Mathematics 5490: Parallel Computing II

Craig C. Douglas, mailto:cdougla6@uwyo.edu, http://www.mgnet.org/~douglas

Syllabus, Fall 2009

Online syllabus: Throughout the entire semester, this syllabus will be online at the URL

http://www.mgnet.org/~douglas/Classes/na-sc/2009f-notes/syllabus.pdf

Course Description: A second semester course on parallel computing. We will apply knowledge of parallel architectures, hardware accelerators, programming paradigms, communications methods, applications, algorithms, and how different scales of parallelism affect performance to design and implement an application in parallel on a tradition cluster and a nontraditional GP-GPU cluster.
Prerequisites: Permission of the instructor.

Classrooms: Ross Hall 247 (TR, 1:20-2:35)
Class web page: http://www.mgnet.org/~douglas/Classes/na-sc/2009f-index.html
TA: none
Office hours: Tuesday 12:15-1:15, Wednesday 9:30-10:30, and by appointment (send me email or call for an appointment). Please call 766-6580 before coming to Ross Hall 227.

Textbook: None
Suggested Readings:

- J. Dongarra, G. Fox, K. Kennedy, and L. Torczon, W. Gropp, I. Foster, A. White, The Sourcebook of Parallel Computing, Computer Architecture and Design series, *Morgan Kaufmann*, San Francisco, 2002. ISBN-10: 1558608710, ISBN-13: 978-1558608719.
- C. C. Douglas, G. Haase, U. Langer, A Tutorial on Elliptic PDE Solvers and their Parallelization, vol. 16, Software, Environments, and Tools (SET) series, *Society of Industrial and Applied Mathematics* (SIAM), Philadephia, 2003.
- W. Gropp, E. Lusk, and A. Skjellum, Using MPI, *MIT Press*, Cambridge, 1997.
- W. Gropp, E. Lusk, R. Thakur, Using MPI-2, *MIT Press*, Cambridge, 1999.
- G. E. Karniadakis and R. M. Kirby II, Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and Their Implementation, *Cambridge University Press*, 2003 (with a cdrom of software).

Homework: This is going to resemble a seminar course. I will assign topics and expect students to present lectures using the LCD projector. You can scan handwritten notes for the lectures.
Exams/Projects: There will not be any exams in this course. There will be at least one project that the whole class will work on cooperatively in the standard way all large scale parallel applications are developed. The students will divide the project(s) up into pieces that will allow individual students to apply their own strengths to.
Grading: 100% of the grade will come from the class lectures (50%) and the project (50%). You cannot pass this course by only doing one half, but not the other half.

Advice:

- It is a good idea to already know Matlab to prototype algorithms and at least one legacy programming language (e.g., C, C++, or Fortran). I will suggest that the class use C. The class may decide to use another language, however. This will be decided in class.
- Do not be late to class.
- Take notes in class. My lecture notes will not be online during the semester.

Cheating Policy: Getting caught cheating or plagiarizing will result in a failing grade and possibly much worse, including expulsion from the university and legal proceedings against you. I have zero tolerance for cheaters. I will enforce whatever the latest university policy is. When in doubt, ask me first. Life is too short to experience the penalties of getting caught cheating.