Optimization of LMS for Improving User Response Time

M. Tech. Project Stage-1 Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology

by

B. RAJU
Roll No.: 123050091

under the guidance of

Prof. Deepak B Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

October 2013
Acknowledgements

I would like to thank my guide, Prof. Deepak B Phatak for the consistent directions towards my work. I would also like to thank Mr. Nagesh Karmali for his valuable inputs. Due to their consistent encouragement and right directions, I am able to contribute better to this project work. Lastly, I would like to thank Srinu, Prashanth, Sukh Deo and Firuza for proof reading this report and providing their valuable feedback.
Abstract

e-Learning effectively useful for students and teachers in the form of effective communication. LMS defined as interactive learning, in which the learning content is available online, and provides automatic feedback to the student's learning activities.

Moodle is one of the Learning Management System which is an Open Source Software. Moodle developed by Martin Dougiamas. Our primary goal of this project is to optimize Moodle, which will improve the user response time. This can be done in many ways, but here we are discussing two ways: Th first one has two methods namely a). by analyzing a the moodle database logs, b). making new index on updated tables, and the second one is to save Moodle from security attacks i.e, is trying to crash the remote servers, and making server overload, etc., which will degrade the performance of Moodle.

Second part of this project is edX, which is a non-profit MOOC created by Harvard and MIT. edX source has been made available to open source community in different modules. In this project, i had done literature survey on edX, edX database structure. Lastly we have compared the performance of Moodle LMS and edX, based on the accessing database tables by conducting quiz.
Contents

1 Introduction .................................................. 1
  1.1 Motivation .................................................. 1
  1.2 Learning Management System ................................. 1

2 Moodle LMS and Other LMS’s to Moodle LMS .................. 3
  2.1 History of Moodle ........................................... 3
  2.2 Reasons and Features ......................................... 4
    2.2.1 Moodle Features ......................................... 4
  2.3 Feature comparison of Moodle with Other LMS’s[10] ............ 5
  2.4 Available Users in Moodle .................................... 9
  2.5 Moodle Statistics ........................................... 10

3 Literature Survey on MOODLE LMS ............................. 13
  3.1 Moodle as Modular System .................................... 13
    3.1.1 Moodle Three Layer Architecture .......................... 13
    3.1.2 Architecture of Moodle Site ............................... 14
  3.2 Moodle Database ............................................. 16
    3.2.1 Tables in Moodle Database ................................ 16
    3.2.2 Logical view of MySQL server architecture[33] .......... 17
    3.2.3 Log file analysis .......................................... 18
    3.2.4 Security Level[10, 34] .................................... 20
  3.3 Performance Testing Tools .................................... 20
    3.3.1 Jmeter[9] .................................................. 20
    3.3.2 mk-query-digest[4] ....................................... 21
    3.3.3 PhpMyadmin[5] ............................................ 21
  3.4 Experiments .................................................. 21
    3.4.1 Performance of different Moodle Modules .................... 21
    3.4.2 LOC in Moodle ............................................ 22
    3.4.3 Limitations of Moodle LMS ................................. 23

4 edX ............................................................. 24
  4.1 Introduction ................................................ 24
  4.2 edX database ............................................... 25
    4.2.1 Tables in edX ............................................. 25
    4.2.2 quiz ..................................................... 26
  4.3 Problem Statement ........................................... 28
5 Conclusion and Future work

5.1 Conclusion ................................................................. 30
5.2 Future Work ............................................................... 30

A Some details on Moodle and edX

A.1 Moodle ........................................................................... 31
A.1.1 Moodle Installation[7] .................................................. 31
A.1.2 LOC of Moodle ......................................................... 32
A.1.3 phpMyAdmin setup ..................................................... 33
A.1.4 Changes in MySQL Configuration file ......................... 35
A.1.5 MySQL Log file results .............................................. 35
A.2 edX .............................................................................. 36
A.2.1 edX Installation Steps[8] .............................................. 36
Chapter 1

Introduction

1.1 Motivation

Moodle LMS, an open source software is used by 239 countries all over the world. Moodle has a strong support of communities where the experts are involved in the discussion. DBP Moodle in IIT Bombay, has performance issues with respect to user response time in moodle 2.5. Those, by improving the user response time of Moodle, it is going to be useful for 73.7 Million users and also 1.3 Million teachers, and so on, across the world. User response time can improved in many ways, i.e., by changing the Moodle architecture, hardware or software configuration, optimizing the slow queries in databases, etc.

1.2 Learning Management System

Distance education represents the mode of delivering a course of study, in which most of the communication between teacher and student takes place non-contiguously. e-learning is an interactive learning, in which the learning content is accessible online, which offers automatic feedback to students learning behaviour[23].

e-learning will effectively be useful for students and teachers in the form of effective communication, conducting online exams, and return immediate feedback to students, etc., which also tries to remove the time constraints and location constraints.

Comparison of traditional and e-learning method is shown in figure 1.1. By looking at figure, we can say that the process of learning is easy using internet. Students can interact with their instructor for 24 hours and 7 days per week, via discussion forums, emails, chat, and so on. In traditional classroom lecture, content is manually dis-

Figure 1.1: Comparison of LMS in traditional and e-learning[31]
tributed, and once the class is over, the students can’t interact with lecturer.

e-learning provides the opportunity to the students to communicate with teacher, and as well as themselves. This communication is done through emails or using discussion forums or chat rooms. Virtual Learning Environment (VLE) is defined as interactive learning, in which the learning content is available online, and provides automatic feedback to the student’s learning activities[1]. Open and Distance learning mode is especially for working adults, who need to support of their families as well as manage their career while pursuing higher education.

Requirements

The following features[12] are required for LMS in an e-learning process.

- Multiple language support
- Modular structure
- Discussion forums, group work
- Ease of installation
- Database support
- Video conferencing support
- Creating input in different input format (like Office file, javascript, php, MPEg file, etc.)
- Online exam, exam module
- chat, wiki, survey, search
- Whiteboard
- Backup support
- XML support to work with different systems

LMS softwares are available in commercial as well as Open source softwares.

1. Open Source LMS’s are Moodle, edX, SAKAI, WebCT, Bscw, Illias, eduplone, Claroline, Atutor, Dokeos and so on. These open source softwares have wide developer communities, which presents robust arguments for considering open source software, as a straightforward and potentially feasible competitor to commercial products[12, 13].

2. Commercial LMS’s are Blackboard, Desire2Learn, KEWL, Blackboard Learning System, ANGEL Learning Management Suite, and eCollege, etc.

This report discusses, literature survey on various LMS’s and Importance of Moodle LMS in Chapter 2. The detailed structure of Moodle is explained in chapter 3. Lastly, the report explains edX platform in chapter 4.
Chapter 2

Moodle LMS and Other LMS’s to Moodle LMS

Moodle is an open source software e-learning platform (also known as Course Management System (CMS), or Learning Management System (LMS), or Virtual Learning Environment (VLE)), which provides educators tools to create a course website [13, 20]. It is a web-based Learning Content Management System (LCMS) i.e. a CMS, and VLE, which provides information and collaboration among the Moodle Users such as System Administrators, developers, course designers, etc..

