Laboratory-Oriented Teaching in Web and Distributed Computing[*]

Hongchi Shi, Yi Shang, Michael Jurczyk, and Anupam Joshi[♣]
University of Missouri-Columbia

Abstract

In this paper, we present our experiences in teaching two laboratory-oriented courses in the Web and distributed computing area using our NSF-sponsored Distributed Systems Laboratory (DSL). The DSL consists of one Sun Enterprise 3500 server and 12 Sun Ultra 10 workstations connected through fast Ethernet and ATM networks. The computer systems in this laboratory are solely for the purpose of instruction in computer engineering and computer science, allowing system-level class projects to provide students hands-on experience. Science and Engineering of WWW (CECS 383) and Parallel and Distributed Processing (CECS 486) are two of the system area courses enhanced significantly by the laboratory. Science and Engineering of WWW introduces the fundamental technologies and their applications on the Internet and the Web. Students taking the course are given a sequence of projects to experiment with the technologies. They are asked to set up and configure their own Web servers, study performance and security-related issues, develop e-commerce applications supported by their Web servers. Parallel and Distributed Processing is a graduate-level course. It studies theoretical foundation and provides hands-on experience in the area of distributed and parallel computing. In addition to the theoretical studies, the students are offered a sequence of laboratory projects with the state-of-the-art technologies. These new laboratory-oriented courses have attracted many students and received good feedback. The projects in these courses are shown to be effective in helping students understand the concepts and theories of those subjects.

1. Introduction

As the computational environment in the "real world" becomes increasingly distributed with the advent of the National Information Infrastructure, the area of distributed computing has become an important part of the computer science curriculum. Science and Engineering of WWW and Parallel and Distributed Processing are among the system area courses, such as Networking and Operating Systems, we have been offering to our students. The courses in this area cover most aspects of distributed systems, from low-level network protocol and programming (often at the packet level) to system programming at the operating system level to the application level for Web-based computing and distributed computing.

Hands-on experimentation with system software and protocols can significantly enhance the learning experience for the students and enable them to better appreciate the more practical

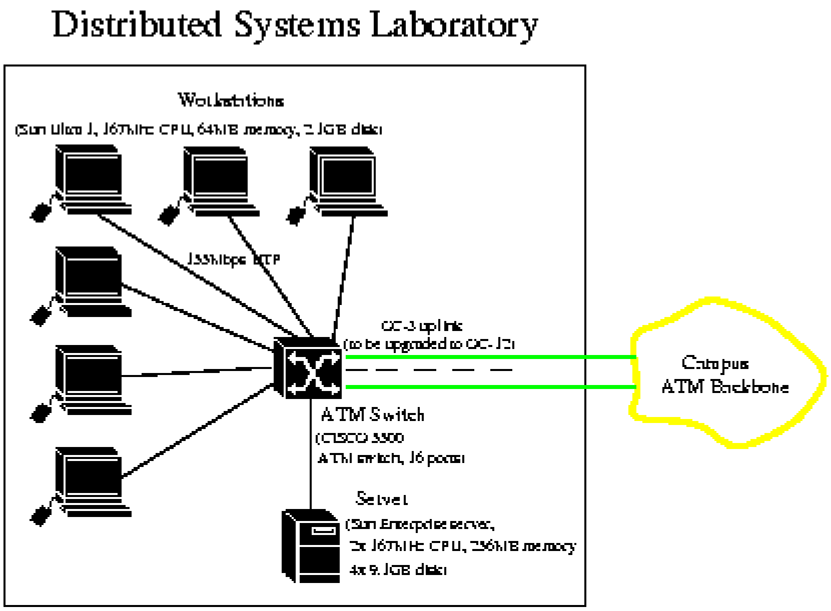
[*] Work partially supported by the National Science Foundation under grants DUE-9851485 and DUE-9980375.

