2010 International Conference of Frontiers in Education: Computer Science and Computer Engineering (FECS'10)
Las Vegas, Nevada, USA, July 12-15, 2010.

Constructing Distributed Computations in an Undergraduate Grid Computing Course

Clayton Ferner[1] **and Barry Wilkinson**[2]
[1] Department of Computer Science, University of North Carolina Wilmington, Wilmington, NC USA
[2] Department of Computer Science, University of North Carolina Charlotte, Charlotte, NC USA

Abstract - *This paper describes our efforts to design assignments to contrast three approaches for constructing distributed computations in an undergraduate grid computing course, namely using a production-style grid portal, using the command line, and using a grid computing workflow editor called GridNexus. Survey instruments were used to collect results for a comparative analysis. We report our findings from the surveys.*

Keywords: Grid Computing, Workflow Editor, Undergraduate Education

1 Introduction

Grid computing uses geographical distributed computers for high performance computing and resource sharing. We have taught grid computing to undergraduate students since 2004 using geographically distributed computers and the North Carolina Research and Education televideo network (NCREN) connecting universities across North Carolina [1,2,3]. In Fall 2008, we redesigned our course to contrast the differences between interfaces used to work with a grid. The three interfaces were using a command line, using a portal, and using a workflow editor. The course was taken by students at five universities (University of North Carolina Charlotte, University of North Carolina Wilmington, University of North Carolina Asheville, University of North Carolina Greensboro and North Carolina A & T University).

2 Related Work

Apart from our grid course, others have taught grid computing to undergraduates during the same period, most notably Apon and Mache [4,5,6], and Ramamurthy and Jayaraman [7]. However, our course has a number of unique features. Our course uses geographically distributed computing resources and geographically distributed students with a televideo network to broadcast the course across the state. It is also co-taught from two universities on this network. Our course begins with a production-style grid portal and then delves into command-line actions. It also uses the GridNexus workflow editor for constructing distributed computations. We are not aware on these combined features being used elsewhere.

3 Design Of Assignments

We had two goals we wanted the students to reach in this offering of the grid course:

1) To contrast the different interfaces (portal, command-line, and workflow editor) used to work with grids; and

2) To understand the steps necessary to work with multiple applications using multiple resources at multiple sites on the grid where data needs to flow across the grid.

Understanding the differences in interfaces is not simply about being able to work with them or being able to choose the best one. The main reason for contrasting these interfaces is to gain an understanding of what they do and the messy details that are hidden from the user.

The second goal is to bring the students to an understanding of the main reason for the existence of grids. Simply executing applications on remote machines is not enough reason to justify the creation of a grid. We have had the ability to compute in this model for many decades. The main contribution of grid technology is the concept of virtual organizations and the sharing of resources. So to better emphasize this point we want the students to accomplish the steps of 2) above as well as to contrast the ease or difficulty in doing that with the three interfaces.

3.1 Assignment 1 – Portal

The first assignment required the students to interface with the grid using a portal. One of the reasons why we chose to start with the portal assignment was so that we could have students request their credentials early in the semester and have time to sign certificates and put them in place.

After credentials were created and placed on the MyProxy server, the students were asked to acquire a new proxy, learn to transfer files to and from their local machine and the server, and run a program in batch mode on the server. After completing those tasks, the students had to write a Java™ program that would integrate a piecewise function using trapezoids. The piecewise function represented the area of the flower bed. After verifying their program worked successfully, they had to upload the program to the server and run it using the portal.

3.2 Assignment 2 – Command Line

The second assignment required the students to interface with the grid using the command line. The students were asked to request credentials all over again, but using the commands that come with the Globus toolkit. After receiving their new certificate, they were required to run programs on the server in batch mode using the Globus command “globusrun-ws”. They also had to submit a request to run a job from one grid site to another as well as transfer the input and output files using the Globus command “globus-url-copy”.

After the students accomplished these simple tasks, they then had to write, compile, and execute the second half of a multi-step application. The first half is the computing of the area of a flower bed using integration by trapezoids from assignment one. The second half was a calculation of the volume and cost of mulch to cover the bed. Although this is a trivial program, the point of the assignment was to make use of multiple remote resources on the grid that requires file staging. They had to transfer the program to the remote machine, run it by submitting the request from one grid resource to run on another and handle the file transfers.

