# Parallel and Distributed Processing Laboratory for CS 2nd Grade Students: An Active N to N Networking Approach

Nobuhiko Koike and Norihiro Fujii
Department of Computer Science, Faculty of Computer and Information Science, Hosei University
3-7-2 Kajino-cho, Koganei-shi, Tokyo 184-8584, Japan

**Abstract: This work describes authors' approach at the authors' university during last three years for the computer science second grade undergraduate students to experience parallel and distributed computing. The goal is set to give a solid understanding of parallel and distributed processing technologies and to build up basic skills in the field, such as Parallel algorithms, multi-thread/network programming, IP/socket communication, MVC paradigm, RPC/Remote Method Invocation (RMI), Database/SQL, and JAVA/JDBC. The course features a combination of active experimental Learning and N to N networking approach. Unlike typical laboratories where central parallel servers or parallel machines are used (N users to one system networking), our laboratories do without them and instead organize groups of student PCs to form virtual parallel/distributed systems (N users to N systems networking). All PCs work as Servers as well as Clients. Parallel Buckets Sorting and Virtual Shopping Mall implementations are employed for the course projects. The course consists of 14 ninety-minutes sessions within a semester, including introductory JAVA network programming and two projects. As the time is limited, the homework and pre-laboratory experiments are encouraged. The Web based course material distribution and the virtual laboratory environment contributed to bring most students to success.**

## I. INTRODUCTION

Parallel and Distributed Processing Laboratory was designed to offer the first real life experience for the second grade undergraduate student of the computer science department in the faculty of information sciences, Hosei university. The faculty was newly founded in 2000 and about 640 undergraduate students (320 CS students) are now studying wide range of technologies in IT area, such as computer programming (Java, C++ and LISP), software design methodologies, AI/agent, Web-centric distributed applications, media contents creation/ transformations and hardware design. The faculty maintains two computing laboratories and a hardware design laboratory. The campus is very well inter-connected with broadband LAN, and the staff, faculty, and students are all accessible to Web course materials and the Internet. As all students are equipped with Laptops, those materials are also accessible from any place in the campus and at home via the Internet. The department is emphasizing on both parallel processing and distributed processing technologies. Especially, recent advancement in Web based high performance computing, Grid, server side technologies .and e-commerce, motivated us to offer an introductory laboratory to these field at an earlier study stage. The course was created in response to such strong needs to integrate parallel processing and distributed processing into a traditional computer science curriculum. All CS students are required to take this laboratory after Java programming, and introduction to parallel and distributed processing studies.

The paper describes the design and implementation of the course and the N to N networking approach, where all students participate in developing both server side and client side programs and actively work together to realize parallel and distributed applications.

## II. LABORATORY AND COURSE ORGANIZATIONS

The course is offered using two CS computer rooms. Each computer room is equipped with 90 student host PCs, a teacher's PC and a file server, which are connected with the campus LAN and accessible from outside the laboratory via the Web as shown in the Figure 1.

