IEEE distributed systems ONLINE
Expert-authored articles and resources

June 2006 (vol. 7, no. 6), art. no. 0606-o6003
1541-4922 © 2006
Published by the IEEE Computer Society

Education

Teaching Grid Computing in North Carolina: Part I

Barry Wilkinson • **University of North Carolina Charlotte**
Clayton Ferner • **University of North Carolina Wilmington**

Since 2004, North Carolina undergraduate students have had the opportunity to learn and apply grid computing techniques using collective resources from several local universities. Three grants funded this work—a US National Science Foundation grant for developing the course and broadcasting it across the state and two University of North Carolina Office of the President grants that provided funds for grid computing equipment for universities and support for students and faculty.

Most grid computing courses are graduate-level seminar or special-topics courses within a single department. Our course is unique both because it targets undergraduates and because several universities participated in forming a grid of computers (their students enrolled in the course collectively to use this grid).

Course details

We classroom-tested the 15-week course in fall 2004 and 2005. Both times, we featured state-of-the-art presentations from internationally known guest speakers. The participating universities included minority-serving universities, state universities, and private colleges.

Most students were junior or senior computer science students. The prerequisite was programming skills in Java and C, preferably on a Linux system. We expected all students to meet the prerequisite or quickly learn the languages and system tools. We didn't assign a course textbook because there wasn't a suitable one; however, we informed students of available grid computing books and important papers that were suitable for reference and reading. Apart from preparing new lecture materials specifically for undergraduate students, we produced extensive documentation for both a coherent set of programming assignments and for grid software installation. We also made everything available online to help others create their own grid courses.

Grid computing uses geographically distributed computers connected on the Internet for high-performance computing and resource sharing. It often involves computers from multiple organizations, crosses organizational boundaries, and enables the creation of distributed teams (so-called "virtual organizations"). Therefore, to teach grid computing properly, we first needed a distributed operational grid of computers for students to access. Our course uniquely combines distance-learning tools with grid computing. Students from many universities enroll and become the virtual organization together with supporting faculty from their home institutions. Integrating with this virtual organization is the physical grid of computers provided at major sites for students.

We took advantage of the existing statewide North Carolina Research and Education Network to present the lectures to geographically distributed sites. NCREN, an instructional telecommunications network, has interconnected universities, medical centers, research institutions, and graduate centers in North Carolina since 1985. It provides face-to-face video and audio communications, employing a combination of microwave links and long-distance digital fiber-optic lines. Each site has "teleclassroom" facilities where students have microphones and can interact and participate in discussions with the instructor via video cameras. Each teleclassroom also has a camera operator who records all lectures, so students can view lectures again later or catch up on any missed lectures.

Testing the course

The fall 2004 grid computing course originated from Western Carolina University (WCU) and broadcast on the NCREN network simultaneously to the University of North Carolina Wilmington, UNC Asheville, Appalachian State University, North Carolina State University, UNC Greensboro, Elon University (a private institution), and North Carolina Central University. Some lectures originated from UNC Wilmington.

More than 40 students enrolled in the course, including two from Cape Fear Community College attending classes at UNC Wilmington. Students received course credit at their home institutions. Several faculty members attended. Near the end of the semester, students heard presentations from several guest speakers, including Daniel Reed of NC State University and Wolfgang Gentzsch of the Microelectronics Center of North Carolina. The Microelectronics Center's Chuck Kelser discussed legal issues. Students viewed a taped presentation by Ian Foster of the University of Chicago and the Argonne National Laboratory, the so-called father of grid computing. An NC State University graduate student also presented his work on grid computing.

The fall 2005 grid course originated from UNC Charlotte and broadcast to Appalachian State University, Elon University, NC State University, UNC Asheville, UNC Chapel Hill, UNC Pembroke, UNC Wilmington, Western Carolina University, Winston-Salem State University, Lenoir Rhyne College, and Wake Technical Community College. Again, some lectures originated from UNC Wilmington. Figure 1 shows a map of the universities in the UNC system, identifying the fall 2005 participating sites.

Figure 1. Sites participating in the fall 2005 grid computing course. Pink dots indicate participating UNC campuses; yellow dots indicate private institutions.