2.1 History of Moodle

Moodle LMS was developed from a social constructivist viewpoint by Martin Dougiamas at Curtin University in western Australia [1]. In 2002, he was a Webmaster of a university and a system administrator of WebCT installation. He started to develop Moodle to solve some problems with WebCT. The original version was targeted for small classes and a case study, but steadily many features were added by developers and other contributors from all over the world. On 20th August 2002, first version of Moodle was released.

The main intention of this project is to help educators by creating online courses with focus of interaction and collaboration of content and continuous evaluation by learners themselves or teachers. Moodle used in many environments like education, training, development, etc.. Now a days, Moodle is used for primary and secondary schools, non-profit organizations, hospitals, private companies, libraries, etc..

Technologies Used in Moodle:

This will give the overview of how moodle works at a technical level. This will be helpful to devel-

![Moodle Architecture](image)
opers and administrators of Moodle. The Figure 2.1 represents the layers of current versions of Moodle software.

1. Moodle source code is written in PHP language.

2. Moodle requires a platform running Linux, Apache, MySQL and PHP; This set of applications is referred to as a LAMP.

Moodle LMS competes with famous commercial learning systems such as Blackboard, SharePoint LMS and Desire2Learn, and so on[32]. The main advantage of this Open Source Moodle LMS is that, anyone can modify according to their requirements, and can also be able to add new modules.

2.2 Reasons and Features

Moodle community has grown extensively, and both developers and users have actively participated in the discussion forums, by sharing their ideas, tips, resources, and also helping new users. Some of the reasons for using Moodle are listed below[10]

1. Moodle is an Open Source software. This means, that anyone can download, and modify moodle source code as per our needs, also we can distribute it. Under the same license.

2. Easy to learn and provides high quality online courses. Instructors are sharing documents/lecture materials, graded assignments, discussion forums, etc., amongst students.

3. Provides good educational tools/plugin, and easy to install.

4. Can be used on almost all servers that run PHP. Moodle can be easily download from http://moodle.org, and can be upgraded to the latest version.

5. One Million users are registered who interact through Moodle Community, which helps to share of ideas. Non-core developers also provide their efforts to develop plugins based on their requirements. Due to these factors and support, Moodle has emerged as a global project[1].

6. Moodle allows users to post news items, assignments, electronic journals and resources, and to collect assignments etc.

7. Moodle is CMS & VLE, and lets teachers provide and share documents, graded assignments, quizzes, discussion forums, etc. with their students in an easy-to-learn manner and to create quality online courses.

2.2.1 Moodle Features

The various features[1] of moodle are as follows

1. Assignment submissions

2. Discussion forums

3. Files download

4. Grading
2.3 Feature comparison of Moodle with Other LMS’s[10]

Is comparison is done keeping in mind various metrics. Some of them are given below.

For simplicity and clarity, these features and capabilities[19], are further divided into three phases, which are given below

1. **Learner Tools**:
   These can be divided into three types, a). student involvement tools(group work, community), b). Productivity Tools(Book marks, calendar, search engine), c). Communication Tools(Discussion forums, File exchange, Emails, chat).

2. **Support Tools**:
   It contains three kinds of Tools i.e, Administrative, Course Delivery, and Content Development.

3. **Technical Specification Tools**:
   There are two kinds of tools, Hardware/software tools and Pricing/Licensing Tools. If the product doesn’t have cost, then that product has an advantage, because we are mostly encouraging the open source softwares only.
Moodle LMS is compared with other LMS's (edX, Bleakborad, Desire2Learn, ANGEL, eCollege, Dokeos, Sakai) based on the above mentioned features.

Here we consider no cost product as Yes, and with cost product consider as No.

We conclude that Moodle LMS supports almost all features except two, which are book marking and company profile under licensing category of Technical specification[10]. If we will add these features, then Moodle will work in more a productive way.

Moodle LMS is also good in all technical aspects like architectural, implementation, community Internationalization, etc., it is accessibility way, where accessibility is concerned, is average[26]

<table>
<thead>
<tr>
<th>1) Learner Tools</th>
<th>2) Support Tools</th>
<th>3) Technical Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Communication Tools</strong></td>
<td><strong>1. Administration Tools</strong></td>
<td><strong>1. Hardware/Software</strong></td>
</tr>
<tr>
<td>- Discussion Forums</td>
<td>- Authentication</td>
<td>- Client Browser Required</td>
</tr>
<tr>
<td>- File Exchange / Internal Email</td>
<td>- Course Authorization</td>
<td>- Database Requirements</td>
</tr>
<tr>
<td>- Online Journal/Notes</td>
<td>- Registration Integration</td>
<td>- Server Software</td>
</tr>
<tr>
<td>- Real-time Chat</td>
<td>- Hosted Services</td>
<td>- UNIX Server</td>
</tr>
<tr>
<td>- Video Services / Whiteboard</td>
<td></td>
<td>- Windows Server</td>
</tr>
<tr>
<td><strong>2. Productivity Tools</strong></td>
<td><strong>2. Course Delivery Tools</strong></td>
<td><strong>2. Pricing/Licensing</strong></td>
</tr>
<tr>
<td>- Bookmarks</td>
<td>- Course Management</td>
<td>- Company Profile</td>
</tr>
<tr>
<td>- Orientation / Help</td>
<td>- Instructor Helpdesk</td>
<td>- Costs</td>
</tr>
<tr>
<td>- Searching Within Course</td>
<td>- Online Grading Tools</td>
<td>- Open Source</td>
</tr>
<tr>
<td>- Calendar / Progress Review</td>
<td>- Student Tracking</td>
<td>- Optional Extras</td>
</tr>
<tr>
<td>- Work Offline/Synchronize</td>
<td>- Automated Testing and Scoring</td>
<td>- Software Version</td>
</tr>
<tr>
<td><strong>3. Student Involvement Tools</strong></td>
<td><strong>3. Curriculum Design</strong></td>
<td></td>
</tr>
<tr>
<td>- Groupwork</td>
<td>- Accessibility Compliance</td>
<td></td>
</tr>
<tr>
<td>- Self-assessment</td>
<td>- Course Templates</td>
<td></td>
</tr>
<tr>
<td>- Student Community Building</td>
<td>- Curriculum Management</td>
<td></td>
</tr>
<tr>
<td>- Student Portfolios</td>
<td>- Customized Look and Feel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Instructional Standards Compliance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Instructional Design Tools</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Content Sharing / Reuse</td>
<td></td>
</tr>
</tbody>
</table>
## Names of LMS’s

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Learner Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Communication Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discussion Forums</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Discussion Management</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>File Exchange</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Internal Email</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Online Journal/Notes</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Real-Time chat</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Video services</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>whiteboard</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>1.2. Productivity Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Book Marks</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Calendar/Progress review</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Orientation/Help</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Searching Within Course</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Work Offline/Synchronize</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>1.3 Student Involvement Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Groupwork</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Student Community Building</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Student Portfolios</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Total features</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Total available feature</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>Total missing features</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.1: Learner tools[2, 10, 19, 26]
### Names of LMS's

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
</table>

#### 2. Support Tools

<table>
<thead>
<tr>
<th></th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.1 Administration Tools</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Course authentication</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Hosted Services</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Registration Integration</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.2 Course Delivery Tools</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test types</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Automated Testing Management</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Automated testing Support</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Course Management</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Online Grading</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>student tracking</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.3 Content Development Tools</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accessibility Compliane</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>content sharing /reuse</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>course templates</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Customized Look and Feel</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Instruction Design</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Instructional standards compliance</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Total features</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Total available feature</td>
<td>16</td>
<td>16</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Total mission features</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.2: Support tools[2, 10, 19, 26]

<table>
<thead>
<tr>
<th></th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3. Technical Specifications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3.1 Hardware/Software</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client Browser required</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Database Requirements</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Unix Server</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Windows server</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3.2 Pricing/Licensing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Company profile</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Costs</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Open source</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Optional extras</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Total features</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Total available features</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>total missing features</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2.3: Technical specifications[2, 10, 19, 26]
2.4 Available Users in Moodle

The roles of people using moodle are as follows

1. Administrator: He/she has all permissions to do anything. The entire site will run under his/her supervision. Admin has permissions to delete/add any person, course, etc..

