A Policy-Based Scheduling Tool for Networking Labs

Jenwei Lai and James T. Yu
jlai3@students.depaul.edu  jyu@cs.depaul.edu
School of Computer Science, Telecommunications, and Information Systems
DePaul University, Chicago IL

Abstract

This paper presents a web-based scheduling tool for the networking labs. The growing demand of the distance learning students requires an innovative networking lab that would support students to perform hands-on lab exercises from anywhere in the world with 24×7 availability. However, a hands-on lab is mutually exclusive for multiple users. To address this issue, we developed a web-based scheduling tool where instructors can define the policy for students to access the labs. The architecture and design of the tool are presented in this paper, along with interesting findings from the lab log. Although this tool is designed for the networking labs, it can be easily adopted by any environment with sharing resources.

1. Introduction

Hands-on lab exercises are an integral part of the telecommunication and networking curriculum. With the growing demand of the distance learning students, it becomes a challenge to support hands-on lab exercises. In the summer of 2005, we started a pilot program to support students to access the networking labs from the public as illustrated in Figure 1.

![Figure 1. Remote Lab Access Architecture](image)

An immediate challenge of this remote lab is the scheduling conflict. A lab configuration can be used by only one student at a time. Before we introduce the remote labs, students need to go to the physical labs. If two students come to the lab at the same time, one would need to wait until the other student completes the work. In the case of the remote lab, students would not see each other. It is possible that one student is doing the router configuration while another student is trying to reload the configuration. The pilot program had only nine students during the summer, so this scheduling problem can be easily resolved by using a sign-up paper.
When we decided to provide the remote lab environment for regular classes, it is obvious that a sign-up paper would not work as there are frequent requests to change schedules. This issue motivates us to deploy an on-line scheduling tool to support the remote networking labs. We first explored the Internet to search for commercial or shareware tools that could meet our lab scheduling needs. However, we cannot find any as most of the scheduling tools are designed for appointment rather than for management of shared resources. Two examples are OrgScheduler [1] and Mimosa Scheduling Software [2], and they are primarily for general purpose calendar events. Although they could be configured for lab scheduling application, they do not provide students to access the system over the public internet. In addition, many of those tools require a software client installed on student workstations, while we prefer a simple and standard web interface for lab scheduling. Most importantly, we need to implement policies for the lab scheduling, and the policy could vary from classes to classes and labs to labs. Another important requirement is to track the lab usage for postmortem analysis, and this information needs to be stored in a database for ease of access and analysis.

2. System Environments

We choose Linux as the development and target production environment due to the ease of using scripting language for task automation. We choose Apache Server 2.0 which comes with the Linux Fedora 2.4. The database used in this system is MySQL which is a fast, multi-threaded, and multi-users SQL database server. MySQL server can support up to 4 TB data size with the Linux kernel 2.4 [3]. Based on the data of the last two quarters, a networking course with 10 lab exercises and 50 students uses less than 1MB of disk space. A typical mySQL server on Linux can handle up to 1,500 transactions/queries per second, and our environment for lab scheduling would peak at no more than 2-3 transaction per second. Therefore, scalability would not be an issue on this environment. We discussed between PHP and Perl for the programming language for the web development, where both languages provide powerful API to interface with MySQL. We decide to use PHP because we can reuse much code from a similar system. This environment is expected to be portable to other operating system environment such as Windows. The performance of using Linux, Apache Server, MySQL Server, and PHP is also shown to be better than using the IIS server [4]. The only requirement of user side is the Internet browser. Users can access the schedule system via any browser which supports HTML and JavaScript. The system environment for the server and clients are summarized in Table 1.

<table>
<thead>
<tr>
<th>Table 1. System Environment for the Server and Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
</tr>
<tr>
<td>Web Server</td>
</tr>
<tr>
<td>Web Development</td>
</tr>
<tr>
<td>Server Task Control</td>
</tr>
<tr>
<td>Database</td>
</tr>
<tr>
<td><strong>Client</strong></td>
</tr>
</tbody>
</table>
3. System Architecture and Design

3.1 Data Modeling

The physical data model for a relational database is illustrated in Figure 2. From which, we can see two types of accounts in the admin table. One is for administrator, and the other is for instructors. An instructor has one or more courses, and each course has multiple lab exercises. The database also provides the lab reservation log which provides the capability for postmortem analysis. The Reservation table is the key component of the data model as it contains the actual lab scheduling data. This table also provides the reporting function of showing available time of each lab exercise. The system authenticates each student who may reserve a lab session according to the policy specified for each lab by the instructor. Students can access the labs only during their reserved lab time.

We should also note that there is a Normal Form Violation [5] in the Student_Profile table. If a student takes two classes, the student name and ID become redundant data which has the potential issue of data integrity. However, this table is generated automatically from an input file in Campus Connect. Because of this nature, it is unlikely to have a data integrity issue, and we keep the redundant data to avoid additional join operation. In summary, this design has the performance advantage without the issue of data integrity.
3.2 User Interface Design

The system provides three levels of users: administrators, instructors, and students. Figure 3 shows the interfaces provided for each user group. All users interface with the system via the web.

![Figure 3. Interfaces provided for different kinds of users.](image)

### 3.2.1 Administrator
Lab administrator is responsible for creating instructor accounts which include account ID, password, and course data. It should be noted that an instructor could have multiple courses. Administrator can also modify and delete instructor data.

### 3.2.2 Instructor
The system provides a web interface to allow instructors to upload the student data from the Excel file downloaded from Campus Connect. The PHP program reads the Excel files and writes the data into the database table for each course. Instructor can also delete the student data via the management interface.

