In last month’s issue, I surveyed efforts to determine the best practices in teaching Grid computing (see http://dsonline.computer.org/portal/pages/dsonline/2008/03/o3002edu.html). This month, I present a case study of my experiences in teaching Grid computing to PhD students over the past three years. In general, my experiences have confirmed the suppositions made by other teachers experimenting in teaching Grid computing.

Course description
Starting in 2002, a group of teachers in the Department of Computer Science at the Western University of Timișoara (Romania) conducted research projects specializing in Grid topics. Five years before, the university began a master’s program in artificial intelligence and distributed systems, attracting students from different parts of the country. Consequently, I started a new course called Grid Computing in October 2005. Since then, I have aimed this course at PhD students in areas related to distributed computing. You can find details and useful links to other courses at http://web.info.uvt.ro/~petcu/grid.html.

Course aim
The Grid Computing course provides a practically oriented introduction to current technologies, focusing on Grid technologies and standards, and a theoretical overview of the underlying issues that arise in supporting Grid-based e-infrastructures.

Skills to be developed
I planned this course to give students

- an understanding of the facilities a Grid-based e-infrastructure provides,
- an awareness of the current issues in Grid architecture and potential areas for improvement, and
- skills in using current Grid tools and technologies.

Prerequisites
This course requires a good understanding of the fundamental principles of distributed systems, such as students would obtain in a senior undergraduate course in computer networks and security and a first-year graduate course on distributed systems. Students should also be competent in Java programming, have experience with parallel and distributed algorithms, be familiar with Unix operating systems, and have done some work on databases at the level of detail covered in an undergraduate computing science curriculum.

Content
The module consists of 14 lectures (50 minutes twice weekly) and 14 laboratory sessions (50 minutes weekly). Additional time is required for reading and practical work for assignments.
The course content focuses on understanding Grid architectures, standards, and tools as well as specific topics related to resource management, security, and data management. It puts particular emphasis on reviewing the applications and tools currently available on Grid-based e-infrastructures.

**Labs**

Laboratory sessions let students gain practical experience with Grid computing systems and Grid-based application development. At the end of the course, students should be able to effectively manage and use the middleware and tools available in Grid environments.

In computer assignments, the students explore all the major components of Globus Toolkit 2 (GT2) and gLite and gain experience in Grid service development in Globus Toolkit 4. Mandatory assignments cover using certificates and job submission in GT2 and gLite, developing a Web service, developing a Web Services Resource Framework service, and developing a simple portal for job submission. The exercises concerning Grid services are similar to ones that Amy Apon and her colleagues describe elsewhere.¹

**Testbeds**

Three testbeds are available for exercises. One is a department platform that includes several personal computers currently used for research purposes. Globus Toolkit 4, Tomcat, Axis, and Java are installed on different Linux operating systems. The students have their own accounts and certificates from a local certificate authority. They can connect into the experimental Grid from other resources—for example, the ones from the university campus—and provide certificates for these resources. Students do classroom exercises and homework concerning Grid services and remote execution of specific simple applications on this network. They can also build, deploy, and test simple Grid services on this platform.

The second testbed is dedicated to the SCIEnce infrastructure (Symbolic Computation Infrastructure for Europe, www.symbolic.computation.org) and connects three European sites. Teachers supervise exercises on this platform, which students perform from unique student accounts with restricted rights. PhD students working on topics related to the SCIEnce project have their own accounts and can deploy new services. They can create experiments on porting parallel computing applications using a combination of distribution and parallelism. Middleware similar to the one deployed on the departmental platform is installed. This testbed is different, however, in that the e-infrastructure connects three high-performance multicore clusters that let students do experiments in a real-world distributed environment.

The third testbed is based on the SEE-Grid (South Eastern European Grid-Enabled eInfrastructure Development, www.see-grid.eu) and EGEE (Enabling Grids for E-SciencE, www.eu-egee.org; training events at www.egee.nesc.ac.uk/schedreg/index.cfm) e-infrastructure. For these two European e-infrastructure projects, the computer science department offers a productive site based on a PC cluster located in the same place as the multicore cluster I mentioned earlier. Students perform experiments with gLite during the interactive laboratories.

**Materials**

The textbook I developed for the course² is derived primarily from research papers, edited research books, and tutorials. I also found the IBM redbooks³⁻⁵ very useful, especially with regard to GT2 programming, MPI programming, and simple portals. I also intensively used the GT4 tutorial⁶ for programming Grid services in Java. Other course materials include papers, webpages of Grid projects or Grid-related courses, and specification documents.

The code provided by the IBM Redbooks and GT4 Tutorial and tested on the departmental testbed was available in a directory accessible from the students’ accounts.
Evaluation

At the end of the course, the students take a written exam testing their knowledge and familiarity with Grid computing concepts. This exam constitutes half of their grade.

The second half is related to an individual semester-long project in which students are required to port an application from a list of application fields into a GT4 environment by exposing legacy code as a Grid service. The application fields have varied each year I've taught the course, depending on the subject of the Grid-related research projects that were executed that year.

