Using the TeraGrid to Teach Scientific Computing

Frank Löffler, Gabrielle Allen, Werner Benger, Andrei Hutanu,
Shantenu Jha, Erik Schnetter

Louisiana State University, Baton Rouge, LA

Jul 20th 2011

Supported by NSF awards EPS-0701491, OCI 0947825 and PHY 0904015, as well as the TeraGrid allocation TG-SEE100004, the LONI allocation loni_cactus05 and the FutureGrid team (supported by NSF award 0910812)

Course Description

HPC for students

Scientific Computing

- Graduate Course at LSU
- Taught during Fall 2010
- Utilized TeraGrid resources
- Broad and practical introduction to scientific computing
- Enabling students to work in research groups using computing resources
- Offered through the Department of Computer Science
- Complements other courses, including *High Performance Computing: Models, Methods and Means*

Course Justification

- *Scientific computing* is increasingly taking its place as the third leg of scientific research.
- Computational advances in many fields of science, e.g.:
  - understanding climate change
  - designing new “smart” power grids
  - national security
- Traditionally taught methods of purely experimental or theoretical scientific research are not sufficient.

Future researchers need to have skills to

- enable them to contribute to projects that involve diverse software
- take part in significant collaborations between researchers from multiple disciplines
- utilize a broad understanding of computational science

Course Setting

Center for Computation & Technology (CCT)

- Louisiana State University
- On-campus research unit
- Includes over 30 faculty from some 13 different departments
- Working in numerous large-scale collaborative projects typically involving computation
- Projects usually involve
  - real world application problems
  - interdisciplinary approaches
  - collaboration between faculty
  - the use of high-end cyberinfrastructure
  - software development

Participants

Thirteen graduate students

- Six masters students (System Science)
- Seven doctoral candidates
  - Four computer scientists
  - Three civil engineers

Course Organization

Scientific Computing

- Set of five modules
- 25 lectures, 10 class-works, 3 projects, final exam
- Different instructor for each module
- Instructors drawn from faculty and PhD level researchers at CCT
- Motivation for teaching
  - opportunity to train and recruit students to join research projects
  - encourage collaboration and overlaps between different projects
  - for some staff: opportunity to gather teaching experience

Course Modules

A: Basic Skills

B: Networks and Data

C: Simulations & Application Frameworks

D: Scientific Visualization

E: Distributed Scientific Computing

Basic Skills - Motivation

Challange with incoming graduate students into our research groups:

- Wide variability in their basic computer and computational skills and experience
- Unless students had already been partially trained in other research groups:
  - Rarely have previous experience with HPC
  - Would not know about the existence of, or the use of the TeraGrid and other national resources
  - Rarely worked in large, distributed teams
  - Often didn't work on large software projects yet

Basic Skills - Overview

- Largest Module within course
- Covering fundamental topics important for all the other modules
- Spread within lectures of other modules
- Homework given, and reviewed, but not directly influencing grade
- Provides 20% of final exam questions

TeraGrid Access

TeraGrid

- Educational allocation: quick turn-around
- Student accounts requested through this allocation
- Quickly dealt with by TeraGrid staff, but difficult to do in advance
  - dynamic compisition of class in intial phase
  - not all necessary information available through standard class systems

LSU CENTER FOR COMPUTATION & TECHNOLOGY

Basic Skills - Lectures 1 - 4

1. Preliminaries
    - Establish common level between students
    - Overview of Unix/Linux systems, shells, basic SSH usage, text editors, basic compiling and linking, Make-system and simple visualization
    - Task of activating and testing TeraGrid accounts
2. Introduction to Numerical Methods
    - Overview of common numerical methods, e.g., root finding, interpolation, integration, differentiation, differential equations, random numbers and Monte-Carlo methods
    - Task of applying learned method on TeraGrid system
3. Vector Algebra, Basic Visualization Programming
    - vector definition, notation and algebra, unit and base vectors, vector components, vector products
    - practical: visualization programming using OpenGL/GLUT
4. Advanced Secure Shell Usage
    - definition of authentication and general authentication mechanisms, public-key cryptography
    - essential SSH usage examples on TeraGrid resources

Basic Skills - Lectures 5, 6 & 7

5. Best Coding Practices
   - software project planning
   - coding styles and good programming practices
6. Software Development, Revision Control
   - introduction to high-tech communication channels, issue trackers, and documentation practices and formats
   - specific focus on revision control systems
     - present common revision control systems
     - usage examples
7. Compiling, Debugging, Profiling
   - given towards end of course: students had some prior experience, however most only followed instructions without understanding
   - compiler sub-steps by examples, explaining object files, libraries and executables, filesystem placements and program loaders
   - bug prevention techniques, general debugging techniques, example *gdb* session, *gprof* example session

Simulations and Application Frameworks

Aim: explain the essential elements of modern simulation codes that are run on parallel supercomputers, especially TeraGrid systems.