Thirty-two students enrolled in the course, several faculty members attended, and invited guest speakers participated near the end of the semester. Jim Jokl from the University of Virginia, Art Vandenberg from Georgia State University, and Mary Fran Yafchak from the Southeastern Universities Research Association, gave a team presentation describing the SURAGrid. The Renaissance Computing Institute's Lavanya Ramakrishnan described the institute's work including the Bioportal and ocean and atmospheric modeling using the grid. Genesis Molecular's Jeff Schmitt discussed creating a large unique grid of computers across a school district for molecular modeling. WCU's Mark Holliday presented his "end-to-end" demonstration, describing all the underlying elements of job submission and illustrating them with a live demonstration. His student, James Ruff, also gave a presentation on using the Sun Grid Engine scheduler, which was used in a course assignment.

Setting up the grid of computers across the state was a major undertaking and required the assistance of many people. In fall 2004, computer clusters were set up with grid software at three major sites (WCU, UNC Wilmington, and NC State University) where many students were registered (roughly 13 at each site). At sites with only one or two students, students received accounts on WCU computers. In fall 2004, we used Globus 3.2 and the Condor/Condor-G scheduler. In fall 2005, computer clusters were set up with grid software at six sites (Appalachian State University, UNC Asheville, UNC Charlotte, UNC Wilmington, NC State University, and WCU), and they used Globus 4.0 with the Sun Grid Engine scheduler. We installed a workflow editor called Grid-Nexus and used it later in the course (both in fall 2004 and fall 2005).

Each major site also maintained a certificate authority. In fall 2004, we preinstalled all the necessary files for each student—that is, a signed certificate and grid-map entries (the way Globus handles access control). In fall 2005, we changed the certificate-creating process to make it more realistic to a production grid. A new production grid user often must ask the administrator for permission to join and then follow the instructions to set up the system. Similarly, on the first day of classes in fall 2005, each student created a certificate request to the CA system administrator using the Globus grid-cert-request command. Upon receiving the request, the administrator placed the signed certificate in the student's .globus directory and emailed a confirmation to the student.

The next part of our article will run in *IEEE Distributed Systems Online*'s July 2006 Education department. There, we discuss both a grid computing workflow editor developed at UNC Wilmington and experiences from the course. For more details on the course and its assignments, see www.cs.uncc.edu/~abw/ITCS4010F05.

Barry Wilkinson is a professor in the computer science department at the University of North Carolina Charlotte. Contact him at abw@uncc.edu.

Clayton Ferner is an associate professor in the computer science department at the University of North Carolina Wilmington. Contact him at cferner@uncw.edu.

Related Links

- DS Online's Web Systems Community Education Page
- "Cluster Computing and Grid 2005 Works in Progress," IEEE DS Online
- "Dependability in Grids," IEEE DS Online
- "Open Source in the Classroom," IEEE DS Online

Cite this article:
Barry Wilkinson and Clayton Ferner, "Teaching Grid Computing in North Carolina: Part I," *IEEE Distributed Systems Online*, vol. 7, no. 6, 2006, art. no. 0606-o6003.

July 2006 (vol. 7, no. 7), art. no. 0607-o7003
1541-4922
Published by the IEEE Computer Society

Teaching Grid Computing in North Carolina: Part II

Barry Wilkinson, *University of North Carolina, Charlotte*
Clayton Ferner, *University of North Carolina, Wilmington*

North Carolina students studied grid computing in a course taught across multiple universities in fall 2004 and fall 2005. This department describes the student assignments and the instructors' general experiences in offering the course.

In Part I (http://doi.ieeecomputersociety.org/10.1109/MDSO.2006.41), we described a grid computing course taught across multiple North Carolina universities using the North Carolina Research and Education Network (NCREN) televideo network. We also described the grid infrastructure and distance-learning environment. Here, we describe the student assignments and our general experiences in offering the course.

Student assignments and evaluations

The coursework consisted of a suite of five assignments developed by members working on the course. We used WebCT to submit programming assignments and quizzes. The coursework began with Web service assignments because today's grid computing software infrastructure is aligned with Web services. Then, we asked the students to develop, deploy, and test a grid service. The next assignments involved job submission using Globus with a scheduler (either Condor or Sun Grid Engine). Finally, the students put it all together in a final assignment where they each created a workflow.