[♣] A. Joshi is currently with University of Maryland at Baltimore County.

aspects of the theoretical constructs taught in lectures. To provide laboratory facilities to support teaching in this area, we have established a dedicated Distributed Systems Laboratory (DSL) with an NSF grant. The DSL consists of one Sun Enterprise 3500 server and 12 Sun Ultra 10 workstations connected through fast Ethernet and ATM networks. The system mimics the environment that the students are likely to encounter in their workplaces in a few years. The DSL enables a laboratory-oriented approach to teaching the theory and practice of distributed computing and algorithm development. Unlike the existing general-use computing facilities provided by the University and the College of Engineering, the DSL provides students system-level access to the computing environment, enabling them to do experiments that are not possible with general-purpose computing facilities.

In this paper, we present our experience of laboratory-oriented teaching in two courses, Science and Engineering of WWW and Parallel and Distributed Processing, enabled by the DSL. In these courses, we are able to ask our students to perform experiments that require changing machine configurations, running instructional operating systems, testing new protocols for distributed systems or networks, and trying out the state-of-the-art software technologies, etc. This kind of experiments is effective in helping students understanding the concepts and theories of those courses.

This paper is organized into four sections. Section 2 shows the laboratory setup of DSL. Section 3 presents our experience in teaching the Parallel and Distributed Processing course. Section 4 presents our experience in teaching the Science and Engineering of WWW course. Finally Section 5 summarizes the paper.



2. Laboratory Setup

The DSL holds an Ethernet and ATM connected cluster of workstations shown in the above figure. The DSL setup closely resembles the networked computational environment that our graduates will increasingly find in their workplaces in the high-technology industry. The high-end server further enhances the sense of realism --- such asymmetric client server environments are increasingly being enabled with the growth of high end computing and communication. Unlike non-switched technologies, using ATM enables our students to study the effect of having separate control and data channels, QoS considerations, and dynamic reconfigurability of the network topology. Thus, with limited expense, the lab is unique and on the leading edge of technology.

3. Teaching Parallel and Distributed Processing

The Parallel and Distributed Processing course studies the fundamental issues and the current research in distributed and parallel computing including systems, architectures, programming models, languages, and software tools. Topics covered include parallelization and distribution models, parallel/distributed architectures, cluster and networked meta-computing systems, distributed programming and applications, and performance analysis.

The course strikes a balance between four elements: lecture, student presentation, independent research, and hands-on programming experience. The lectures cover fundamental issues and the current research in distributed and parallel computing [4,18]. Going through a recent book, "The Grid: Blueprint for a New Computing Infrastructure" written by over 30 distinguished experts in high-performance computing and networking [9], students then gain a clear vision of the future of parallel and distributed computing through their own presentations. Furthermore, students in this course get opportunities to do independent research by selecting a topic of his interest related to parallel and distributed computing, researching the topic in technical journals and other publications, and submitting a research report detailing the findings. Finally, students obtain hands-on experience through a sequence of programming projects involving the state-of-the-art software technologies.

To achieve the best learning result, the students taking this course are given a sequence of five small laboratory projects and one term project. These projects are implemented on the Unix workstations in DSL using Java [10]. The Java programming language is used exclusively in these projects due to a number of reasons. Java can be summed up as a simple, object-oriented, multi-thread, platform-independent programming language designed for distributed applications. It is the most widely used language for the Internet and WWW applications and is becoming a major force in distributed computing. It provides a rich set of extensions supporting networking, multimedia presentation and communication, distributed objects, security, component-based design, and distributed databases. Complicated distributed software systems can be developed in Java much faster than using other languages.

The five small laboratory projects are as follows:

1. Networking in Java. The objective of this assignment is to learn the networking support in Java. It has two parts: (1) Implementation of multiple agents on different hosts that can perform multicasting communication; (2) Implementation of a URL-based Java ClassLoader.
2. Java RMI. The objective is to learn Java RMI, a distributed object system, and implement a search engine for multimedia documents using RMI.
3. Multithreaded Streaming Video Application. The objective is to learn multithreaded

programming, how to cope with limited bandwidth, and concepts of streaming media. Java Media Framework (JMF) is used for transferring and presenting streaming video clips [15].