3.3 Assignment 3 – Workflow Editor

The third assignment required the students to interface with the grid using a graphical workflow editor called GridNexus. They first had to install the software and perform the necessary steps to set it up correctly. One of those setup steps was creating a new proxy and downloading it to their local machine. Since proxies usually expire in a short length of time (such as 24 hours), they may have to repeat that process, which leads to confusion.

The students were asked to run their two programs on the two different servers at the two different grid locations using the workflow editor. After doing this successfully, they then had to create and execute a workflow that would:

1) Run the integration program on the first server at the first site.

2) Transfer the intermediate results to the other server at the other site.

3) Run the second program on the second server.

4) Transfer the final result back to the first server.

3.4 Assignment 4 – WSRF Service

In the fourth assignment, the students were asked to create a stateful WSRF web service that would perform a simple math operation. This assignment was done using the command line. The students install the Globus core on their own machines for this assignment. The reason is to avoid interference between students. Even though the Globus toolkit allows students to start their own container on separate ports, all student services are deployed in everyone's containers. This means that each service has to be uniquely named, which is easily overcome. However, the real problem is that if one student makes a mistake that prevents the container from starting, then all of the students will also be unable to start their containers. Furthermore, so much repeated deploying/ undeploying services and starting/stopping containers tends to bring the server down frequently. It is a much better solution to have the students install the Globus toolkit core on their own machines and work locally.

After being able to install and run the Globus core, then students had to deploy their own server, start the container, and run their client program, all from a command line interface.

3.5 Assignment 5 – Workflow Client to a WSRF Service

In the fifth assignment, the students had to create GridNexus workflows that could make use of WSRF Services. We provided the WSRF services on the grid servers, so that students could still do this assignment even if they had trouble with the last assignment. First, the students had to create a workflow that interacted with a WSRF service that performed the simple math operations of assignment 4. After successfully performing this step, they had to create a workflow that performed the same multi-step processing as in Assignment 3, except that the applications were WSRF services, which we provided, instead of executable programs. An example of this workflow is shown in figure 1. The white boxes are the generic web service clients (modules) of GridNexus after they have been configured to work with particular WSRF services. The input ports (left side of the boxes) are the operations the web services provide. The ports on the right side are the outputs of the web services. The transfer of data between machines simply requires a connection of the output of one module in the workflow to the input of another module. This multi-step workflow using resources at different locations is very tedious using the command line.

