

IEEE DS Online Exclusive Content

Education

Teaching Grid Computing in North Carolina: Part II

Barry Wilkinson • **University of North Carolina Charlotte**

Clayton Ferner • **University of North Carolina Wilmington**

In [Part I](#), 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](#), 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.

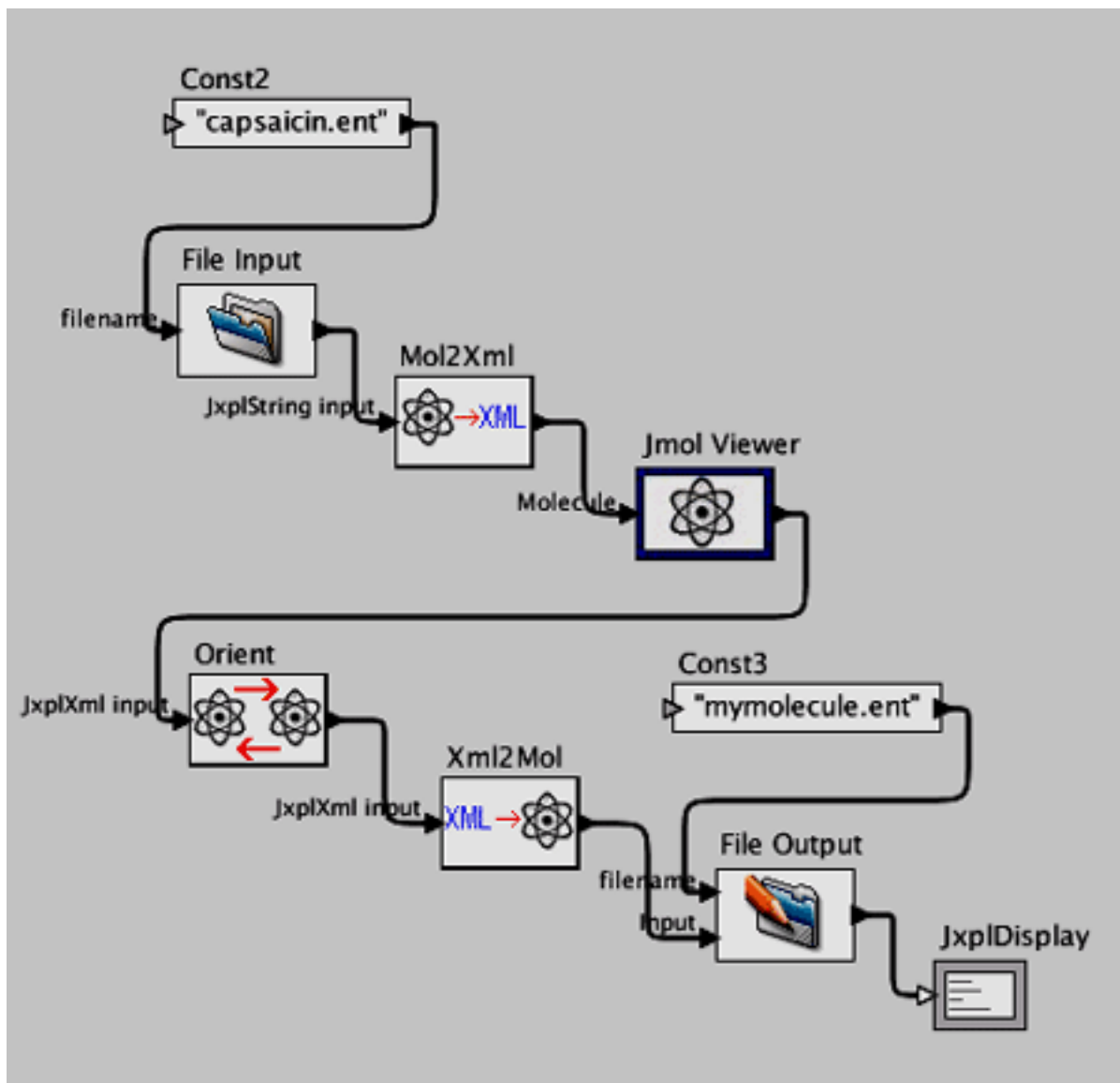


Figure 1. A workflow created in GridNexus.