4. JavaSpaces. The objective is to learn the basic features of JavaSpaces technology, a powerful Jini service that facilitates building distributed applications for the Internet and Intranets [7].
5. A Multi-Agent System for Asynchronous Learning. This project utilizes the knowledge gained in the previous projects and implements a simple Java-based multi-agent system for interactive, asynchronous learning on the Internet [8,13]. Students are free to choose different Java technologies in their implementations.

In addition to these small projects, a term project is used to further develop the problem analysis and system design abilities of the students. The project is based on a main theme, Educational Information Architecture (EIA). EIA is a distributed multi-agent system supporting a student-centered, self-paced, and highly interactive learning approach for computer science education. It integrates existing results on distributed systems, intelligent agents, digital libraries, and educational technologies. The architecture is based on a hybrid of learning models such as problem-based and case-based learning models, for both creative problem solving and mechanical experience simulation in computer science. Note that the EIA has not been implemented and only has a high-level description, which is provided as the basis for the open-ended design in the term project.

The term project consists of selecting a topic related to EIA, analyzing the topic, developing a parallel/distributed design, implementing it on a parallel/distributed computing environment on the Unix workstations in the DSL using Java, measuring system performance, and analyzing performance issues including processing speed and latency, scalability (no change in software and guaranteed QoS as the number of users increases indefinitely), robustness, reliability, flexibility, and the ability of the system to dynamically adapt to changing conditions. The project requires demonstration of a working program and submission of a final report describing the topic, design, implementation, results obtained, and future work.

The feedback from students regarding to this course is excellent, even though the workload of the programming projects are relatively heavy due to the fact that Java is a new language and many of the students in the class do not have extensive experience in programming in Java. It is rated as one of the best courses at the graduate level. These programming projects give the students a taste of distributed computation and information processing in the real world. The support provided by DSL in developing these projects is invaluable. Since many technologies used in the projects are very new, frequent software installation and system setup are required during the course of the instruction. Furthermore, due to improper configuration and usage, these new software environments may bring the systems to a halt and the computers have to be rebooted. Without the access to system-level functionalities supported by the DSL, the projects of this course could not be developed in a timely fashion and achieve such good results.

4. Teaching Science and Engineering of WWW

The Science and Engineering of WWW course is primarily meant for CECS seniors who have already taken a course in the basics of the Internet and the Web. Besides teaching them further details such as SGML/XML, etc., this course mostly deals with introducing them to the Web as a prototypical distributed system and gaining an understanding of all the pieces of the distributed computing puzzle, including the languages, protocols, services, and tools that enable the Web. The topics covered in this course includes client/server computing, hypertext transfer protocol, Web server installation and configuration, Web security, Java applets, Java servlets, Java Swing, extensible markup language, and XML and Java based e-commerce [1-6,11,12,14,16,17,19].

To explore the forefront of WWW technology, the students are assigned significant hands-on work, which is reflected in four small class projects and one term project.

The four small class projects are as follows:

1. Extending browser's functionality. The students are asked to write an application program (maybe script) that wraps a standalone mpeg2 video player "mpeg2player" that a few command-line options. The application program should take from the standard input a list of options and pass the options to the "mpeg2player" program. They then your application program to the Netscape Communicator browser using "x-streamed-mpeg2" as the application type. They finally write a Web page for playing streamed mpeg2 video clips. The Web page a form to collect options from the user and a CGI script to send the options to the application program. They assume a video server is started to send out streamed mpeg2 packets upon request through "TCP" connections.
2. Implementing Java client/server pair. The students are to write a Java client/server pair for a chef to provide the recipe to cooks upon their request. The project helps the students understand how client/server systems, Web browser/server in particular, work.
3. Setting up Web server. Each student is required to install his own Apache Web server on a machine with a port number uniquely assigned to him. They are to experiment with configuring, starting, and stopping the server.
4. Extending Web server's functionality. The students are asked to add the following capability to the servers they installed. If a browser requests an XML document (whose file extension is ".xml"), say an address book XML document (addressbook.xml), the server converts it into an HTML document (that displays the information of the XML document in an appropriate way) and sends it to the browser.

