPI and CUDA extensions to emphasize key parallel programming concepts. C language is required.

Topics to include:

- Pipelining
- Caching
- Multi-level caches
- Program optimization
- Amdahl’s law
- MPI extensions to C and Fortran languages – the 6 basic MPI calls
- MPI collective communication
- Working in a Unix cluster environment
- Batch job submission using Torque/OpenPBS
• GPU programming in the CUDA environment
• Writing CUDA kernels
• CUDA programming for computationally intense problems
• Optimization of GPU codes using shared memory

Applications:
• Parallelizing numerical integration
• Parallelizing integration rules in higher dimensions
• Multiplying dense matrices using MPI
• Multiplying dense matrices using CUDA
• Multiplying dense matrices using CUDA and MPI
• Fox’s algorithm
• Iterative methods for $Ax = b$ – splitting Methods
• Richardson’s and minimum residual methods
• Preconditioned iterative methods – parallel preconditioners

Special note: Content conflict with Fall 2012 CS596 session. Due to certain circumstances the current (Spring 2013) session of COMP/CS605 overlaps material with the previous offering of CS596. Students will not be given degree credit for taking both sessions. COMP students who took CS596 during Fall 2012 will be given credit for COMP 605, in fulfillment of the basic parallel computing requirement for the COMP program. Due to this overlap, students who enroll in the 596-2012/605-2013 sequence will have advanced requirements in the 605 session: 2 class presentations, a midterm computer project, and an advanced final computing project: the ability to enroll in this capacity is subject to limitations in overall class enrollment.

Platforms. Unix is the environment of choice, with accounts provided on a small but powerful Linux workstation cluster.

Grading. Based on programming assignments, final projects, class presentations, and unannounced (pop) quizzes, and class participation – as follows,

• Class presentation(s) – 5%
• Unannounced quizzes – 10%
• Class participation – 10%
• Programming assignments (including programming project(s)) – 75%

All assignments and projects must be completed on time. In order to submit a late assignment a student must inform the instructor that an assignment is being completed past the due date. Even with notice, a penalty will be applied to the grade of any late assignment, with a minimum of 10%, and additional 10% deductions for each week beyond the due date.

Class participation is extremely important, and constitutes 10% of the student’s final grade. In order to promote an environment for consistent, participatory learning students are not allowed to miss more than 2 class sessions without penalty. Additional absences will incur a potential 5% deduction to the student’s final
grade. Included in class participation are the following: being punctual, participating in class discussions, asking pertinent questions in class, paying attention in class (not sleeping, not using laptops, tablets, or cell phones, etc.).

80% of the student’s grade is divided between frequent homework assignments with equal weighting, a 20-minute class presentation, and a final programming project equal to 2-3 homework assignments. (But for 2013, see the special note as regards Fall 2012 CS596 students.)

Precision and neatness commensurate with graduate level work are expected for all submitted work and presentations. Sloppy or lazy work will necessarily incur a grading penalty of no less than 10%. This includes typos, spelling errors, and errors in grammatical usage. In cases where the level of work falls below minimum standards only nominal credit (commensurate with a failing grade) will be given, and the student will be asked to resubmit the assignment.

Because the class is “hands on”, programming work will be emphasized. Reading assignments as mandatory homework are emphasized and will be tested by way of in-class unannounced quizzes.

Special notes: Due to the 596/605 overlap, students who enroll in the 596-2012/605-2013 sequence will have advanced requirements in the 605 session: 2 class presentations, a midterm computer project, and an advanced final computing project: the ability to enroll in this capacity is subject to limitations in overall class enrollment.

Office hours. In EBA 109; students may stop by or request to schedule a meeting (jotto@mail.sdsu.edu, 594-3505). Students can email “quick” questions; but if a question is along the lines of “my program doesn’t work”, I will help you debug your program in person; your presence is required at debugging sessions.

Required Texts.
Parallel Programming with MPI by Peter S. Pacheco – buy now (see amazon.com, for example).
Designing and Building Parallel Programs by Ian Foster – don’t buy (see http://www-unix.mcs.anl.gov/dbpp/).

Optional Texts.
CUDA by Example: An Introduction to General-Purpose GPU Programming by Jason Sanders and Edward Kandrot.

URLs.
Course web page (homework assignments, this course description, etc.): http://www.csrc.sdsu.edu/comp605

Additional References.
The C Programming Language by Kernighan and Ritchie
Managing Projects with make by Oram and Talbott

Windows ssh client.
Download and run putty-0.60-installer.exe from www.chiark.greenend.org.uk/~sgtatham/putty/download.html