2. Course Creator: He/she can create/add/delete the courses in bulk or single, and also retrieve the course data, store, and so on.

3. Teachers: This role has permissions to do anything for particular course. He is involved in discussion forums, announcements, grading, and enrolling or unenrolling the students, etc..

4. Non-editing Teachers: This role has permissions to teach the course, and give the grades. He is not allowed to modify the course, upload documents, and etc..

5. Students: Users with this role can review the course content, search content, download the material, etc.. When a student enrolls to the course he/she can check grades, respective.

6. Guest: This role has lowest permissions. This role can have permissions to see the public course or teachers names but he does not have permissions to download content, participate in discussion forums, etc..

These people can access and maintain Moodle. User access level of Moodle is shown in Figure 2.3

Student Perspective

- Can attempt online exams like quiz.
- Can easily submit the assignments, project work, etc..
- They can collaborate between students while doing project. New point Can ask questions to the instructor in forums, which will make the communication between teacher and student effective.

Faculty View

Moodle allows the instructor to present and provide feedback to conduct course exams, upload the assignments, share the graded assignments, documents, workshops, chat, and conduct online quizzes, forum for learners, etc., with the students in an easy-to-learn way, to generate quality online courses. In this way, students can easily learn, and communicate between instructor and students effectively.

Administrator view

Moodle site administrator has full permission site. Administrator is capable of adding or deleting courses, users, etc.. Admin can change any profile.
Academic view

1. Moodle LMS is an Open Source Software, so any institute can use Moodle, which provides all functionalities equivalent to commercial softwares.

2. Some of the higher educational institutes are developing plugins depending on their requirements.

2.5 Moodle Statistics

Moodle Statistics have been collected from www.moodle.org/stats in September 2013. These values are updated dynamically as and when they register.

Moodle Registrations

Table 2.5 represents the Moodle registration in different views.

By observing last two months of registration which will drastically increases.

Top 10 countries registration

These results shows registered top 10 countries in Table 2.6[16] out of 239 countries.

Moodle Versions

The official releases of Moodle is shown in the following Figure 2.4.

<table>
<thead>
<tr>
<th>Registration</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moodle Registered sites</td>
<td>87,079</td>
</tr>
<tr>
<td>Countries</td>
<td>239</td>
</tr>
<tr>
<td>Courses</td>
<td>7,879,675</td>
</tr>
<tr>
<td>Users</td>
<td>73,735,447</td>
</tr>
<tr>
<td>Teachers</td>
<td>1,300,207</td>
</tr>
<tr>
<td>Enrolments</td>
<td>78,437,509</td>
</tr>
<tr>
<td>Forum posts</td>
<td>130,594,058</td>
</tr>
<tr>
<td>Resources</td>
<td>69,986,407</td>
</tr>
<tr>
<td>Quiz questions</td>
<td>199,610,906</td>
</tr>
<tr>
<td>Moodle Translated languages</td>
<td>121</td>
</tr>
</tbody>
</table>

Table 2.4: Moodle Registrations[1]
### Table 2.5: Moodle Registration variation[1]

<table>
<thead>
<tr>
<th>Registration</th>
<th>2-October-2013 Count</th>
<th>10-July-2013 Count</th>
<th>variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moodle Registered sites</td>
<td>87,079</td>
<td>83,059</td>
<td>4,020</td>
</tr>
<tr>
<td>Countries</td>
<td>239</td>
<td>236</td>
<td>3</td>
</tr>
<tr>
<td>Courses</td>
<td>7,879,675</td>
<td>7,592,349</td>
<td>2,87,326</td>
</tr>
<tr>
<td>Users</td>
<td>73,735,447</td>
<td>70,696,570</td>
<td>30,387,77</td>
</tr>
<tr>
<td>Teachers</td>
<td>1,300,207</td>
<td>1,200,890</td>
<td>99,317</td>
</tr>
</tbody>
</table>

### Table 2.6: Top 10 countries registration[16]

<table>
<thead>
<tr>
<th>Country Name</th>
<th>Registrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>14,469</td>
</tr>
<tr>
<td>Spain</td>
<td>7,579</td>
</tr>
<tr>
<td>Brazil</td>
<td>6,526</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>4,546</td>
</tr>
<tr>
<td>Mexico</td>
<td>3,794</td>
</tr>
<tr>
<td>Germany</td>
<td>3,352</td>
</tr>
<tr>
<td>Colombia</td>
<td>2,771</td>
</tr>
<tr>
<td>Portugal</td>
<td>2,356</td>
</tr>
<tr>
<td>Australia</td>
<td>2,064</td>
</tr>
<tr>
<td>Italy</td>
<td>2,049</td>
</tr>
</tbody>
</table>

### Figure 2.4: Moodle version releases[1]
Version Registration

1. New Moodle Registration:
Comparison of last two months of moodle versions is shown in Figure 2.5.

![Figure 2.5: New Moodle Registrations in Last two months][16]

2. All Moodle Registrations: Moodle registration version wise is shown in Figure 2.6.

![Figure 2.6: All Moodle registrations in Last two months][16]
Chapter 3

Literature Survey on MOODLE LMS

3.1 Moodle as Modular System

Moodle is structured as an application core, surrounded by numerous plugins to provide specific functionality. Moodle is designed to be highly extensible and customizable without modifying the core libraries. So when you are trying to upgrade to a newer version, it always creates problem. To avoid this, Moodle follows a plugin architecture.

Standard Moodle distribution includes Moodle core and number of plugins of different types. These are helpful for users to start teaching and learning immediately. Most of the plugins shared publically are listed out in Table 3.1

Moodle basic code contains minimum functionalities needed to build the LMS. Moodle core needs to be used by every newly developed plugins. Moodle core includes the following

1. Courses and Activities like wiki’s, quizzes, etc.
2. Users
3. User functionalities
4. Logs and Statistics
5. Libraries

3.1.1 Moodle Three Layer Architecture

Moodle can be represented in a three layer architecture as shown in the Figure 3.1. Moodle distinguishes between code (PHP, HTML, and CSS) and data. Moodle library, modules (such as resources and activities), blocks, plugins, and other entities are represented in the code. This code contains all elements that deal with front-end and back end operations. Interface is a boundary, across which the systems communicate. User interface refers to communication between software and human beings. For example, in moodle, this is what we see and click in the web browser, and in our mail programs. It provides ways for us to access, understand, and change the database at the heart of any Moodle site[17]. The design of such an interface requires collaboration between software. Users makes this user friendly and maximise overall usability.
Moodle administrator can change system settings of Moodle courses, roles, groups, and other data, such as learning resources added by teachers, and student involved in discussion forums, chats, and so on. All of these are stored in Moodle database. The pictures uploaded by users, conducted workshop tutorials, uploaded assignments, etc., are also stored in moodledata directory.

