

ACMS60212/40212 Advanced Scientific Computing, Spring 2015

Instructor: Zhiliang Xu

Email: zxu2@nd.edu Office: Hayes-Healy 242 Phone: 631-3423

Class time and place: MWF 9:25am – 10:15am, Hayes Healy Center 125

Office hours: W 4:00pm – 5:00pm, HH242

Prerequisite: ACMS60690/ACMS40390 or equivalent. Significant experience in C/C++ programming and applications to science or engineering.

This course covers fundamental materials necessary to use high performance computing to support research in science and engineering. There is a special emphasis on algorithm development, computer implementation, and the application of these methods to specific problems in science, engineering.

Topics to be covered:

1. Review of C/C++ programming.
2. Parallel Computing, MPI basics
3. Parallel algorithms for implementing direct and iterative methods for solving system of linear equations
4. Sub-domain decomposition method for solving time-dependent partial differential equations on large domains.
5. Computing on GPUs
6. OpenMP basics
7. Monte Carlo and stochastic simulations
8. If time permits, mesh generation, adaptive mesh refinement and scientific visualization, and parallel implementation of finite element method for solving elliptic equations.

Grades: Course grades will be based on homework and projects. Homework and projects should be done individually or in small groups. Undergraduate and graduate students will have different assignments.

| | |
|---|-----|
| Homework: | 10% |
| 5 projects: | 60% |
| Final project including project presentation: | 30% |

Honor Code: As a member of the Notre Dame community, I will not tolerate academic dishonesty. All examinations, homework and computer projects are conducted under the Honor Code. You are encouraged to work together on the homework assignments and projects, but copying in any form or submitting work done by others as your own is a violation of the Honor Code.

Attendance: You are expected to attend every class including your assigned tutorials. Excessive absences may result in lowering your grade and even failing the course.

Textbooks:

1. A. Grama, A. Gupta, G. Karypis, V. Kumar, Introduction to Parallel Computing, ISBN-0-201-64865-2
2. G.E. Karniadakis, R.M. Kirby II, Parallel Scientific Computing in C++ and MPI, A Seamless Approach to Parallel Algorithms and Their Implementation, ISBN-9780521520805

References:

1. V. Eijkhout, E. Chow, R. van de Geijn, Introduction to High-Performance Scientific Computing by (Public draft)
2. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, second edition, 2002.
3. Iterative Methods for Linear and Nonlinear Equations by C.T. Kelley, SIAM 1995
4. J. Sanders, E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, ISBN-10: 0131387685

Getting Started

1. Requesting an Account

<https://crc.nd.edu/index.php/services/usersupport>

2. Accessing the machine

http://wiki.crc.nd.edu/wiki/index.php/CRC_Quick_Start_Guide

CRC provides the following front-end machines for compilation and job submission. Each machine is configured with identical software stacks.

crufe01.crc.nd.edu (4 16 core 2.4 GHz AMD Opteron processors with 256 GB RAM)

crufe02.crc.nd.edu (4 16 core 2.3 GHz AMD Opteron processors with 256 GB RAM)

crufeIB01.crc.nd.edu* (4 16 core 2.4 GHz AMD Opteron processors with 256 GB RAM)

You can securely log into the front-end machines (enabling X forwarding for GUI displays) using a ssh client e.g.:

On Linux/unix: > ssh -Y netid@crufe01.crc.nd.edu

On Windows: use Putty (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>)

3. The sample codes on CRC machine are available at: ~zxu2/Public/ACMS40212

4. Software and Development Environment

4.1 Interaction with machine through an Unix Shell

The C shell. This shell uses a command structure and syntax similar to the C programming language.

On Linux, the csh is a link to tcsh.

A system cshrc is executed, followed by the following files located in your home directory (in order of their execution). You may override settings in the system cshrc from within your .cshrc and .login.

.cshrc

.login (executed only at login)

.logout (executed only at logout)

4.2 Development Environment

The software environment on the front-end machines is managed with *modules*. You can easily modify your programming and/or application environment by simply loading and removing the required modules.

The most useful module commands are:

- module avail (view the complete module list)
- module load xyz (load the module xyz)
- module unload xyz (unload the module xyz)
- module list (list currently loaded modules)

We typically use “intel/12.1(default)” compiler and “mpich/3.0.4-intel(default)” for MPI. To setup both of them, use the following commands:

```
> module load intel/12.1
```

```
> module load mpich/3.0.4-intel
```

4.3 Debugging tools

gdb, ddd, totalview

4.4 Text Editor

vi, vim, emacs