A Consortium to Promote Distributed Computing

Rahman Tashakkori, Barry L. Kurtz
Appalachian State University
Boone, NC 28608
{rt, blk}@cs.appstate.edu

Barry Wilkinson
UNC Charlotte
Charlotte, NC 28223
abw@uncc.edu

Mark A. Holliday
Western Carolina University
Cullowhee, NC 28723
holliday@email.wcu.edu

ABSTRACT

We have been funded by the University of North Carolina Office of the President [1] to establish a consortium to promote high performance computing at comprehensive universities throughout the state [2]. Seven courses were offered twice each over a two year period; students attended remotely via the North Carolina Research and Education Network (NCREN). Local clusters of eight or more computers were established at the twelve universities in the consortium. A prime focus was to promote undergraduate research. In this paper we report on the outcomes of these efforts.

Categories and Subject Descriptors

K.3 [**Computing Milieux**]: Computer and Education – K.3.1 *computer uses in education*, K.3.2 *computer science education*

General Terms

Algorithms, Performance, Experimentation, Security

Keywords

Distributed computing, grid computing, distance education

1. INTRODUCTION

The University of North Carolina Office of the President (UNCOP) announced a request for proposals "to jumpstart the development of advanced research and education applications in high-performance computing, information systems, and computational and computer science." [1] Twelve proposals were submitted and four awards were made. We were awarded $650,000 over two years to establish our consortium. Initially eight schools were involved; during year two we expanded to 12 schools.

Our ***mission*** is to provide the opportunity for undergraduate students at comprehensive universities to study high performance computing at a level comparable to students at Research I institutions, to promote faculty research and to involve undergraduate students in cutting- edge research projects.

Our ***vision*** is that by pooling knowledge, resources, and courses at multiple collaborating comprehensive universities and by establishing a shared grid computing network, we can satisfy our mission statement. Our ***goal*** is to provide students at comprehensive universities the opportunity to take advanced courses in computational science and high performance computing at the undergraduate level and allow these students to participate in cutting-edge team-oriented research projects in support of faculty research programs. Our consortium currently involves twenty faculty members at twelve universities, as shown in Table 1.

Table 1: Consortium Membership

Barry L. Kurtz Rahman Tashakkori	Appalachian State University (lead institution)
David Powell Joel Hollingsworth	Elon University
Bill Hightower Roger Shore	High Point University
Dick Hull Bjarne Berg	Lenoir-Rhyne College
Yaohang Li Stephen Providence	North Carolina A & T
Barry Wilkinson	UNC Charlotte
Shan Suthaharan Sue Lea	UNC Greensboro
Doek-Hyun Hwang William Campbell	UNC Pembroke
Mark Holliday	Western Carolina University
Dean Brock	UNC Asheville
Kwok Wong	Fayetteville State University
Stan Thomas David John	Wake Forest University

2. BACKGROUND

In the realm of high performance computing there has been a shift from traditional parallel computing using tools like MPI to using web-based grid computing services such as Globus GT4. There is more of an emphasis on processing distributed data than on parallelizing a single algorithm. Computer scientists are adjusting to this paradigm shift in high performance computing but these services remain out of the reach of users in other disciplines. Our program helped overcome this problem by providing computer science students trained in the latest technologies that would enable them to work in cross-disciplinary teams to develop cutting edge software applications in related disciplines.

Although there are many high-performance computing initiatives across the U.S., we only mention two of the more popular ones here. The *San Diego Supercomputer Center* (SDSC) focuses on computational and data-oriented science and engineering applications, and serves as an international resource for data cyberinfrastructure through the provision of software, hardware, and human resources in multidisciplinary science and engineering. SDSC is the data-intensive site lead in the NSF-funded TeraGrid. The *Boston University Center for Computational Science (CCS)* fosters computational science education and supports the expansion of computational resources. CCS provides a forum for the multidisciplinary exchange of ideas

ACMSE 2007, March 23-24, 2007, Winston Salem, N. Carolina, USA.

among researchers, educators and students. Regularly scheduled seminars as well as workshops and symposia are offered to highlight advances in computational science. Both of these centers depend on a large technical staff at a major research university. In contrast we wanted to investigate what computational resources could be provided using a geographically distributed set of local clusters at comprehensive universities where the "computing staff" is interested professors and students.

3. THE EDUCATIONAL ENVIRONMENT

We developed and delivered seven courses each academic year: three in the Fall semester, three in the Spring semester, and one in the summer. These courses were available to students throughout the consortium and other universities connected via NCREN. The course titles and enrollment figures appear in Table 2.