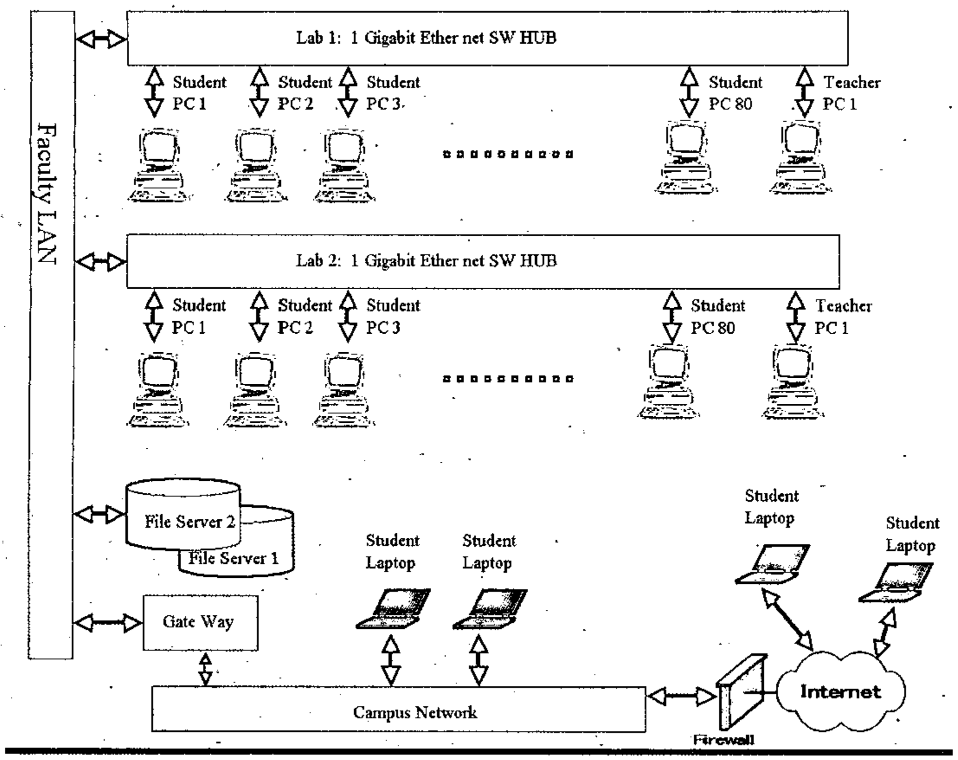


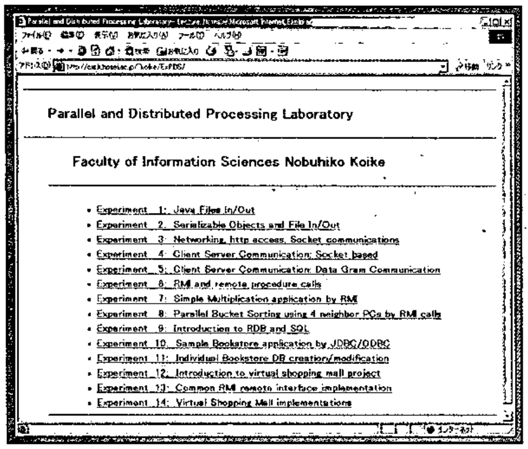
Figure 1. Laboratory Computer and Network Organizations

The course features a combination of active experimental Learning and N to N networking approach. Unlike typical laboratories[1],[2], where central parallel servers or parallel machines are used (N users to one system networking), our laboratory does without them and instead organizes groups of student PCs to form virtual parallel/distributed systems (N users to N systems networking). In the laboratory, all students are requested to set up individual servers and to organize a set of small groups to work together in order to realize the N to N networking. All students participate in developing parallel and distributed programs for both server and client sides. As all method invocation/ RPC calls implement the same pre-determined interface, any student can communicate with other students to exchange data.

This environment is used as a test bed for the final two projects. As few CS students are enthusiastic about scientific parallel applications such as computational fluid dynamics, familiar and easily understandable projects are selected: namely parallel buckets-sorting and virtual shopping mall realization. Thus, most students showed their interests in the projects and kept their motivations to the final goal. These projects are also suitable for applying many important techniques of network programming.

We only show basic programming samples, hands-on and the goals, the rests are up-to students' active participations. The course materials including source codes are stored in the file servers and provided via the Web. As all students are equipped with their own laptop PCs, all course materials and students' host PC facilities are also accessible from out-side, in the campus or at home. Thus, a virtual parallel and distributed processing laboratory is realized. Students' laptops in the home can even take participate in the project as well.

As the second grade students has a few programming skills only in JAVA, network programming based on RMI and JAVA were naturally selected as the basis for the course. The RPC/RMI based parallel programming is appropriate to decouple communications between a master processor and worker processors and can hide complex network related details.



Parallel and Distributed Processing Laboratory

Faculty of Information Sciences Nobuhiko Koike

- Experiment 1: Java Files In/Out
- Experiment 2. Serializable Objects and File In/Out
- Experiment 3: Networking, http access, Socket communications
- Experiment 4: Client Server Communication; Socket based
- Experiment 5: Client Server Communication: Data Gram Communication
- Experiment 6: RMI and remote procedure calls
- Experiment 7: Simple Multiplication application by RMI
- Experiment 8: Parallel Bucket Sorting using 4 neighbor PCs by RMI calls
- Experiment 9: Introduction to RDB and SQL
- Experiment 10. Sample Bookstore application by JDBC/ODBC
- Experiment 11: Individual Bookstore DB creation/modification
- Experiment 12: Introduction to virtual shopping mall project
- Experiment 13: Common RMI remote interface implementation
- Experiment 14: Virtual Shopping Mall implementations

Figure 2. Course Organization

Each student PC works both as a server (worker) and as a client (master). RMI invokes the remote procedure in the worker.