The workflow assignment uses GridNexus (http://www.gridnexus.org), a workflow editor that researchers at UNC Wilmington developed. Figure 1 shows a chemistry example of a GridNexus-created workflow that involves several steps for processing a molecule. The actual workflows the students created are simpler. Instructors asked students to construct a workflow that uses Web services and grid services from their previous assignments. The students started the Globus container and verified that their services were working. Then, they configured a generic Web service module in GridNexus that communicated with their services. Finally, they linked their services' inputs and outputs into a more complicated workflow.

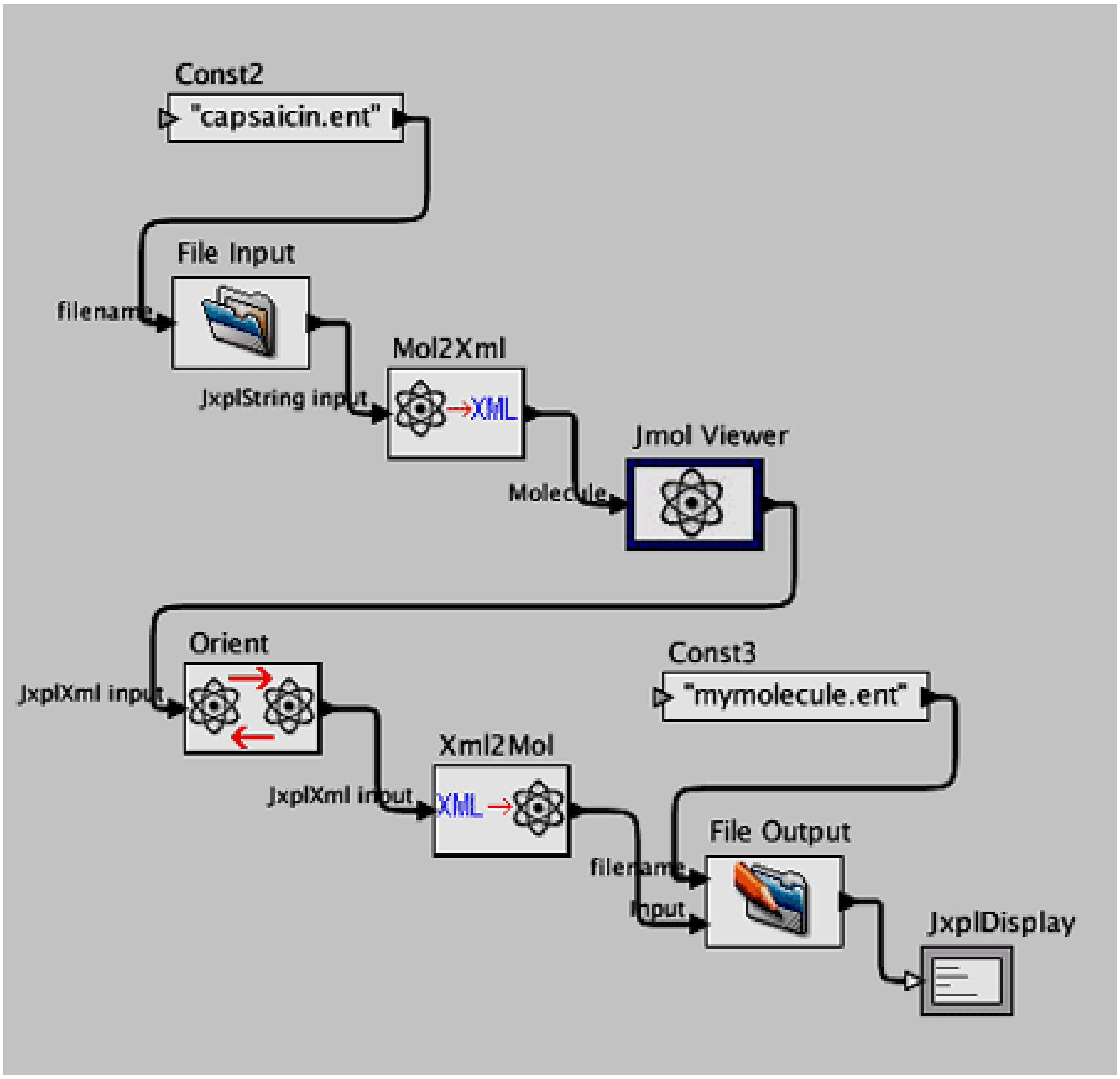

Figure 1. A workflow created in GridNexus.