Table 2: Course Enrollments

Course Name Term	Number of students	Number of schools
Grid Computing		
Fall 2004	43	7
Fall 2005	33	9
Cryptography		
Fall 2004	15	2
Fall 2005	30	4
Image Processing		
Fall 2004	23	2
Fall 2005	9	2
Parallel Algorithms		
Spring 2005	8	3
Spring 2006	13	1
Intel. Decision Making		
Spring 2005	16	3
Spring 2006	25	7
Monte Carlo Methods		
Spring 2005	10	1
Spring 2006	11	2
Bioinformatics		
Summer 2005	5	2
Summer 2006	8	1
TOTAL	**264**	**16**

3.1 Grid Computing

Grid computing involves using geographically distributed computers collectively for computation and resource sharing. It leads to collaborative teams working together to share their resources to solve problems. Hence to teach Grid computing in this spirit, one should have geographically distributed resources. The Grid computing course is perhaps the first such course in the country at the undergraduate level to involve a large number of geographically distributed sites.

Course materials were developed specifically for undergraduate students centered on a suite of carefully written hands-on assignments that were linked to the lecture materials. Briefly the course starts with web services, as web services are now the basis of grid computing infrastructure. The first assignment calls for the students to create a simple web service and a client to access it. The second assignment requires the student to create a grid service (the form of web services used in grid computing) and a client to access it. These assignments require students to deal with XML (WSDL) files and other details of deploying services, and have final parts that require the students to extend what they have learned. The third assignment requires the students to run a job through the Globus job execution environment (GRAM). The fourth extends this job submission to use a local scheduler (Condor in Fall 2004, Sun Grid Engine in Fall 2005). The fifth assignment introduces students to a grid computer workflow editor called GridNexus developed at the University of North Carolina at Wilmington. This assignment uses web and grid services created in the first and second assignments in a workflow, where now the services can be geographically distributed. Other student work includes each student making a presentation on aspects of grid computing. More details of the course can be found in [3, 4].

3.2 Digital Image Processing

This course is designed to cover digital techniques for image representation, enhancement, compression and restoration. Topics include mathematical background, intensity transformations and spatial filtering, frequency domain processing, image restoration, color image processing, and image compression.

Due to its required mathematics background, this course is difficult to teach at the comprehensive universities. We used a problem-based application-oriented approach; the textbook [5] used MATLAB and provided many different examples on application of image processing.

Some exemplary student projects were:

- Investigating content-based search techniques in medical images of different modalities [6]
- Parallel implementation of Hexagonal Lifting Schemes
- A Java-based enhancement MATLAB Toolbox
- Time sequences of ocean wave crest images

A problem we encountered was the distortion of images that were transferred to the remote sites over the NCREN medium. The compression used in the process would change the image quality such that it would make it very difficult to determine the difference between the original images and the processed ones. To solve the problem, students at the remote sites would download the file containing displayed images on a local machine or on their laptops. Some of the images were also generated dynamically during the class at the remote sites using MATLAB.

3.3 Parallel Algorithms

This course was taught using Barry Wilkinson's textbook [7]. Each of the local clusters at the consortium schools had grid-enabled MPI installed. All programming was done on the local cluster. The major topics were an introduction to parallel computers and message passing, some simple parallel algorithms, divide and conquer strategies, pipelining, synchronization, load balancing, and shared memory vs. distributed memory. A wide range of applications were introduced throughout the course.

3.4 Cryptography and Network Security

The disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security. The major topics in the course are fundamental concepts, mathematical background, the modern development of cryptography and secure encryption protocols and their applications, network security practices, and system security. In addition to the traditional assignments and

labs, this course requires students to complete a substantial project. Two exemplary projects were:

- Allow the user to investigate the performance for RC5 encryption using an elaborate graphical display, and
- Allow students to investigate Elliptic Curve Cryptography using a friendly user interface (see Figure 1).