Figure 2 shows a course Web page, which displays the course organization. The course consists of 14 ninety-minutes sessions within a semester, including introductory JAVA network programming and two projects.

The course starts by Java file input/output, serializable objects, and http/web network access[3]. Simple client/server communications employing both socket and data gram communication follows. The similarity and deference between RMI and local method invocation are studied by making use of simple arithmetic multiplication examples. Brief introduction to database/SQL and JDBC/ODBC are carried out using a simple bookstore example[4]. The remaining 4 sessions are dedicated to two projects: parallel sorting and virtual shopping mall.

As the time is limited, the homework and pre-laboratory experiments are encouraged. The Web based course material distribution and the virtual laboratory environment contributed to bring most students to success.

## III. PARALLEL BUCKET SORTING PROJECT

As a typical parallel processing application example, parallel bucket sorting project is employed, as shown in Figure 3. Although the RMI implementation is rather slow, it contains important technologies, such as parallel algorithm, remote procedure calls (RMIs), client –server application and distributed object.

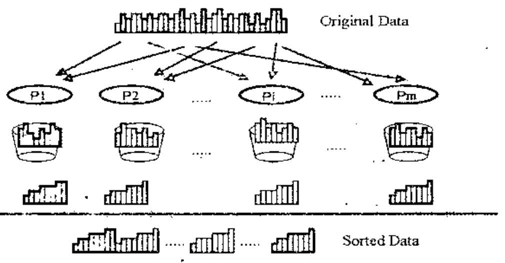


Figure 3. Parallel Bucket Sorting

Unlike conventional client-server systems where many clients and one server are working, our system uses one client and number of servers at once to realize the parallel/concurrent processing.

In order to work servers together in parallel, a multi-threaded client programming is crucial. All client threads generate RMIs to the corresponding servers at once to perform the sorting of the corresponding data parts concurrently, and then merge the result data, as shown in the Figure 4. The data to be sorted can be prepared at client side and sent to servers. For advanced students, an optional realistic example is provided, where data to be sorted can be stored in remote server databases in a distributed fashion and the client can ask servers to sort them in parallel and send the results back to the client to merge.

In order to realize such environment, each student deploys a server program and publishes the stub information to other students. Also, each student constructs a client program, organizing arbitrary number of nearby servers (other students' servers).

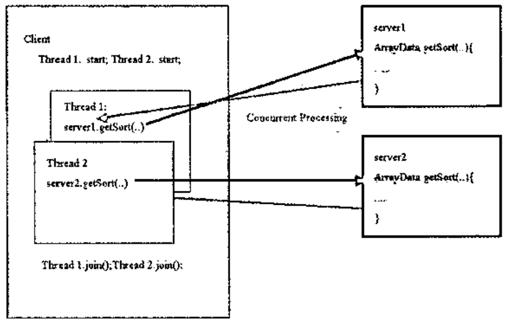


Figure 4 Multi-threaded RMI calls

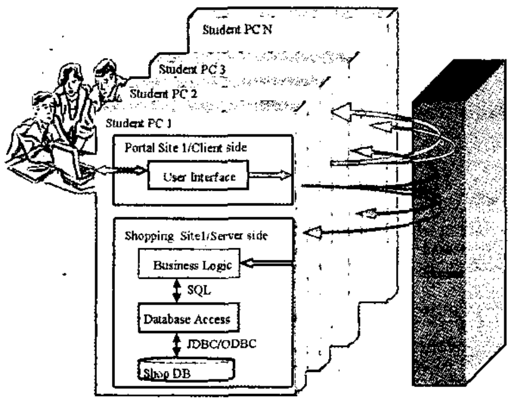


Figure 5. Virtual Shopping Mall Project

## IV. VIRTUAL SHOPPING MALL PROJECT

The next virtual shopping mall project becomes a good introduction to distributed Web-based business applications. Again RMI was employed for the communication media, as all Student PCs are connected within the same LAN and it is convenient to implement such distributed applications.