The term project is an "open ended" problem to be completed in groups. Given some sample projects, the students are required to propose your own projects. Each project should address a problem closely related to the Web. The students research the literature for existing solutions, develop a new solution or improve an existing one, and finally implement their projects and write project reports. Each report includes a clear description of the problem addressed, the significance of the problem, existing work, their design and implementation, results obtained and performance analysis, merit of the project, and future work. Sample projects include dynamic Web server load balancing (dispatching client requests transparently among multiple server nodes), Web browsing in low bandwidth scenarios (building a proxy that negotiates with the browser for QoS, actively manipulates the outbound HTTP stream, and transcodes it to reduce bandwidth consistent with the QoS), and Web personalization and mining (developing smart new tools to analyze log and traversal path data to create an adaptive Web site).

The class and term projects enabled by the DSL significantly enhance the results in mastering the concepts of the Web by the students. From the feedback of the students taking the course, the laboratory-oriented approach to teaching Science and Engineering of WWW is very effective.

5. Summary

In this paper, we present our experience of teaching Science and Engineering of WWW and Parallel and Distributed Processing among a number of distributed systems related courses enabled by the Distributed Systems Laboratory (DSL). The two courses have been significantly enhanced by the new facilities in DSL which allow us to adopt a laboratory-oriented teaching approach. The students are given a sequence of projects to experiment with the state-of-the-art technologies such as Java, XML, JavaSpace, and CORBA, and they are allowed to access system-level functionalities. These laboratory-enhanced courses have attracted many students and received very good feedback. The projects are effective in helping students understand the concepts and theories of those subjects.

Bibliography

1. Apache 1.3 User's Guide, http://www.apache.org/docs/, 1999.
2. M. Campione and K. Walrath, The Java Tutorial, 2nd Edition, Addison-Wesley, 1998.
3. D.E. Comer, Computer Networks and Internets, 2nd Edition, Prentice-Hall, 1999.
4. D. Cullur, J.P. Singh, A. Gupta, Parallel Computer Architecture: A Hardware/Software Approach, Morgan Kaufmann, 1998.
5. Document Object Model (DOM) Level 1 Specification, http://www.w3.org/TR/REC-DOM-Level-1/, 1999.
6. Extensible Markup Language, http://www.w3.org/TR/1998/REC-xml-19980210.html, 1999.
7. J. Farley, Java Distributed Computing, O'Reilly, 1998.
8. J. Ferber, Multi-Agent Systems, Addison-Wesley, 1999.
9. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1998.
10. E. Freeman, S. Hupfer, and K. Arnold, JavaSpaces Principles, Patterns, and Practice, Addison-Wesley, 1999.
11. M. Hall, Core Web Programming, Prentice-Hall PTR, 1998.
12. Hypertext Transfer Protocol, http://www.w3.org/Protocols/HTTP/, 1999.
13. JATLite, http://java.Stanford.EDU/java_agent/html/, 1999.
14. Java 2 Platform API Specification, http://java.sun.com/products/jdk/1.2/docs/api/, 1999.
15. Java Media Framework, http://java.sun.com/products/java-media/jmf/, 1999.
16. S. McGrath, XML By Example, Prentice-Hall PTR, 1998.
17. Netscape's DevEdge Documentation, http://developer.netscape.com/docs/, 1999.
18. R. Shuey, D. L. Spooner, and O. Frieder, The Architecture of Distributed Computer Systems, Addison-Wesley, 1997.
19. L.D. Stein, How to Set Up and Maintain a Web Site, 2nd Edition, Addison-Wesley, 1997.

HONGCHI SHI
Hongchi Shi is currently an assistant professor of computer engineering and computer science at the University of Missouri-Columbia.

YI SHANG
Yi Shang is currently an assistant professor of computer engineering and computer science at the University of Missouri-Columbia.

ANUPAM JOSHI
Anupam Joshi is currently an assistant professor of computer science and electrical engineering at the University of Maryland at Baltimore County.

MICHAEL JURCZYK
Michael Jurczyk is currently an assistant professor of computer engineering and computer science at the University of Missouri-Columbia.