The important feature of the lab scheduling tool is the policy specified by instructors for individual labs. Instructors are provided with a management interface to define a lab exercise within a course and the system will create an internal ID of each lab exercise for internal tracking. The management interface allows an instructor to define the policy which includes:

- The starting and ending dates of a lab exercise
- The number of sessions per lab exercise allowed for each student
- The number of hours for each lab session

The system has a general policy that each student is allowed for only one lab per day. This policy is based on the experience that many students will wait for the last day to reserve multiple lab sessions, and some students would not be able to get any lab sessions on the last day. With the policy of only one lab session per day, students are required to plan ahead. After creating a lab exercise, the instructor can use the management interface to modify the lab information or
delete a lab exercise. Reservation Log interface provides instructors with the status of student reservation. An instructor may also overwrite and delete a student lab reservation. For example, if a student has a special need for more lab sessions, the instructor can use the management interface to grant additional lab sessions for that student.

3.2.3 Student. The web interface for students supports multiple classes. When instructors upload the student files, the system collects the last five digits to authenticate students. Only authenticated students can reserve lab sessions. Each lab session has starting and due dates, and students must reserve lab sessions during this interval. The lab reservation is 100% menu-driven and students are given the available slots of time (day/hour) from a friendly calendar interface and available time table. Figure 4 is the screenshot of lab reservation interface for students who can select only those dates that are between the lab starting and due dates. We use different colors to show the status of a given day: time slot available, no time slot available, and time slot already reserved. If a student chooses a date with available time slots, he/she will be presented a new page showing which time slots are available and which are already reserved by other students. With this function, the system does not need to check schedule conflict because only available time could be selected and reserved. After students reserve a lab session, they can also view their reservation history with the option to cancel a reservation. It should be noted that students are not allowed to cancel a lab session that is past. Without this policy, students can delete a past lab session and reserve a new one to beat around the policy of max sessions per lab.

![Figure 4. Screen Shot of Lab Reservation.](image-url)
3.3 Security Control

3.3.1. **Data Control Structure.** The system has three levels of security control as specified for administrator, instructor, and students. Administrator controls the data of courses and instructors who control the data of labs and students who control the data of their lab sessions. Administrator can also overwrite the data of labs and students and instructors can overwrite the data of lab sessions. This control structure is illustrated in Figure 5.

![Figure 5. Data Control Structure](image)

3.3.2. **Password encryption.** Each account has its own password which is used for access control. In order to prevent hackers from sniffing passwords over the network, we use encoding function [6] of PHP to encrypt the passwords. The passwords in the database are stored in the encrypted format as well.

3.3.3. **Activities Log.** Both the activities of lab reservation and Linux server access are logged, including IP address and timestamps. If there is any unauthorized access or use of the system, the administrator can check the activity log to determine the cause and severity of the activity. Two examples of log data are given below:

<table>
<thead>
<tr>
<th></th>
<th>TDC511</th>
<th>lab01</th>
<th>09-19-2006</th>
<th>23:00</th>
<th>jyu</th>
<th>Delete</th>
<th>09-11-2006 15:34</th>
<th>140.192.33.153</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>TDC511</td>
<td>lab01</td>
<td>09-19-2006</td>
<td>17:00</td>
<td>jyu</td>
<td>Reserve</td>
<td>09-11-2006 14:01</td>
<td>140.192.33.153</td>
</tr>
</tbody>
</table>

4. Data Analysis of Lab Usage

4.1 Usage and Major Change Issue

The system was developed during the summer of 2006, and put into production use in the Fall of 2006 (School Calendar 2006/2007). Since then, it has supported nine TDC courses with more than 200 students. The system is currently supporting 4-6 courses per quarter.
One issue with the original design is the use of four digits of student ID for authentication. We didn’t use all digits of student ID for protection of individual identify. When the system was used in the second quarter, we experienced a problem of the duplicated IDs and students complained there lab sessions being interfered by other students. This problem is similar to the famous birthday paradox problem [7]. In this incident, the class has three sessions: in class, distance learning, and a special session in Jordan with a total of sixty-five students. We calculate the probability of duplicate is (1-82%) = 18% which is significant. To understand the severity of this issue, we study the probably of unique IDs with four and five digits and the results are illustrated in Figure 6. Given that it is unlikely to have a class with over 100 students, we are confident of using five digits as the unique key for student IDs where the probability is less than 5%. Although the probability of duplicate IDs is extremely low, instructors should be advised to check duplicate IDs.

![Figure 6. Probability of Unique Student IDs.](image)

4.2 Data Analysis

The lab log also provides an opportunity of postmortem data analysis. We choose one lab exercise with 33 students where this lab duration is two weeks. The data shows that the majority of the lab sessions are logged in the last three days as illustrated in Figure 7.

![Figure 7. Utilization of Labs.](image)
Although this data would not surprise anyone, it provides an evidence for instructors to help students to plan their lab sessions ahead of time. This data also shows that students should have no excuse of not having enough time doing the lab.

5. Conclusion

The lab environment represents a limiting resource that needs to be shared by a large student population. With the design of the remote lab, we are able to provide 24×7 service for students to access the networking labs from anywhere in the world. The scheduling tool provides an effective mechanism to balance the resource utilization. One unique feature of this lab scheduling tool is the support of policy for individual labs, and the policy is controlled by the instructors. Although this scheduling tool is designed for the networking labs, its functionality is applicable to any environment with shared resources.

Additional features are being considered to enhance the tool capability. One enhancement under consideration is for students to report lab problems where critical problems are automatically forwarded to the lab assistant and instructors. Students could check the web site if there are known problems before their lab sessions, and problem resolutions are automatically posted on-line.

References