Each student is requested to develop and deploy a virtual shop with its own database, and also to develop a portal shopping site (client part), which shows a list of available virtual shops and links to those virtual shops in other student PC sites, as shown in Figure 5.

A user selects one of the shops in the list and the portal site (client part) accesses the designated remote shop (remote/local server part) and gets the menu list via the RMI.

Common Remote Interface for All Shopping Sites

```
public interface Shopping extends Remote{
    public Hashtable getProductList() throws RemoteException;
    public integer processOrder ( Hashtable ht )
        throws RemoteException;
}
```

Figure 6. Common Remote Interface (RMI Calls) for the Virtual Shopping Mall

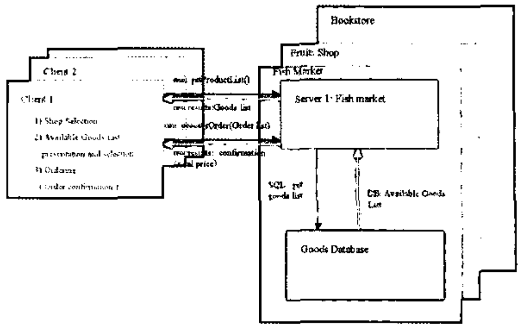


Figure 7. Distributed Three-Layered Implementations and RMI /SQL calls in Virtual Shopping Mall Project

Then, the user selects the goods to buy and submits the order and the total amount of money is returned as the evidence.

In order to achieve a flexible and easy shopping mall organization, where any portal client can communicate with any shops using the same procedure, all shops are requested to implement the common remote interface (RMIs); getProductList() request and processOrder() RMIs, as shown in the Figure 6.

This virtual shopping mall project realizes a typical distributed three layered implementations, as shown in the Figure 7. Any client organizes arbitrary number of available shops as a portal site. As all students work as portal clients, and as shop servers at the same time, the N to N networking is achieved.