Figure 2 shows a workflow similar to those the students created. This workflow's modules make calls to either grid services or Web services that the students created in previous assignments. This assignment shows the students how a workflow editor can help build more complex scenarios in which one grid service's output is another's input.

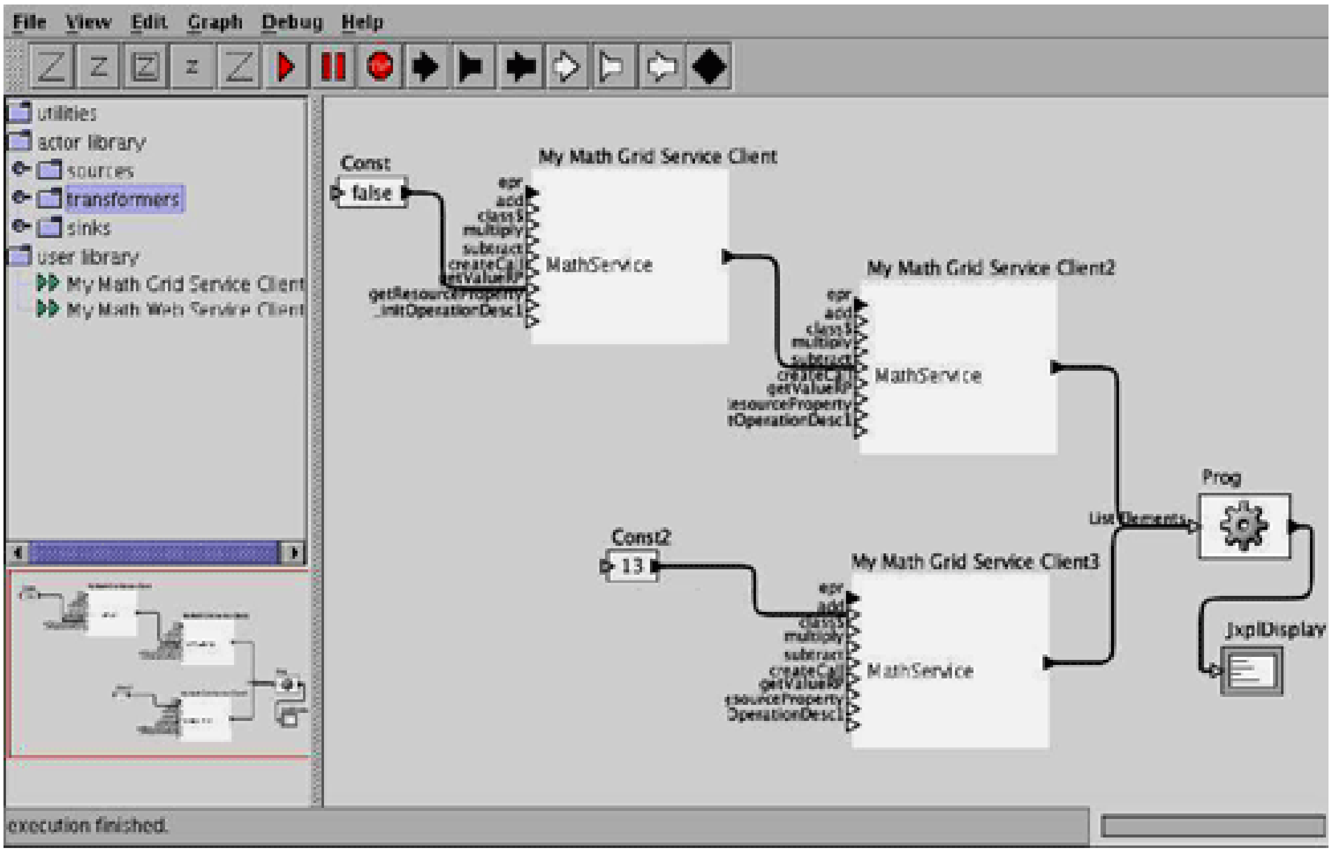

Figure 2. A workflow example using GridNexus, similar to the one assigned in the course.

A workflow editor can invoke several services and help pipeline dependent data. In the first few assignments, the students work with detailed, obscure commands. By the end of the course, they're thinking in terms of linking services to build more complex functionality.