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.

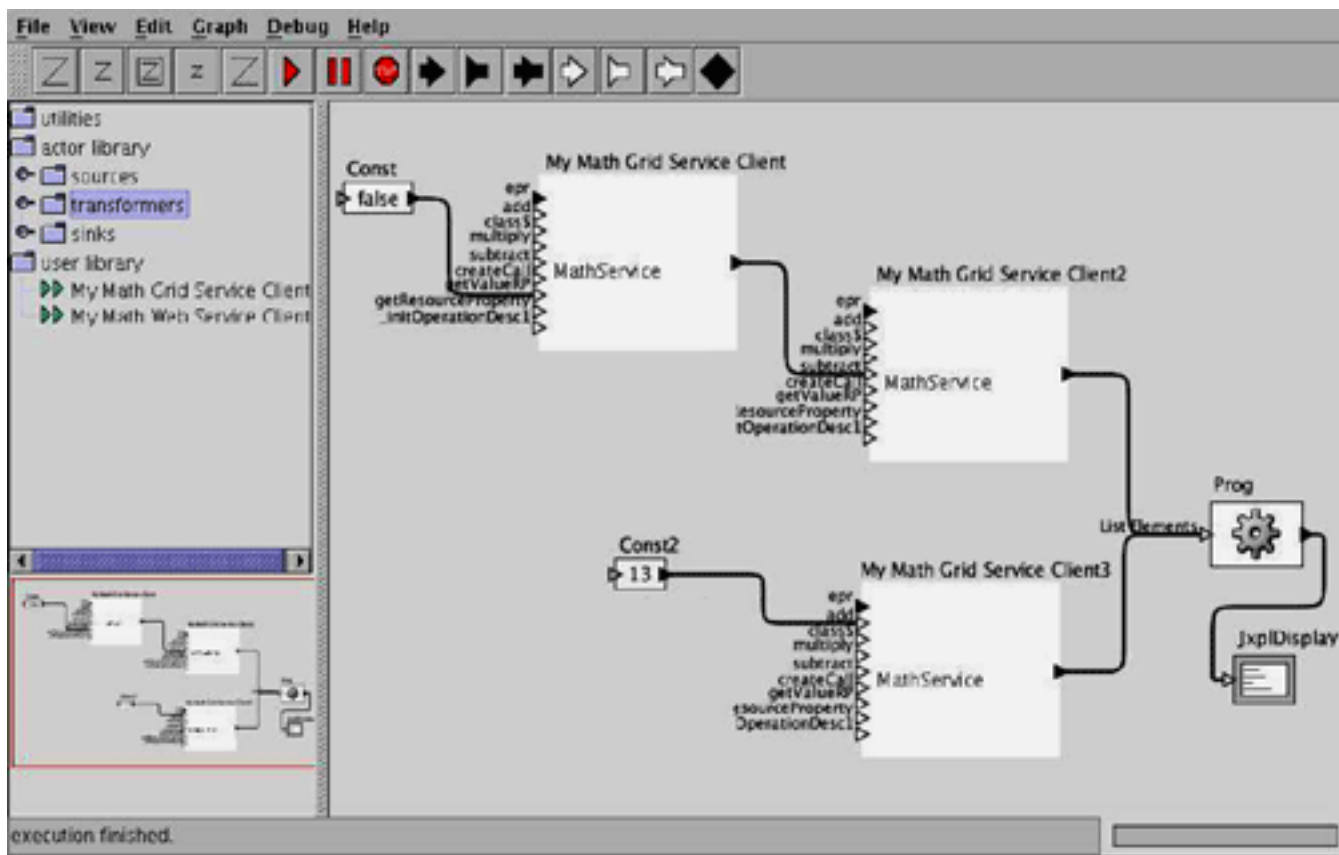


Figure 2. A workflow example using GridNexus, similar to the one assigned in the course.

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 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](#)

- "Teaching Grid Computing in North Carolina: Part I," IEEE DS Online
- "Dependability in Grids," IEEE DS Online
- "Cluster Computing and Grid 2005 Works in Progress," 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 II," *IEEE Distributed Systems Online*, vol. 7, no. 7, 2006, art. no. 0607-o7003.



dsonline.computer.org