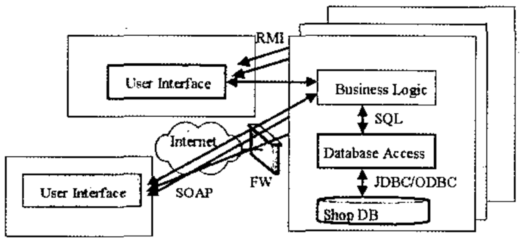


Figure 8. RMI and SOAP calls

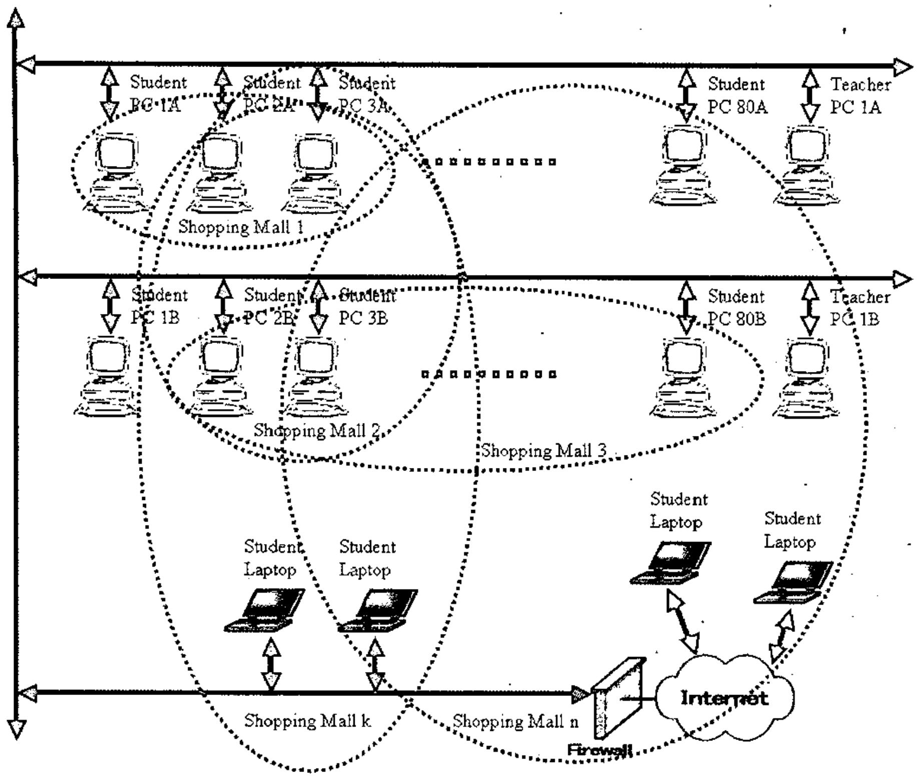


Figure 9. N to N Networking Scheme for Virtual Shopping Mall Projects

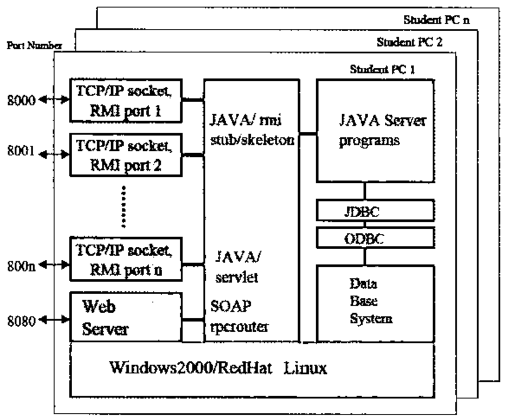


Figure 10. Software Organization of student PCs

Although this example is too simple and unsafe for real applications, it still contains important technological aspects, such as database construction/access through SQL, and multi-site network distributed MVC applications.

For advanced students, the Web services options are also provided in order to expand to the Web environment, by wrapping the RMI calls in XML/SOAP messages based on TOMCAT, as shown in the Figure 8. So, the project can be experimented in wide area network environment including home laptop PCs as clients and servers.

## V. N TO N NETWORKING SCHEME

Both parallel bucket sorting and virtual shopping mall projects employ the N to N networking scheme, where all PCs work as clients and servers. Arbitrary number of servers (up to N) and a client can be organized as a group and intra-group communications are carried out by making use of common RMI/SOAP messages. Up to N groups can be accommodated in the system. Thus, the N to N networking is realized as shown in the Figure 9. As there is no central server, the system is scalable and can avoid the server bottleneck. Applications can be realized like a Per-to-Peer technology. This scheme is also suitable for students to acquire distributed processing technologies, as it gives them chances to develop both client and server side programs and to organize groups in an active way.

The Figure 10 shows the software organization and port connections for each student PC. The system is implemented on Windows/Linux platform, applying standard free software packages, such as Apache, Tomcat and Eclipse. Port number from 8000 to 8079are reserved for individual RMI communication ports, port number 8080 is for http/SOAP communication. MS ACCES is employed for the database part, as it is already installed in every PC.

## VI. CONCLUSION AND FUTURE WORKS

The parallel and distributed laboratory employing the N to N networking scheme is described. It is shown that the proposed scheme is suitable and effective for university laboratory environment. Both parallel bucket sorting project and virtual shopping mall project provided the CS students good chances to program near-real life applications and to build up skills in the field. Most students participated in the projects actively with interests. Web based course material distribution and http/SOAP based virtual laboratory networking allow students to do homework and pre-/post-laboratory experiments effectively in a seamless way. Although the allocated time is rather short, thanks to these functionalities, we could bring most students to success.

We are preparing for the next round sessions. In the future, a combination of recent advanced technologies, such as Web Grid, the Web Services, and the distance laboratory /learning is scheduled.

## REFERENCES

[1] Jean Mehta, "Database-Backed Websites", 32nd ASEE/IEEE Frontiers in Education Conference, Nov. 2002, Session T3G

[2] G.J. Conti, J.M.D. Hill, and C.A. Carver Jr, "Developing an Undergraduate Distributed Development Course", 32nd ASEE/IEEE Frontiers in Education Conference, Nov. 2002, Session T3G

[3] Deitel and Deitel, "Java How to Program – Fourth edition", Prentice Hall, 2002

[4] Deitel, Deitel and Santry, "Advanced Java2 Platform –How to Program—", Prentice Hall, 2002