### 3.1.2 Architecture of Moodle Site

Moodle site architecture is shown in Figure 3.2. This shows the available people, administrations, summary of activities, etc.. This is useful for understanding the Moodle site.

**Figure 3.1: Moodle Three Layer Architecture[33]**

**Figure 3.2: Moodle Site Architecture[10]**

**Module wise explanation[10, 24, 34]**

Moodle architecture is divided into 6 groups,
1. **Communication Module:** This module is very important to provide the high quality of education. This includes the discussion forums, real time chat, etc.,

2. **Productivity Module:** This module contains the search, calendar, help, progress and review modules.

3. **Student Involvement Module:** This module represents the workshop, group work, and self assessment modules.

4. **Administration Module:** This type of module is very important to the entire system because gaining access into these modules result in having the access in all other modules. The well known authentication, course, and user authorization, registration integration and any other module goes into this group. Finding and fixing the bugs in these modules become crucial in any LMS development.

5. **Course delivery module:** This module can be defined as helpdesk, course management, student tracking, automated and testing module for supporting assignments.

6. **Curriculum design module:** Customization and course templates and any other module is classified under this group.

These modules shown in Table 2.1, and 2.2

**Plugins in Moodle**

Moodle developers can extend the Moodle by adding plugins. This means Moodle is designed in a modular way, which allows flexibility to add or remove functionality of Moodle at many levels. Plugins can be divided into the following categories

1. Activities
2. Blocks
3. Themes
4. Users
5. Course formats
6. Filters
7. Reports
8. Gradebook
9. General plugins
10. Editors
11. Messaging
12. Repositories
13. Portfolios
14. Plagiarism

15. Admin tools

Some of of Plugins listed out in table 3.1

<table>
<thead>
<tr>
<th>Question types</th>
<th>Admin reports</th>
<th>Assignment types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication plugins</td>
<td>Workshop evaluators</td>
<td>Content editors</td>
</tr>
<tr>
<td>Course formats</td>
<td>Course reports</td>
<td>Course importers</td>
</tr>
<tr>
<td>Database fields</td>
<td>Database presets</td>
<td>Enrolment plugins</td>
</tr>
<tr>
<td>Filters</td>
<td>Gradebook report</td>
<td>Gradebook export</td>
</tr>
<tr>
<td>Gradebook import</td>
<td>Message senders</td>
<td>Plagiarism detection plugins</td>
</tr>
<tr>
<td>Activity modules</td>
<td>Question import/export formats</td>
<td>Quiz reports</td>
</tr>
<tr>
<td>Repository plugins</td>
<td>Resource types</td>
<td>Search engine adapters</td>
</tr>
<tr>
<td>Themes</td>
<td>Web services</td>
<td>Workshop allocators</td>
</tr>
<tr>
<td>Workshop forms</td>
<td>Blocks</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: List of Available plugin’s in Moodle

3.2 Moodle Database

Moodle is a collection of many different plugins instead of just single complex application. Moodle database schema contains many tables (more than 310 approximately). These tables are a collection of core database tables and the tables which are belonging to each plugin. This Moodle database structure is defined in install.xml file placed under the db folder in each plugin.

The entire database is defined, edited, and upgraded using the XML system. This XMLDB is the Moodle database abstraction layer which contains the library code and this allows Moodle to interact and accessing the database. Moodle uses ADODB, which is abbreviated as ActiveX Data Object. This is the database abstraction library for PHP.

For analyzing the Moodle performance, we used different tools like Jmeter, mk-query-digest, and http-load, and so on.

3.2.1 Tables in Moodle Database

The present Moodle database schema has the 314 tables and uses InnoDB by default. Other storage engines like MyISAM, Merge MyISAM, etc., are also supported. When you create a table, MySQL stores the table definition in a .frm file.

These tables are categorized[17] based on type of data they store. They are

1. Configuration
2. Users and Profiles
3. Roles and Capabilities System
4. Courses
5. Groups
6. Logging System
There are 11 quiz tables in the database of Moodle. In addition to these tables, this module references other tables like questions, grades, etc. The detailed description of these tables can be seen in Fig A.2

3.2.2 Logical view of MySQL server architecture[33]

This will help us understand the server in the form of MySQL components. Figure 3.3 shows the logical view of MySQL's architecture. In the Figure 3.3, the topmost layer contains services most network based client/server tools or servers need: connection handling, authentication, security, etc.

Second layer consist of query processing, analysis, optimization, caching, and all predefined functions like date, math, time, etc., In this level, storage engine will provide functionalities like views, triggers, etc.

The last layer contains the storage engines like InnoDB, MyISAM, Blackhole, etc., The response of storage engines are storing and retrieving all data stored in MySQL. The server communication is done through storage engine API(Application Programming Interface) only. These storage engines don’t communicate with each other, they simply respond to the request from server.

Figure 3.3: MySQL Server Architecture
Moodle Database Storage Engines:

Storage Engines are defined as a particular set of method which is used to store and manage information in a database. Each storage engine offers a different way to store, index, and lock data. The ability of MySQL is to allow you to use more than one storage engine. Indexes are implemented in the storage engine layer, not the server layer. List of Moodle database supported, and default storage engine are shown below

1. **InnoDB[17]**
   - Moodle database server uses InnoDB storage engine as the default one for MySQL 5.5 or later versions.
   - This provide the ACID transaction features. Along with this it supports foreign key, table spaces(it can specify only database storage locations)][1], Full text indexes from MySQL 5.6 onwards.
   - It works more robustly, performs better with big sites, and allows better data integrity features (transactions).
   - InnoDB is better for write-intensive websites, i.e, sites that heavily utilize inserts and updates.
   - InnoDB uses more system resources (such as RAM) than MyISAM.

2. **MyISAM**: Allows to perform fast read operation structure of its indexes. Moodle 1.9 or earlier uses MyISAM as the default storage engine.

3.2.3 **Log file analysis**

As moodle uses its own abstraction layer to convert the PHP queries to SQL queries, it is quite interesting to know, if there is a performance issue in the SQL queries that are generated.

Application profiling and server profiling are used for finding the performance. Application profiling provides the complete picture of system performance. MySQL profiling also provides the entire information which is not available at application profiling. For example, PHP code doesn’t show you, how many rows of MySQL are examined for executing queries.

We can only find out where MySQL spends most of the time by using application profiling. MySQL has two kinds of query logs.

1. **General query log**

2. **Slow query log**

**General query log[33]**: These log queries have all queries which are received by the server and the queries that may not be even executed once due to some error. The server writes information to this general log when the client is connected or disconnected. It logs each SQL statement received from the clients. This means that, whatever queries a server receives, it will write into the log. Thus, it contains queries that may not even be executed once due to errors. These will be useful for any client that is facing connection problem. In this case, administrator will look at these log. These general query logs does not contain the execution time of queries or after completion of query information. It also does not maintain general log.
Log file generates 554 queries per second while attempting quiz.