We evaluated the course in several ways. First, we conducted traditional end-of-course student evaluations. We performed additional evaluations as part of evaluating the larger funded grid computing projects within which we had developed this course. Student feedback was generally very favorable. Students found the programming assignments relatively straightforward—perhaps too easy. In fall 2005, we provided detailed assignment write-ups (15 to 20 pages) that led students through a series of basic tasks, culminating in a final task. It might have been better not to provide such detail but instead let the students find the correct way to do each task.

Course experiences

We installed the grid computing software before the class started so that the course would run smoothly. We had little time to fix problems afterward, and problems delayed students from completing their assignments quickly. Occasionally, we had to assign students to a different grid site, which both caused delays and disrupted due dates. In the second course offering, each assignment had three dates: the date assigned, a deadline to report any system problems, and the due date. Typically, the deadline for reporting system problems was a few days after the date assigned, which meant that students had to begin the assignment immediately. The due date was typically one week later. That approach helped identify problems quickly.

Using NCREN to provide the grid course to many sites simultaneously with high-quality video was a great asset. Still, engaging in two-way dialog with so many sites was difficult. To increase student engagement, the fall 2005 course added brief student presentations. Each student had to find an interesting grid application (for example, TeraGrid), present it to the entire class using a few PowerPoint slides, and answer questions from the class.

This course was offered at each institution as a special-topics course, and it generated significant enrollment interest at many universities. However, we'll delay the next offering until at least spring 2007 to reflect on our experiences with the course.

Most students graduated within a year of taking the course. Some continued with grid computing. For example, in 2005, one student from the fall 2004 course at Elon University attended a one-week intensive summer workshop program in grid computing and its application in scientific data analysis. The University of Texas at Brownsville and Texas Southmost College, the International Virtual Data Grid Laboratory, and the Open Science Grid sponsored the program. A student from the fall 2005 course entered UNC Charlotte as an early-entry MS student and joined the Charlotte VisualGrid research group in 2006. That student and others are implementing a multi-institution research grid for visualization and environmental research. Several UNC Wilmington students have been working on a project to build chemistry and bioinformatics workflows for scientists. Several workflows have been used in chemistry classes.

This grid computing course's challenges are similar to those we face in creating and using any grid. First, configuring the machines and the grid software was a formidable task involving many expert system administrators. Setting up the users with accounts, certificates, and authorization took a week or two. Getting collaboration to a productive level was difficult in such a distributed environment. Finally, overcoming interorganization politics was difficult. Although we'll strive to overcome these challenges in the future, they were a learning experience for students. By offering such courses despite their difficulties, we'll develop future grid workers who will strive to overcome grid computing's challenges. For more information about the course and assignments, see http://www.cs.uncc.edu/~abw/ITCS4010F05.

Acknowledgments

This project required the cooperation of many people at many universities to establish the grid and coordinate the activities. We're grateful to Mark Holliday and James Ruff at Western Carolina University, who developed several course assignments and supported the course activities extensively. We're also grateful to faculty at each site—including Mladen Vouk and Gary Howell at NC State University, Dean Brock at UNC Asheville, and Joel Hollingworth and David Powell at Elon University—for setting up local grid nodes and being faculty contacts for students. Support for this work came from the National Science Foundation (DUE #0410667/0533334) and the University of North Carolina Office of the President.

Barry Wilkinson is a professor in the computer science department at the University of North Carolina Charlotte. Contact him at abw@uncc.edu.

Clayton Ferner is an associate professor in the computer science department at the University of North Carolina Wilmington. Contact him at cferner@uncw.edu.

Related Links

- DS Online's Web Systems Community Education Page cms:/dsonline/topics/was/education.xml
- "Teaching Grid Computing in North Carolina: Part I," IEEE DS Online, cms:/dsonline/2006/06/o6003.xml
- "Dependability in Grids," IEEE DS Online, http://csdl2.computer.org/comp/mags/ds/2005/12/oz001.pdf
- "Cluster Computing and Grid 2005 Works in Progress," IEEE DS Online, http://csdl2.computer.org/comp/mags/ds/2005/10/ox002.pdf
- "Open Source in the Classroom," IEEE DS Online, http://doi.

ieeecomputersociety.org/10.1109/MDSO.2005.32

Cite this article: Barry Wilkinson and Clayton Ferner, "Teaching Grid Computing in North Carolina: Part II," ,*IEEE Distributed Systems Online*, vol. 7, no. 7, 2006, art. no. 0607-o7003.