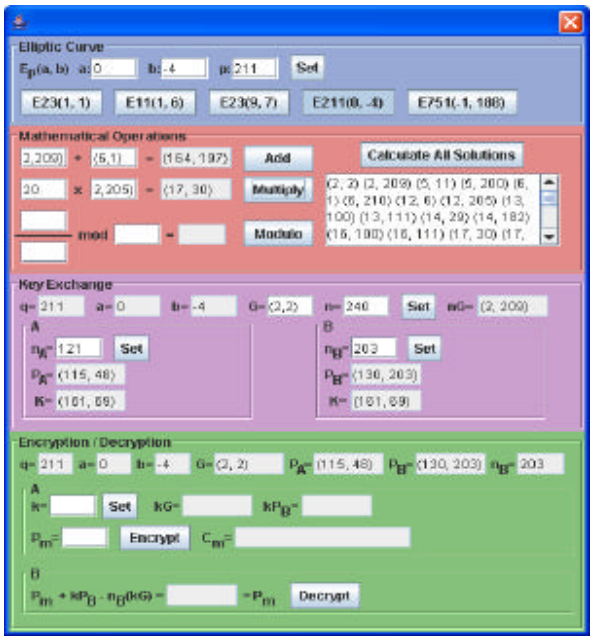

Figure 1: Exploring Elliptic Curve Cryptography

3.5 Intelligent Decision Making

This course is similar to classical operations research in many ways but is more modern and takes advantage of distributed computing. The major topics are: Introduction to Formulation and Classification of Optimization Models; Elements of Improving Search Based Optimization Algorithms; Formulation and Classification of Linear Problems; the Simplex Algorithm; Sensitivity Analysis; Formulation of Unconstrained NLP; Golden Section and Gradient Search; Formulation of Constrained NLP; Penalty Methods and Formulation of Mixed Integer Problems.

A variety of software tools were used. The programming language was AMPL: A Mathematical Programming Language, developed by Brian Kernighan. A user interface in Java provided wrapping using the Facade design pattern. Condor-G provided for multiple and parallel job submission. Grid Services were provided using Globus. An example exemplary project was:

- An Unconstrained Nonlinear Programming Algorithm Using AMPL as a Grid Service

3.6 Monte Carlo Methods

The major topics in this course were Markov Chain Monte Carlo (MCMC), General Monte Carlo Principles, Variance Reduction, Random Number Generation, Quasi-Monte Carlo techniques, and Monte Carlo Applications in Nuclear Simulation, Biology, and Computational Financing. Students completed term papers on the following topics:

- Computational Financing
- Monte Carlo Computation on the Grid
- Ray Chasing Problem using Monte Carlo method
- Network Simulation
- Protein Structure Prediction using Markov Chain Monte Carlo
- Ab initio Monte Carlo

3.7 Bioinformatics

Bioinformatics is a hot topic in computational science. The topics covered in this course included gene structure and information content, protein structure and functions, bioinformatics software tools, including programming in Perl, and bioinformatics algorithms. A number of software tools have been developed. One of the most widespread is BLAST (**B**asic **L**ocal **A**lignment **S**earch **T**ool). We developed the following software:

- Set up a stand alone Blast system with database
- Design and built Globus-enabled HTC Blast on a grid
- Implement parallel MPICH-Blast on a grid

3.8 The NCREN Distance Education Network

All public universities and community colleges in North Carolina are connected by the North Carolina Research and Education Network (NCREN). Three private schools in our consortium, Elon University, High Point University, and Lenoir-Rhyne Collge, were not connected. The grant provided funds for a single classroom to connect these schools to NCREN.

The video portion of the network provides high speed connection in both directions for multiple remote classrooms. One problem was the stimulation of student interaction at remote sites. If the instructor at the delivery site has to monitor three or fewer remote sites then all classrooms can be shown on a single screen. Although interaction is less natural than with local students, at least it is feasible. One of our courses, Grid Computing, was so popular that it was connected to eight remote sites simultaneous. In this case instruction was primarily through an uninterrupted lecture with minimal student interaction.

A second problem with class sessions delivered through video conferencing was the display of technical information. This was particularly obvious with complex figures, mathematical equations, and digital images, such as in the digital image processing class, when small changes needed to be observable. We found two satisfactory solutions: provide the presentation software locally so it could be viewed full screen as the instructor talked about the images and use of third-party software, such as Persony, that allowed independent transmission of PowerPoint presentations. It was critical to have a full time attentive lab technician available at the delivery site to switch between the various formats.

3.9 The Summer Workshops

We held two three-day summer workshops; the first was an "internal" workshop of grant personnel where everyone became acquainted and the second was an open workshop for all participants, including both faculty and students. In particular, 28 faculty and 17 students from 16 different universities participated. The major topics were:

- Day one: distributed MATLAB, MPI
- Day two: grid computing, Globus, lab activities
- Day three: grid computing applications

We were granted a ten month no cost extension during the 2006-2007 academic year at Appalachian, the lead institution. We are

currently sponsoring additional workshops as described under "Future Work" at the end of this paper.