**Slow query log[3, 33]:** This contains the list of SQL statements that are taking more time than the general log query. Slow log query maintains query execution time. These can be defined as logs queries that take more than specified amount of time to execute.

These two log files are useful for finding the bottlenecks of the MySQL server performance. Out of these two, slow query log has the first preference for finding the problematic queries. In order to know what queries get executed on performing a particular action on moodle, First we have to set up a SQL log file. Thus, our first consideration is to get the queries, and later check the performance of those queries. The minimum and default values of long-query-time is 1 and 10 seconds respectively(If the query time is between the specified limit, or more then that query will be stored in slow log file)[3].

Slow query log will helpful for optimization because it contains queries which are taking long time to execute.
The slow log file contains following metrics,

1. Query time
2. Lock time
3. Rows sent
4. Rows examined

The default storage engine for MySQL is INNODB which supports row-level locking. MYISAM storage engine supports full table locks. Every storage engine does not support all functionalities, for example, MYISAM will not implement transactions, and InnoDB does not support FULL TEXT search indexes.

**Basic SQL Server Query Optimization[25]:**
For increasing the performance, try to write your own queries in different number of ways and compare their reads and execution plans. General tips for query optimization are as follows

1. Use the alternate of COUNT (*)
2. Try to avoid usage of HAVING clause in select statements, as it will select all rows first and then it will filter rows.
3. Try to minimize the number of sub query blocks within query
4. Try to use Non-Columns expression
5. When we need only return 'N' rows then try to use TOP keyword or SET ROWCOUNT statements in select statement.
6. Try to use TRUNCATE TABLE command instead of DELETE command when you need to delete all rows of the table.
3.2.4 Security Level[10, 34]

Moodle LMS components are developed without a specific design documentation including its security services[20]. Important disadvantage which can make the system weak in case of attacks. LMS’s are client/server web applications that, among rest, handle user requests coming from client such as web browsers, for handling user request they frequently require accessing security critical resources (such as database and files) at the server end[23]. Moodle LMS has much vulnerability[21, 22], these flaws can be classified into four groups[11] as follows

1. Authentication: further classification of authentication attacks are
   (a) Design attack like password prediction, and username prediction. This can be saved by implementing captcha.
   (b) Session attack: In this can be avoided by providing SSL.
2. Availability
3. Confidentiality
4. Integrity

So we need to develop a mechanism that defends these security flaws of Moodle LMS. Out of all the categories, our main focus is on available attacks, because the objective of available attacks to make e-learning services and data is unavillable to authorized end users. Most popular available attack is denial of service attack[21, 23]. A denial service attack (DoS) or distributed denial service attack is an attempt to make a machine or network resource unavailable(such attacks usually lead to a server overload) to its users. Target of a DoS attack may vary. It generally consists of efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

The two types of DoS attacks available are

1. Logic attacks: These can exploits the existing LMS flaws to crash remote servers or slowly decrease the performance.
2. Flooding attacks: These can overload LMS with a high number of requests, and disable users from accessing e-learning resources.

Thus, Moodle may be facing the problem of logical attacks at the time of conducting online quizzes, which may decrease the performance.

3.3 Performance Testing Tools

For finding the bottlenecks of Moodle, we have used the following tools which are open source softwares. Here we will discuss the brief introduction of these tools in Moodle and its usage.

3.3.1 Jmeter[9]

JMeter was the tool extensively used for testing the performance of Moodle under various test conditions. To compare with other servers, extensive tests were done with the default Moodle configuration. Jmeter, developed in Java, is an open source desktops application. It is developed in Java applications.
<table>
<thead>
<tr>
<th>RAM</th>
<th>4GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel Core™ i3-2370M</td>
</tr>
<tr>
<td>Operating System</td>
<td>Ubuntu 12.04 LTS</td>
</tr>
<tr>
<td>Hard Disk</td>
<td>320GB</td>
</tr>
</tbody>
</table>

Table 3.2: System Configuration

These Jmeter tool is used for Load testing for analyzing and measuring the performance of variety of services and it is mainly focused on web applications. Using Jmeter, we can perform the test on static and also dynamic resources i.e. databases, and queries, etc..

Some of the main activity modules for institute, tested using jmeter is shown in the figure 3.4. By observing resultant graph, we can conclude that, most of the time is taken by the quiz module.

3.3.2 mk-query-digest[4]

This tool is used for analyzing the query execution logs and it will generate the report for MySQL. This tool is not only for MySQL, it is also used for PostgreSQL, memcached, et..

3.3.3 PhpMyadmin[5]

phpmyadmin is used to handle administration of entire MySQL server as well as a single database over the web. For accessing the database, we need to setup MySQL server properly. phpMyAdmin is a free software tool written in PHP. For installing phpMyAdmin, we need to install web server (Such as LAMP) because phpMyAdmin interface is entirely accessed on browser.

Features of phpmyadmin are

- Create, copy, drop, rename, tables, columns and indexes
- Browse and drop database, tables, views, etc..
- Import the text files into tables
- export data to various formats: csv, xml, pdf, etc..
- it will support the InnoDB tables and foreign keys

When user logs into phpmyadmin, the user need to provide username and password of MySQL. phpmyadmin does not have it’s own management.

3.4 Experiments

3.4.1 Performance of different Moodle Modules

Moodle has lot of activity modules such as chat, wiki, and so on. Currently, we have not compared all of them. Thus, the main useful modules like quiz, chat, forum, etc., are chosen for comparisons in the form of response time using Jmeter tool.

Quiz module degrades the database performance which is shown in the Figure 3.4. These results are
Table 3.3: Moodle: Lines of Code

<table>
<thead>
<tr>
<th>Directory name</th>
<th>LOC</th>
<th>Directory name</th>
<th>LOC</th>
<th>Directory name</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod</td>
<td>1,39,393</td>
<td>lib</td>
<td>8,89,205</td>
<td>backup</td>
<td>58,794</td>
</tr>
<tr>
<td>admin</td>
<td>1,02,890</td>
<td>theme</td>
<td>46,178</td>
<td>question</td>
<td>52,200</td>
</tr>
<tr>
<td>grade</td>
<td>17,919</td>
<td>filter</td>
<td>24,493</td>
<td>course</td>
<td>17,344</td>
</tr>
<tr>
<td>enrol</td>
<td>17,518</td>
<td>blocks</td>
<td>11,451</td>
<td>pix</td>
<td>13,287</td>
</tr>
<tr>
<td>lang</td>
<td>10,319</td>
<td>repository</td>
<td>10,777</td>
<td>cache</td>
<td>7,983</td>
</tr>
<tr>
<td>auth</td>
<td>9,960</td>
<td>report</td>
<td>6,620</td>
<td>user</td>
<td>7,365</td>
</tr>
<tr>
<td>calendar</td>
<td>4,407</td>
<td>install</td>
<td>6,296</td>
<td>badges</td>
<td>3,704</td>
</tr>
<tr>
<td>webservice</td>
<td>3,969</td>
<td>message</td>
<td>3,269</td>
<td>group</td>
<td>3,424</td>
</tr>
<tr>
<td>mnet</td>
<td>2,390</td>
<td>blog</td>
<td>2,392</td>
<td>cohort</td>
<td>1,553</td>
</tr>
<tr>
<td>tag</td>
<td>2,013</td>
<td>portfolio</td>
<td>1,401</td>
<td>completion</td>
<td>1,477</td>
</tr>
<tr>
<td>rating</td>
<td>1,050</td>
<td>comment</td>
<td>1,225</td>
<td>files</td>
<td>884</td>
</tr>
<tr>
<td>login</td>
<td>1,025</td>
<td>iplookup</td>
<td>381</td>
<td>notes</td>
<td>854</td>
</tr>
<tr>
<td>my</td>
<td>161</td>
<td>local</td>
<td>212</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

conducted based on the values of Table 3.2. So we need to optimize the database server, by applying the optimization techniques. Figure 3.4 represents the Quiz module having high response ratio. The experiment composed of 5 users, 3 quiz questions and the users accessed all modules simultaneously.