Some examples of Grid services wrapping some legacy codes are

- Maple, GAP (Groups, Algorithms, Programming, www.gap-system.org), KANT (Algebra and Number Theory, www.math.tu-berlin.de/~kant), and other computer algebra tools for symbolic computations for the SCiEnce project,
- GIMP (Gnu Image Manipulation Program, www.gimp.org) for image processing for the MedioGrid project (http://mediogrid.utcluj.ro),
- Gerris Flow Solver (http://gfs.sourceforge.net) for computational fluid dynamic for the project NanoSim (http://nanosim.ieat.ro), and

Some of the students presented their papers at international conferences.

Other complementary activities

Local training in the frame of an EGEE program for a larger audience (for example, Grid Training Days in Timișoara 06, www.info.uvt.ro/~petcu/griddays.html) and project meetings complete the students’ development as Grid users and application developers.

In the last two academic years, I have experimented with teaching Grid topics during the first-semester master’s studies distributed systems course. To introduce undergraduate students to Grids, I recommend a survey of GridCafé resources (http://gridcafe.web.cern.ch/gridcafe). In a recent paper, Barry Wilkinson and Clayton Ferner gave useful hints about lab assignments using portals and portlets that I will try to follow over the next few years.7

Ways to achieve the course aims

One of the most important Grid notions that students must understand is the virtual organization. A visual example of the Grid interconnectivity is a real help in explaining VOs. Inspecting MonALISA repositories (Monitoring Agents using a Large Integrated Services Architecture, http://monalisa.cacr.caltech.edu/monalisa__Repositories.htm) is useful in this sense. MonALISA can monitor several properties of Grid-based platforms, such as system information for computer nodes and clusters, network information (for example, traffic, flows, connectivity, and topology), and performance of applications, jobs, or services.

A difficult task during lectures is drawing borderlines between the Grid and fields such as

- utility computing,
- on-demand computing,
- virtualization,
- service-oriented computing and infrastructure,
- software as a service,
- cloud computing,
• autonomous computing,
• global computing,
• pervasive, ubiquitous, and mobile computing,
• peer-to-peer computing,
• volunteer computing,
• desktop Grids,
• service-oriented knowledge utility,
• cluster computing, and
• Internet computing.

For each of these fields, I must provide practical examples. This is a highly time-consuming task, but it results in a better understanding of what a Grid can provide for users.

An essential prerequisite for teaching Grid concepts using classroom exercises is a testbed. Real Grid experiments require several hardware resources; moreover, the complexity of installing and configuring Grid systems can obscure the aim of learning to use the Grid middleware and tools. (A good knowledge of Linux is required for implementing security features, identifying the location of various configuration files, or installing software packages.) Furthermore, the preparation of Grid services requiring many iterations is inappropriate in real production environments. Therefore, a small experimental environment should be available for preproduction and educational purposes.

Grid-related technologies have considerably different knowledge prerequisites than parallel computing. Although a Grid can be seen as a cluster of clusters, mastering its technologies requires a deep knowledge of computer security and network programming. Cluster-computing-specific skills in parallel algorithms or message passing programming are less important, but still recommended. Simple examples of the benefits of using a Grid-based rather than a cluster-based e-infrastructure for problem solving in different scientific fields are hard to provide for the classroom, and I intend to focus on this in future versions of the course.

It’s also important to consider the background knowledge of the students taking the course. Students who were accustomed to using integrated development environments to build and to launch their projects found the classical batch processing and the classical tools for remote access to their department accounts to be unpleasant and obsolete. To overcome this situation, I promoted using Eclipse tools for designing Web services, g-Eclipse (www.geclipse.eu) for gLite jobs, and the workflow editor GridNexus (www.gridnexus.org) for Grid services, as well as using a local simple portal (Grid Operating Center, http://ui01.info.uvt.ro:8080/ugoc) for submitting jobs. Moreover, Grid services registered by the students in the standard Tomcat container can be made visible through a service portal. This lets any Web-based client inspect the operations that a service exposes in its Web Services Description Language description and to launch automated-generated clients for selected operations.

A step that can be useful between developing a Grid-based application and deploying it in a real production environment is using Grid simulators (I plan to test this the next time I teach the course). I recommend using MONARC 2 (Models of Networked Analysis at Regional Centers, http://monarc.cacr.caltech.edu), a simulation framework built to provide a design and optimization tool for large-scale distributed computing systems.

The rapid changes in Grid computing and related topics are precipitating further development of this course’s curriculum. Learning by experimenting in real-world Grid environments is the key element in the success of such a course.
References


Dana Petcu is a professor and the director of the Computer Science Department at the Western University of Timișoara. She’s also the director of Research Institute e-Austria, Timișoara, Romania. Contact her at petcu@info.uvt.ro.

Cite this article:
Dana Petcu, “Teaching Grid Technologies to PhD Students, Part 2: Course Structure and Experiences,” *IEEE Distributed Systems Online*, vol. 9, no. 4, 2008, art. no. 0804-o4001.