Examples at LSU:

- modeling black holes
- predicting the effects of hurricanes
- optimizing oil and gas production from underground reservoirs

Often not understood by students:

- typical code structure
- scientific goals and needs
- hardware and technology limitations

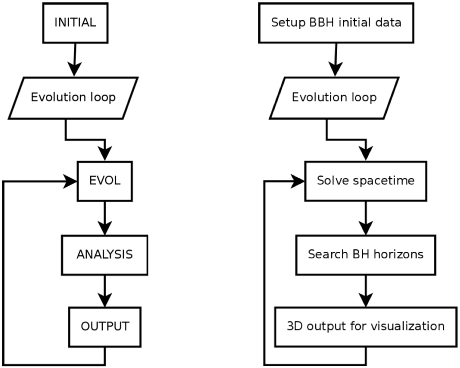
Simulations and Application Frameworks

Coverage:

- What is a simulation?
- Example of an initial value problem
- Typical work-flows on supercomputers, especially TeraGrid, such as batch processing, computing time allocations and login procedures
- Usage of Cactus code: assemble, configure, build and execute across many cores within TeraGrid
- Follow Einstein Toolkit tutorial to use physics example code on TeraGrid
- Visualization of simulation results

Simulations and Application Frameworks Example

Example scheduling tree



Simulations and Application Frameworks Example

TeraGrid™

www.loni.org

Allocations

- Need to ensure fair use of supercomputer, prevent individual users from monopolising it
- Typically, an *allocation process* decides who can use how much of a supercomputer's time during a year (similar to writing a grant proposal)
- 1 CPU hour costs about 5 cents (10 cents on Amazon ECC)
- With this metric, Queen Bee produces about \$270 worth of CPU time every hour

Thursday, September 9, 2010

LSU CENTER FOR COMPUTATION & TECHNOLOGY

Networks and Data

Aims:

- provide students with an understanding of the meaning and utility of networking in computing
- give them an introduction to basic practical methods of using high-speed networks for scientific computing
- introduce them to methods of dealing with large scientific data

Practical part:

- hands-on introduction to using high-speed networks on the TeraGrid
- introduction to *iperf* and *GridFTP*
- assignment to measure the network performance between various sites (Queen Bee, Longhorn, Ranger, Abe and Steele)

Networks and Data

- Introduction to basic networking concepts, such as simple network tools and core transport protocols
- Simple but widely used network applications, e.g., for bulk data transfer and video-conferencing
- Introduction of tools necessary for some TeraGrid sites, most notably *myproxy* and *gsissh*
- Introduction to middle-ware as the means to build distributed applications
- Presentation of high-level applications of networking such as distributed data management and distributed visualization

Networks and Data Assignment

Run `iperf` between TeraGrid sites and write a report on the network performance. The report should include:

- TCP and UPD speeds between at least three sites
- Optimizations by changing window and packet size, number of parallel streams and other methods
- Graphs to show results.
- Optionally include using the UDT library in the analysis

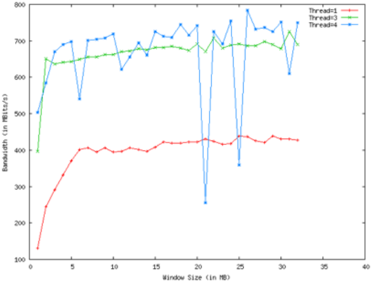


Image courtesy: S. Khurana and B. Bohara

Lessons Learned

Teaching *Scientific Computing* was

- a lot of work,
- a lot of fun,
- frustrating at times,
- rewarding in the end.

Lessons Learned

- TeraGrid provided an essential part of course
- provided students with experience using a real-world computational environment
- TeraGrid staff were very responsive in dealing with account management
- Some changes to account management could be made to better support classes
- Created awareness of the breadth of computational facilities available to academic researchers in the USA
- Provided students with the confidence to work in these environments
- We will do it again!