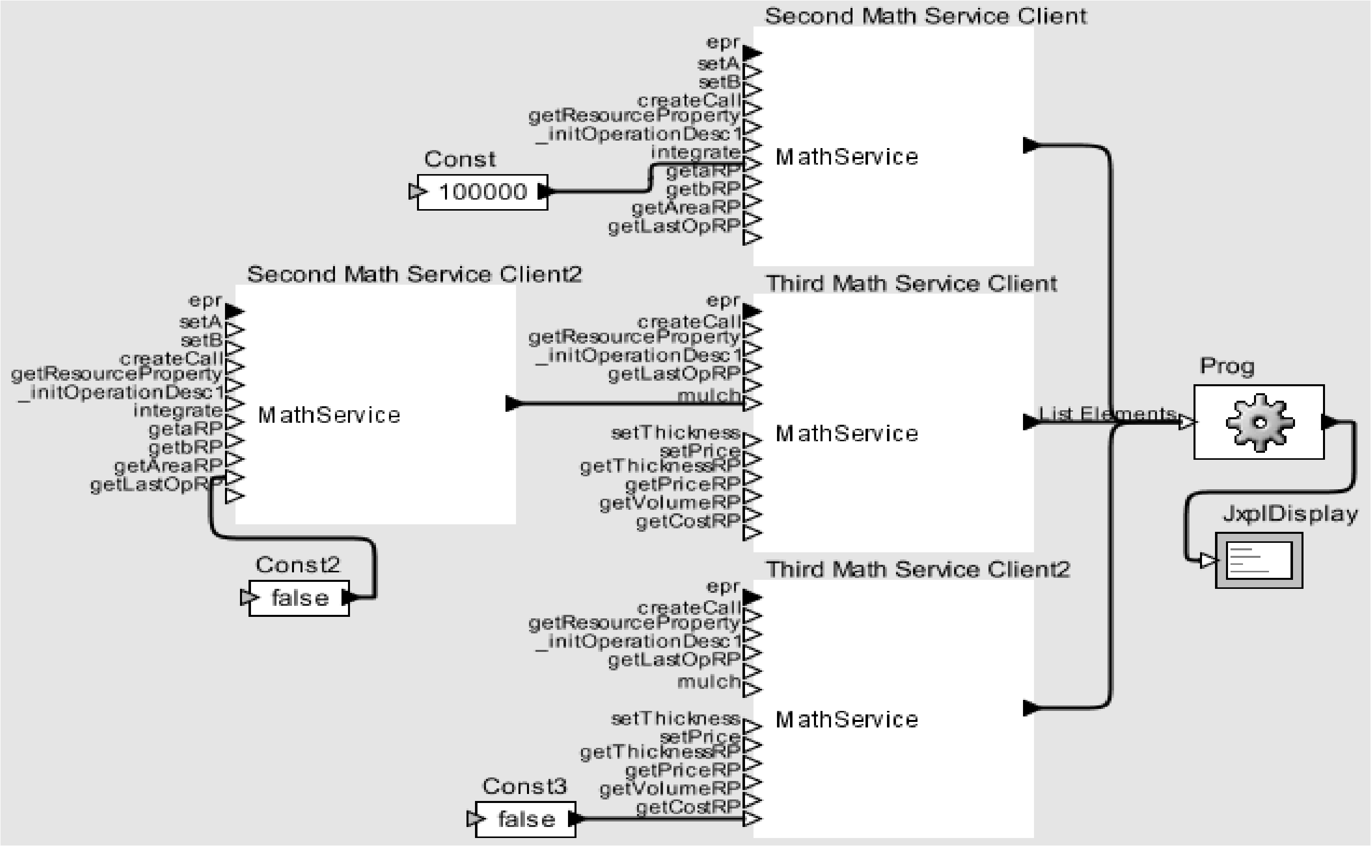

Figure 1. Example of Workflow Solution for Assignment 5.

4 Design Of Surveys

We administered two surveys to the class during the semester. The first came immediately after assignment 3 and was designed to contrast the three methods of executing programs on a grid. The second survey came immediately after assignment 5 and was designed to contrast the differences between using the command line and a workflow editor to work with WSRF services. We did not include the use of the Portal as an interface to WSRF services because it requires one to create a new portlet to do this. The creation of portlet is a topic that is not covered until very late in the semester.

Each survey consisted of Likert-style questions (from “strongly agree” to “strongly disagree”) about the assignments and methods used. The surveys also included free-form questions asking students to compare the different methods. Finally, the surveys included free-form questions asking the students how they would accomplish a multi-step workflow using resources at different physically locations on the grid with data flowing between sites.

5 Results

5.1 Survey I

Survey I was administered to 39 participants at the five universities, with 27 of them responding (69%). Table 1 shows results of the Likert-style questions of the first survey. Surprisingly, when asked about the difficulty of working with a portal, the command line, and GridNexus, they responded that working with GridNexus was more difficult than the other two. The comments given in the free-form questions comparing the three methods gives a clue as to why this was the case. Figure 2 gives a summary of the free-form responses comparing the methods. Several of the written comments by the students indicated that the setup of GridNexus was difficult. Since GridNexus runs on the local machine (usually a Windows machine), there is more setup required to make it work within a grid environment. Portal and command-line interfaces require logging into a remote machine, where the setup was already done for the students. After discovering the reason for the responses, the responses are no longer surprising. However, it does mean that some effort needs to be put into the setup of GridNexus to make it easier for grid novices.