4. THE COMPUTING ENVIRONMENT

4.1 The Local Clusters

Most schools elected to build their local cluster from standard desktop PCs due to their familiarity and ease of maintenance. Two school purchased blade servers. The clusters were typically put on carts so they could be moved to the classroom or other locations as needed. The cluster at Appalachian is shown in Figure 2. All twelve schools in the consortium have a local cluster of eight or more machines.

Figure 2: A Local Cluster

4.2 Globus and Associated Software

The Globus Toolkit is an open source software toolkit used for building Grid systems and applications. This toolkit is developed by the Globus Alliance and many others all over the world. A growing number of projects and companies are using the Globus Toolkit to unlock the potential of grid computing.

Condor is a highly distributed batch system for job scheduling and resource management, and for creating computational grids by linking together computational resources across administrative, geographic, and organizational domains. Recently we have started using Sun Grid Engine that performs similar tasks.

Standard MPI is a library specification for message-passing, proposed as a standard by a broadly based committee of vendors, implementers, and users. MPI was designed for high performance on both massively parallel machines and on workstation clusters.

Grid-enabled MPI allows users to run MPI programs across multiple computers, at the same or different sites, using the same commands that would be used on a parallel computer. This library extends the Argonne MPICH implementation of MPI to use services provided by the Globus Toolkit for authentication, authorization, resource allocation, executable staging, and I/O, as well as for process creation, monitoring, and control.

4.3 Distributed MATLAB

Each participating university was funded to purchase licenses for twenty copies of MATLAB with the following toolboxes: Wavelets, Signal Processing, Image Processing, Neural Networks, Statistics, and Bioinformatics. Additionally, Appalachian purchased the Distributed MATLAB toolbox when it first became available.

Distributed MATLAB provides the following features:

- Distributed execution of coarse-grained MATLAB algorithms and Simulink models on remote MATLAB sessions
- Control of the distributed computing process via a function-based or an object-based interface
- Distributed processing on both homogeneous and heterogeneous platforms
- Support for synchronous and asynchronous operations
- Access to single or multiple clusters by single or multiple users

Figure 3 pictures the setup when using distributed MATLAB.

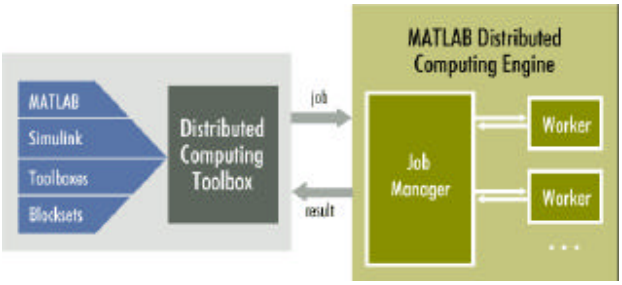

Figure 3: Distributed MATLAB

5. PROJECT OUTCOMES

5.1 Impact on Students

- 15 courses were taken by 264 students at 16 different universities
- 55 students at the 12 consortium universities were employed by the project
- 9 students completed their MS thesis on topics related to distributed computing
- 28 students completed special project courses or honor's courses, see [8] as an example
- 14 students made presentations at conferences or workshops
- 13 students were co-authors on publications
- 19 students attended high performance computing workshops

5.2 Impact on Faculty and Research

- 20 faculty from 12 universities were PIs on the project
- 11 additional faculty attended our high performance computing workshop
- 14 courses related to high performance computing were delivered over two years
- 18 guest speakers participated in the above courses
- 22 faculty made presentations at workshops
- 22 refereed papers were published
- 16 presentations were made at regional, national, or international conferences
- 7 grant proposals were submitted: four were funded and three were not funded

5.3 Impact on Computational Resources

- 12 clusters of computers were purchased and Globus 4.0 was installed on all machines
- Each cluster had 8, 9, or 10 separate computers
- Many terabytes of accessible data storage were installed

5.4 Impact on Access to Distance Education

- Three universities were connected to the NCREN distance education network for the first time
- One of these schools, Elon, developed a new course that was broadcast to seven remote universities

6. Project Evaluation

At the end of the first year we surveyed students who had taken one or more of our courses via NCREN. Questions 1-4 dealt with demographic information; here are the results from questions 5-9.