3.4.2 LOC in Moodle

LOC can be useful for measuring the complexity or productivity or volume of code in Moodle. Total lines of code for entire Moodle is 1,488 Millions. Lines of code for Moodle directory wise shown in the Table 3.3.

From Table 3.3, we can see that the LOC for admin, mod, and lib has huge LOC. Thus from these we can conclude that mod contains additionally developed modules as compared to basic module. Thus developers can add/delete modules easily.
3.4.3 Limitations of Moodle LMS

Following are the limitations of Moodle

1. As compared to edX platform, Moodle LMS accesses more tables while attempting quiz.

2. Moodle does not support database 'views'

3. Moodle can be operated by only IT experts[21].

4. While installation, users need to know the minimum technical terminology. Thus, its difficult for beginners to install.

5. Most of the moodle communities are in english language only. Thus, for clarifying any doubts regarding moodle, user needs to know english. This causes inconvenience for other non english speaking users.

6. Moodle requires course manager for generating, and providing online materials to instructors and technicians, else, Moodle will stay on empty shell[10].

7. Moodle does not maintain the SSL functioning all over the site[11, 22, 34].

8. Attacker can easily attack, as the Moodle stores the user data into cache, which can be used by attacker to launch the attack for next session[20, 23].
edX is a non-profit organization which is formed by the collaboration of Harvard and MIT. Open edX is an open source platform for building MOOCs (Massive Open Online Course which is online education system) with various advanced features to make the online education more effective. edX organization itself has a MOOC named as edX which hosts courses from various institute across the world.

edX organization provides certification by the institutes through protected test exams. Though edX provides various common features (like study materials, quizzes, and online exams, etc.), it also provides various assessment strategies for the essays like peer assessment, staff assessment, instructor assessment and AI assessments.

4.1 Introduction

edX is available on Github website which is a version control system. The Open edX consists of modular structure, and is implemented mostly in Python (with some Ruby and NodeJS as well), and the code is made available under an AGPL license.

Basic Modules in edX are as follows,

- **edX Platform**: This contains both LMS and CMS-Studio. In LMS, students are it's main users, and it provides the user with interface to pursue online education. CMS has instructor as it’s user and it provides interface to instructor to create courses.

- **edX- ORA** : edX-ORA is used for the assessment of open response problems on the edx platform. It has support for Self, Peer, Staff and Machine Grading. Each of these graders is implemented as separate django applications, with the controller having common logic, such as submission of assignments and returning results to LMS.

- **codejail**: It deals with the security for the system by providing a sandbox (a testing environment) for the execution of the untrusted code.

- **xqueue**: xqueue takes assignments being submitted by the user from the LMS and handles the assignment using queues, and provides the assignment to the external grader for getting graded. There after the grader module returns the graded assignments back to the xqueue which will be returned back to the LMS and proper changes to database is made.
• **cs_comment_service:** By using this module, we can enable group discussion forums in the edX system. cs_comments_service module is written in “Ruby” language, and it communicates with both type of database present in the system i.e SQLite and MongoDB.

All of modules can be accessed on Github, and can be contributed to enhance the edX-MOOC.

### 4.2 edX database

The edX database system uses Sqlite and Mongodb. It is built on python framework known as django. The role of SQLite in edX platform is used to store the user information, user profile, courseware details, course and student modules, sample wiki data, enrollment data, etc.. This is a relational database system, where data is organized in different tables. MongoDB is used to store the course information. Django, a Python web framework, consists of an object-relational mapper (ORM), back-end controller, and template system. MongoDB is a document-oriented database (also known as a NoSQL database) effective for scaling and high performance[6].

#### 4.2.1 Tables in edX

The edX platform uses the django framework for development and production of the edX platform. It inherits the django’s legacy database. The edX platform consist of total 88 tables including django’s tables.

This django database consist of total 11 tables in edX, which are listed below

1. django_admin_log
2. django_comment_client_permission
3. django_comment_client_permission_roles
4. django_comment_client_role
5. django_comment_client_role_users
6. django_content_type
7. django_openid_auth_association
8. django_openid_auth_nonce
9. django_openid_auth_useropenid
10. django_session
11. django_site

Along with these tables SQLite has 77 tables, tables are classified as follows

1. **Authentication**
2. **Bulk_emails**
3. **Celery**
<table>
<thead>
<tr>
<th>Table Name</th>
<th>No. of Times Table Accessed</th>
</tr>
</thead>
<tbody>
<tr>
<td>courseware_studentmodule</td>
<td>56</td>
</tr>
<tr>
<td>auth_group</td>
<td>38</td>
</tr>
<tr>
<td>Django_session</td>
<td>33</td>
</tr>
<tr>
<td>courseware_xmodulecontentfield</td>
<td>19</td>
</tr>
<tr>
<td>courseware_xmodulesettingsfile</td>
<td>19</td>
</tr>
<tr>
<td>Auth_user</td>
<td>10</td>
</tr>
<tr>
<td>student_courseenrollment</td>
<td>6</td>
</tr>
<tr>
<td>certificates_generatedcertificate</td>
<td>1</td>
</tr>
<tr>
<td>external_auth_externauthmap</td>
<td>1</td>
</tr>
<tr>
<td>auth_userprofile</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.1: list of edX database tables accessed for quiz

4. Certificates
5. Circuit
6. Course
7. Courseware
8. Djcelery
9. Licenses
10. Notify
11. Shopping Cart
12. Tracking
13. Verify
14. Student
15. Wiki

4.2.2 quiz

To compare the database tables that are accessed while conducting the quiz, we have conducted experiments on edX and moodle. The quiz is consist of 2 questions which are Multiple choice question and short answer question. These database tables are retrieved from mysql_log file and edx.log file. Table 4.1 shown the database tables accessed in edX platform.

The following database tables are accessed in Moodle LMS

1. mdl_config
2. mdl_context
3. mdl_course
4. mdl_sessions
5. mdl_config_plugins
6. mdl_capabilities
7. mdl_cache_flags
8. mdl_course_format_options
9. mdl_user_lastaccess
10. mdl_user_preferences
11. mdl_groups
12. mdl_group_members
13. mdl_user_info_field
14. mdl_role
15. mdl_role_capabilities
16. mdl_role_assignments
17. mdl_block_instances
18. mdl_message
19. mdl_repository
20. mdl_enrol
21. mdl_modules
22. mdl_forum
23. mdl_forum_discussions
24. mdl_event
25. mdl_log
26. mdl_forum_posts
27. mdl_course_categories
28. mdl_quiz_overrides
29. mdl_quiz_attempts
30. mdl_quiz_grades
31. mdl_grade_items
32. mdl_grade_settings
33. mdl_quiz_feedback
By observing these tables, edX accesses only 10 tables, whereas Moodle database is accessing 46 tables for conducting quizzes which contains only two questions. We can conclude that Moodle database is accessed more tables compared to edX. Thus, load of the database accessing in edX is very less as compared to Moodle database.