Table 1. Results from Survey I

Statement	N	Mean	SD
Rate the difficulty of using portal.	25	1.52	1.36
Rate the difficulty of using command line.	24	1.50	1.18
Rate the difficulty of using GridNexus.	24	2.21	1.41

Scale: Very Easy (0)—Very Difficult (5)

Benefits

- "Portal was easier to use, did not require logging into individual machines, and was more relevant to real world situations"
- "Command line allowed for better visualization of what was happening on the Grid, was better at reporting errors, and was more stable"
- "Portal was easier for performing detailed processes, taught you how to create workflows by hand, and required no changes to the computer being used"
- "GridNexus was a good visual representation of the workflow process and was more fun and less random that portal"
- "Command line was better for smaller jobs, had fewer quirks, and was easier to learn and use"
- "GridNexus had a better visual representation of the workflow, allowed you to execute different programs together, and was more fun"

Drawbacks

- "Portal is unstable, hid what was really happening, and execution of tasks was random. "
- "Command line is visually distracting, harder to execute the correct command, and required more knowledge of Linux/Unix environments. "
- "Portal is less stable, could not perform workflows, and the execution of the echo command was random "
- "GridNexus required initial setup that was inconvenient, had many quirks, and required more detailed knowledge to use"
- "Command line required more background knowledge and was tedious to use"

Figure 2. Summary of student comments made comparing the interfaces.

The survey also asked the students to give free form answers to two questions:

1) "How would you use the portal to accomplish the same thing you did in Task 3 of Assignment 3, which involved two-part processing on two different servers including the file transfers?"; and

2) "How would you use the command line to accomplish the same thing you did in Task 3 of Assignment 3, which involved two-part processing on two different servers including the file transfers?"

Students seemed to have difficulty with these questions. There were only 13 responses on the first survey to these questions and only 5 were correct. Figures 3 and 4 give sample correct and incorrect answers to these two questions. The responses to these questions indicate one of two things:

1) Either the students failed to understand the questions or

2) Students did not understand the course concepts to a level where application to a different situation was possible.

Correct	"It would have to be a manual process. Execute job 1, use the file transfer portlet to transfer the file between hosts, then execute job 2. "
Incorrect	"I would submit a job on a computer on the Grid and have its output be the input of another job that I would schedule on the same machine. "

Figure 3. Sample answers to the question, "How would you use the portal to accomplish the same thing you did in Task 3 of Assignment 3, which involved two-part processing on two different servers including the file transfers?"

Correct	"I would create command line arguments to submit and run the jobs on the machines I needed them to run on. Then I would create command line arguments to transfer the files to the appropriate machine and finally submit and run the next job. "
Incorrect	"Using command line first create job description file with appropriate parameters. Then submit file using globusrun-ws -submit -F hostname -f RSLfile name, where hostname is name of the host here you want to submit the file. "

Figure 4. Sample answers to the question, "How would you use the command line to accomplish the same thing you did in Task 3 of Assignment 3, which involved two-part processing on two different servers including the file transfers?"

5.2 Survey II

Survey II was administered to 39 participants with 14 responding (36%). This survey focused on comparing command line versus GridNexus as interfaces in working with WSRF Services. The results of the Likert-style questions are provided in Table 2 (two students did not answer this question). The results of survey II support the results of survey I. The setup of GridNexus should have been done already, but still required creating and downloading a proxy. Students again reported their concerns for the setup. Sample responses to the question about how one would accomplish the multi-step processing using the command-line are given in Figure 5. Once again, less than 50% provide appropriate and correct answers indicating that many of the students failed to understand the question or failed to understand the concepts we were trying to teach.

Table 2. Results from Survey II

Statement	N	Mean	SD
Rate the difficulty of working with WSRF services using command line.	12	2.83	0.83
Rate the difficulty of working with WSRF services using GridNexus.	12	2.33	0.89

Scale: Very Easy (0)—Very Difficult (5)

Correct	"I have to run flower bed in 1 server, then record to the file, and transfer it to another server, and then use the file the run the mulch cost. "
Incorrect	"I would probably do it with a shell script. "

Figure 5. Sample answers to the question, "Describe how you would use the command line to accomplish the same thing you did in Task 4 of Assignment 5, which involved two-part processing on two different servers where the output of one service was the input for the other."

5.3 Discussion

The response rate for survey I was 69%. The response rate dropped to 36% for survey II. We do not have an explanation for this drop. We speculate that the survey, with the free-form questions, required too much thought to fill out. Even though the students were included in a raffle for a gift certificate for participating, this did not seem to be sufficient motivation.

The scores on the assignments were quite high. Most students received scores above 90%. The lack of understanding of concepts could be the result of students focusing too much on completion of the assignment steps, rather than problem solving. Another explanation is that the three assignments were not quite comparing apples to apples. The first two assignments were done to complete the two halves of the multi-step processing. It was not until the third assignment that students were asked to put the entire multi-step processing with file transfer together into one assignment. Either way, it is clear that the level of understanding is not what we hoped for and thus need to improve students' focus on the concepts rather than the details. We plan to change the first three assignments in the future so that students are performing the multi-step processing from the beginning.