5. Why electing to take this course.

	Frequency	Percent
Advisor recommended	2	4.5
Needed to fulfill requiremen	14	31.8
Personal interest	27	61.4
No Response	1	2.3
Total	44	100.0

6. Ease of following presentations (1 = Very easy; 10 = Very hard).

	Frequency	Percent
1, 2 or 3 (Easy or Very easy)	13	29.5
4, 5, 6, or 7 (About right)	18	40.9
8, 9 or 10 (Difficult or Very difficult)	13	29.5
Total	44	100.0

7. Effectiveness of instruction (1 = Not at all effective; 10 = Very effective).

	Frequency	Percent
1, 2 or 3 (Not very effective)	3	6.8
4, 5, 6 or 7 (About right)	17	38.6
8, 9 or 10 (Effective or Very effective)	24	54.5
Total	44	100.0

8. Pace of instruction (1 = Very fast; 10 = Very slow).

	Frequency	Percent
1, 2 or 3 (Fast or Very fast)	8	18.2
4, 5, 6, or 7 (About right)	32	72.7
8, 9 or 10 (Slow or Very slow)	4	9.1
Total	44	100.0

9. Quality of instructional delivery system (1 = Very low quality; 10 = Very high quality).

	Frequency	Percent
1, 2 or 3 (Low or Very low quality)	1	2.3
4, 5, 6 or 7 (About right)	22	50.0
8, 9 or 10 (High or Very high quality)	21	47.7
Total	44	100.0

We administered an online survey at the end of the second year. The response rate, especially among students, was poor. Nevertheless, it is still possible to draw some conclusions.

It seems apparent that the clarity of slide presentations (diagrams, program code, and formulas) over NCREN continued to pose problems for some of the students. Third-party software such as Persony helped when it was used.

The fact that students at different sites could not see and interact well with each other may pose a problem in the sense that students are unable to form strong class-wide associations with each other. If the cameras occasionally panned over students from the different sites and zoomed-in on particular students when they chose to comment or answer a question, then students might develop a stronger sense of belonging. This, in turn, might encourage more inter-student and instructor-student interaction.

Several of the sites offering courses in the consortium appeared to experience some difficulty in setting up their local clusters. This was particularly true with respect to the installation of the Globus Toolkit. At three schools we were able to solve this problem by sending our Globus expert at Appalachian State University to help set up their systems.

7. CONCLUSIONS AND FUTURE WORK

The success of courses was largely based on the dedication of the individual faculty members. The two most successful courses in terms of wide inter-university enrollment were the Grid Computing course and the Intelligent Decision Making course. The Cryptography course involved true team teaching with instructors at different universities working together.

Setting up a seamless grid of local clusters of computers was a challenging task due to a wide variety of policies and equipment (e.g., firewalls) in place at each university. Although pairs of universities, such as Appalachian State University and Western Carolina University, were able to share resources, we never were able to establish a network where jobs were distributed seamlessly over the entire grid.

Perhaps the most successful aspect of the project was involving undergraduate students and MS students in a wide variety of research projects relating to high performance computing. We have recently extended this effort by offering weekend workshops for students to learn about distributed computing. The first two workshops were held in November 2006 and January 2007 [2], which were designed for training purpose. Students who attended these workshops are to complete a distributing computing project under the direction of a faculty mentor at their home university. We plan a workshop in April 2007 when these students will come together again to report their research results and to share their experience.

8. REFERENCES

[1] UNC Campuses Partnerships Awarded Computing Grants, http://www.northcarolina.edu/content.php/pres/news/releases/pr2004/20040514c.htm

[2] Kurtz, B. and Tashakkori, R. *A Consortium to Promote Computational Science and High Performance Computing*, $650,000, started 7/01/2004, http://www.cs.appstate.edu/nc-hpc/

[3] Holliday, M., Wilkinson, B., House, J. Daoud, S. and Ferner, C., "A Geographically-Distributed, Assignment-Structured Undergraduate Grid Computing Course", *Proc. of the 36th ACM SIGCSE Technical Symposium*, Saint Louis, MO, February, 2005, pp. 206-210.

[4] Holliday, M.A. Wilkinson, B., and Ruff, J., "Using an End-to-End Demonstration in an Undergraduate Grid Computing Course," *ACMSE 2006: 44th ACM Southeast Conference*, March 10-12, 2006, Melbourne, Florida.

[5] R. Gonzalez, R. Woods, and S. Eddins, Digital Image Processing using MATLAB, Prentice Hall.

[6] High Performance Image Content-based Search", Rahman Tashakkori, Steven H. Heffner, Darren W. Greene, and Barry K. Kurtz, The 2006 International Conference on Image Processing, Computer vision, and Pattern Recognition, IPCV'06 - June 26-29, 2006, Las Vegas.

[7] McKinney, S. *Grid System Security Issues and Solutions*, an Honors Thesis at Appalachian State University, May 2006.

[8] Wilkinson, B. and Allen, M. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2nd edition*, Prentice Hall, 2004.