4.3 Problem Statement

1. Conduct efficient online quizzes and surveys of Moodle LMS and edX LMS: Online response collection has to be solved efficiently and effectively. Currently Moodle is well known that it is not scalable for more than 3,500 concurrent users. About edX, currently we do not know yet because edX only installed in IIT Bombay. MOOC does not give any idea on how many number of concurrent users it will support, and time limit of any activity (like quizzes and feedback form), etc. We will investigate, how many number of concurrent users edX platform will support. There is quite possibility that, edX also may have some performance issues that we don’t know now.

2. There following are the roadmaps for IIT Bombay

   Adopt edX platform for all MOOC’s courses. Since many teachers will offer MOOC’s Course, and edX scalable platform. It is possible for IIT Bombay itself to use edX instead of Moodle. This decision is not taken yet. This decision will be taken after 6 months the use of edX. Following are possibilities

   - Moodle continues to use addition of edX-Platform, In this case these optimization problems must be solved for moodle.
   - Moodle is used for internal courses, and edX is used for external courses, including those IIT students who takes external courses. It is required that edX and Moodle could be interfaced with each other, for student response collection.
- IIT Bombay completely eliminates moodle, and completely shifts to edX, however, knowing that moodle has been established. Such a complete shift requires 2 to 3 years.

Hence, in any case, we need to resolve potential performance problems on moodle. So we wish to look at performance problems on moodle as an outline and solve them. If time permits, we would like to integrate and interface data between moodle and edX.

3. **Tables in LMS’s**: Moodle database accesses more tables for quiz activity. If we will merge tables and make new index on those tables, then the number of joins will decrease drastically. This may give good response time.

4. Log data is currently stored in a single standard Moodle database table(mdl-log), which is directly accessed. Database stores information extremely robustly (in terms of protection against loss or corruption), which is not necessary for log information, because some logged actions do not contain important information which is useful later. All logs go to one standard database table(mdl-log) which is not scalable. It would be more efficient to store this information somewhere else.

   The solution is to store the logging file on a separate machine. This will reduce the load and size of database storage. This can be done by making a new plugin.

5. **Security Attack**: Most popular attack is denial of service attack which can degrades the moodle performance while conducting the online exams, and also will crash the remote servers. Denial of service attack is a challenge to make resources unavailable to actual users (or) server load might increase, by sending the bulk number of fake requests. So if we will model new mechanism which will avoid this security problem then Moodle LMS can give good performance and our confidential details also will be in safe.
Chapter 5

Conclusion and Future work

5.1 Conclusion

We had studied in detailed structure of Moodle. When we conducted some experiments like conducting online quiz, we noticed some of the problems like more number of tables is accessed by moodle compared to other LMS. Other than this, we found some more bottleneck in Moodle, which effecting the Moodle performance. we have analyzed slow queries of the database log files, and also security vulnerabilities in Moodle. These problems motivate me to solve performance issues of Moodle, so that the user response time will be improved.

5.2 Future Work

1. eDX has been introduced in IIT Bombay recently. Currently they have only installed edX. We will investigate scalability of edX platform, and if time permits we will integrate the moodle with edX platform.

2. Presently, we started learning django framework to understand the edX database.

3. We will merge tables based on slow queries, and making new index on those newly merged tables in moodle.

4. We are planning to test the security vulnerabilities while conducting any online response collection from students.

5. Studying PHP functions that generate SQL queries and try to minimize the irrelevant queries as far as possible.
Appendix A

Some details on Moodle and edX

A.1 Moodle

A.1.1 Moodle Installation[7]

Step by step Moodle Installation

Here i am going to give the installation of Moodle for only developers or administrators because we can install Moodle in other way( like directly installing LAMP server which is having the Linux, Apache, Moodle, PHP. In these we can't modify code).

1. sudo apt-get apache2 mysql mysql-server php5
2. sudo apt-get install php5-curl php5-gd php5-intl php5-xmlrpc
3. Download Moodle
   - cd /var/www
   - sudo tar -zxf moodle-latest-24.tgz
   - sudo mkdir /var/moodledata
   - sudo chmod 777 /var/moodledata
   - sudo chmod 777 /var/www/moodle
4. Create Moodle database
   - create database moodledb charset=utf8;
   - create user 'mdluser'@'localhost' identified by 'password';
   - grant all privileges on moodledb.* to 'mdluser'@'localhost';
5. Setup Moodle server

Jmeter Tool Setup[9]

- Generate Jmeter Script
- Extract the jmeter script files into [moodle dir]/admin/report/loadtesting
- Login to moodle.
• Select Settings > Site Administration > Reports > JMeter loadtesting
• Start jmeter.
• To open JMeter, visit the downloaded Jmeter folder JMeter> bin > ApacheJMeter.jar
• Double-click on the ApacheJMeter.jar file
• Open the newly generated script
• Select the correct file
• Click Open
• Run the script and view the results
• Repeat the experiment by varying the number of users
• Record the response time and throughput

Extracting the generating script of different moodle modules shown in the following Figure

![Generate JMeter Script](image)

**Figure A.1: Generate Jmeter script**

### A.1.2 LOC of Moodle

I wrote script for finding the lines of code for Moodle that code(loc.py) is shown below

```python
#!/usr/bin/python
import os
import sys

def CountFile(f):
    flag = False
    counter = 0
    f = open(f, "r")
    for line in f.read().split(\"\n\"):
        ...
```

32
```python
if line.strip() == '':
    continue
elif '//' in line:
    continue
elif '/*' in line:
    flag = True
elif flag:
    if '*/' in line:
        flag = False
else:
    counter = counter + 1
f.close()
return counter

def CountDir(dirname):
    counter = 0
    for f in os.listdir(dirname):
        fa = os.path.join(dirname, f)
        if os.path.isdir(fa):
            dcount = CountDir(fa)
            counter = counter + dcount
        else:
            fcount = CountFile(fa)
            counter = counter + fcount
    return counter
print CountDir(sys.argv[1])
```

Execute the script as follows:
$./loc.py path_of_the_moodle_directory

### A.1.3 phpMyAdmin setup

phpMyadmin setup done as follows,

1. Open the terminal (by using CTRL+ALT+T)
2. sudo apt-get install phpmyadmin
3. Choose the web server
4. Setup the database: you need to set up the database in mysql for phpmyadmin
5. Enter MySQL adminstration password
6. Create the phpMyAdmin password, this will allow you to log onto phpmyadmin in your web browser
7. check phpMyAdmin by typing localhost/phpmyadmin in web browser
Quiz database

Figure A.2: Moodle Quiz database tables[17]
A.1.4 Changes in MySQL Configuration file

For generating slow query log file, we need to modify or make changes in my.cnf file which is shown below

1. sudo nano /etc/mysql/my.cnf
2. log-slow-queries = /var/log/mysql/mysql.log
3. long query time = 1
4. save changes
5. Restart mysql server: sudo service mysql restart