6 Conclusions

This paper describes assignments that use three approaches for constructing distributed computations in an undergraduate grid computing course, namely using a production-style grid portal, using the command line, and using a grid computing workflow editor called GridNexus. Survey instruments were used to collect results for a comparative analysis.

What we found from the results of the survey is that the three interfaces are comparable with respect to ease or difficulty of use from the perspective of the students. The portal, with its web-based interface, is fairly easy to work with. However, the main drawback of the portal is that it works with grid resources in a batch-like mode, a model that has existed in computer science for many decades. Creating a single workflow that makes use of several grid resources and handles data transfers is not possible. However, getting started with a portal, from the user's perspective, is fairly straight forward.

GridNexus has the advantage of allowing the user to create workflows that work with multiple grid resources and data transfer. The main disadvantage for GridNexus is the steps for setting it up. Fortunately, this is a problem that can be easily overcome with better scripts or setup wizards to make this process smoother. The other concern for GridNexus from the user's perspective is the work required to deal with their proxies. Since the proxies will often expire in a day, students had to create and download a new one often. Although this is not hard, it seems to be a source for confusion. There are three ways to improve this situation. First, we could have had the student put their full credentials on the local machine, saving the step of downloading the proxy. However, this is a potential security breech for local machines that are not secure (such as a computer in a lab). Second, GridNexus could be developed to work with the MyProxy server to request a new proxy. This improvement to GridNexus has been implemented in the time between this course offering and the publication of this paper.

The command line provides users with the full capability to accomplish the required computing on a grid. However, it is not a graphical interface and requires many steps to accomplish multi-step computing with file transfers. Still, some students preferred it to the other interfaces because of the control they had as well as the complete understanding of all that was taking place. However, for scientists that are not computer scientists, we expect the command line would not to be the preferred interface.

In spite of these issues, we feel that having students use three interfaces to accomplish the same processing is a useful learning experience. We feel we can overcome the issues mentioned above, including this issue of the students not fully comprehending the steps of the multi-step processing.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under grant numbers 0410667/053334 and 0737318/0737269/0737208. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We wish to thank our teaching assistant at UNC-Charlotte, Jeremy Villalobos who was responsible for maintaining the grid platform and assisted in the development of the assignments. Also, we would like to thank East Main Educational Consulting, LLC for conducting and administering the surveys as well as gathering and analyzing the results.

8 References

[1] Ferner, Clayton S., and Barry Wilkinson, "Pair-Teaching a Course on Grid Computing from Two Campuses on NCREN," The 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'09), July 13-16, 2009, Las Vegas, Nevada, USA.

[2] Wilkinson, B., and Ferner, C., Teaching grid computing in North Carolina: Two part series, IEEE Distributed Systems Online, vol. 7, no. 6&7, 2006, art. nos. 0606-o6003 and 0607-o7003.

[3] Wilkinson, B., and C. Ferner, "Towards a Top-Down Approach to Teaching an Undergraduate Grid Computing Course," SIGCSE 2008 Technical Symposium on Computer Science Education, March 12-15, 2008, Portland, Oregon US.

[4] Mache, J., Hands-on grid computing with Globus Toolkit 4, Journal of Computing Sciences in Colleges, Vol. 22, No. 2, pp. 99 – 100, Dec. 2006.

[5] Mache, J., and Apon A., Teaching grid computing: topics, exercises, and experiences, IEEE Transactions on Education, Vol. 50, No. 1, pp. 3-9, Feb. 2007.

[6] National Science Foundation grant, "Collaborative Project: Adaptation of Globus Toolkit 3 Tutorials for Undergraduate Computer Science Students," ref. DUE #0410966/#0411237, PIs: A. Apon and J. Mache, 2004-2007.

[7] National Science Foundation grant, "Collaborative: A Multi-Tier Model for Adaptation of Grid Technology to CS-based Undergraduate Curriculum," ref. DUE # 0311473, PI: B. Ramamurthy, co-PI: B. Jayaraman, 2003-2006.