A.1.5 MySQL Log file results

When i run the Moodle on configuration (shown in table 3.2) then i founded some of the queries which are taking more than their limit, mentioned below,

```
# Time: 131002 10:32:54
# User@Host: debian-sys-maint[debian-sys-maint] @ localhost []
# Query_time: 3.549691 Lock_time: 0.000102 Rows_sent: 31 Rows_examined: 429
SET timestamp=1380690174;
select concat('select count(*) into @discard from 
    TABLE_SCHEMA, 
    TABLE_NAME, 
) from information_schema.TABLES where ENGINE='MyISAM';
```

Here the first line of the output represents the when query was logged, second line shows who executed this query, line 3 shows how many seconds this query took to execute, and how long it waited for table locks at the MySQL server level (not storage engine level), how many rows the query returned, and how many rows query examined. These lines are commented out, so that these won’t be executed into MySQL client. Last two lines are the actual queries. Similarly some more queries are shown below which are consider as slow queries because these queries are taking more than one second for executing query.

```
# Time: 130929 17:55:19
# User@Host: debian-sys-maint[debian-sys-maint] @ localhost []
# Query_time: 18.384700 Lock_time: 0.000213 Rows_sent: 0 Rows_examined: 429
SET timestamp=1380457519;
select count(*) into @discard from 'information_schema'.PARTITIONS;
```

```
# Time: 131002 10:33:06
# User@Host: debian-sys-maint[debian-sys-maint] @ localhost []
# Query_time: 11.543899 Lock_time: 0.000209 Rows_sent: 0 Rows_examined: 429
SET timestamp=1380690186;
select count(*) into @discard from 'information_schema'.PARTITIONS;
```

```
# Time: 130929 17:55:01
```
The first query is given the result for one course, five students, one quiz question conducted the online quiz. And the last query is given result for one course, 20 students, one quiz question conducted the online quiz. These queries are taking different times when increases the users. The last query is executing the slowly when load is increases. These type of queries we can found when increases the load. Above shown queries all are taking more than one second and some queries are exceeded the ten seconds also on the same configuration with same users. If we optimize these queries then it may give good performance.

**Webserver and database server**

Please maintain the Moodle database server and web servers separately, otherwise database may crash. The following screen you may get when database is crash.

![Figure A.3: Moodle Database deleted](image)

**A.2 edX**

**A.2.1 edX Installation Steps[8]**

Installation steps:
How to install edx-platform manually in ubuntu 12.04

______________________________

Set the proxy information

______________________________
set proxy information in /etc/apt.conf
Acquire::http::proxy "http://username:password@netmon.iitb.ac.in:80/";
Acquire::https::proxy "https://username:password@netmon.iitb.ac.in:80/";
set proxy information in ~/.bashrc file (set $http_proxy, $https_proxy)
出口 http_proxy=http://username:password@netmon.iitb.ac.in:80/
export https_proxy=https://username:password@netmon.iitb.ac.in:80/
set proxy environment variable in visudo file
$sudo visudo
Add below line in Defaults. (Note: Edit this file with carefully)
Defaults env_keep +="http_proxy https_proxy"
save and exit. (ctrl+o, enter. ctrl+x)
Installation
------------
Clone the edx-platform repository
---------------------------------
sudo apt-get update
sudo apt-get -y install git vim
mkdir ~/edx_all
cd ~/edx_all
git clone https://github.com/edx/edx-platform.git
cd edx-platform
Install EDX Requirements
---------------------------
sudo apt-get install git python-software-properties
sudo add-apt-repository -y ppa:chris-lea/node.js
sudo add-apt-repository -y ppa:chris-lea/node.js-libs
sudo add-apt-repository -y ppa:chris-lea/libjs-underscore
sudo apt-get -y update
sudo apt-get --purge remove ruby-rvm
curl -sL https://get.rvm.io | bash -s --ruby --autolibs=enable --auto-dotfiles
source /home/raju/.rvm/scripts/rvm
rvm install 1.9.3-p374
rvm use "1.9.3-p374@edx-platform" --create
rvm rubygems latest
gem install bundler
bundle install --gemfile Gemfile
export WORKON_HOME=$HOME/.virtualenvs
source /etc/bash_completion.d/virtualenvwrapper
mkvirtualenv -a "$HOME/.virtualenvs" --system-site-packages edx-platform

37
curl -sL -o numpy.tar.gz http://downloads.sourceforge.net/project/numpy/NumPy/1.6.2/numpy-1.6.2.tar.gz

curl -sL -o scipy.tar.gz http://downloads.sourceforge.net/project/scipy/scipy/0.10.1/scipy-0.10.1.tar.gz

tar xf numpy.tar.gz

tar xf scipy.tar.gz

rm -f numpy.tar.gz scipy.tar.gz

cd numpy-1.6.2

python setup.py install

cd ..

cd scipy-0.10.1

python setup.py install

cd ..

rm -rf numpy-1.6.2 scipy-0.10.1

cd $HOME/.virtualenvs/edx-platform/lib/python2.7/site-packages

curl -O http://pypi.python.org/packages/source/d/distribute/distribute-0.6.28.tar.gz

tar -xzvf distribute-0.6.28.tar.gz

cd distribute-0.6.28/

python setup.py install

cd ..

rm distribute-0.6.28.tar.gz

cd ..

pip install -r ~/edx_all/edx-platform/requirements/edx/pre.txt

cd ~/edx_all/edx-platform/

rvm use "1.9.3-p374@edx-platform"

sudo apt-get install phantomjs

npm config set strict-ssl false

npm config set registry "http://registry.npmjs.org/"

vim ~/edx_all/edx-platform/requirements/edx/github.txt

add https protocol with git protocol.

(example: -e git+https://github.com/edx/django-staticfiles.git@6d2504e5c8#egg=django-staticfiles)

rake install_prereqs

cd ~/edx_all

pip install argcomplete

cd ~/edx_all/edx-platform

bundle install

rake install_prereqs

cd ~/edx_all

mkdir db log data

cd ~/edx_all/edx-platform

rake django-admin[syncdb]

rake django-admin[migrate]

rake cms:update_templates

To run CMS

--------

sudo rake cms[dev,0.0.0.0:8001]
To run LMS
-----------
sudo rake lms[cms.dev,0.0.0.0:8000]

Note: you can't login in lms using email id password which you created at the time of installation.
Then, every time you're ready to work on the project, just run

$ workon edx-platform
$ cd ~/edx_all/edx-platform

Email Configuration

For CMS
vim ~/edx_all/edx-platform/cms/envs/common.py
Go to the email portion. Should start with a comment, # Email
Change EMAIL_BACKEND to 'django.core.mail.backends.smtp.EmailBackend'
Change the DEFAULT_FROM_EMAIL to the email you want to use. Similarly, for
DEFAULT_FEEDBACK_EMAIL and SERVER_EMAIL
Set EMAIL_HOST to the host of your choice
Set EMAIL_HOST_USER to your email id and EMAIL_HOST_PASSWORD to your password.
Set EMAIL_PORT to the port your server uses
Set EMAIL_USE_TLS to True
To Send mail using our IITB Ldap authentication
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
DEFAULT_FROM_EMAIL = 'raju@cse.iitb.ac.in'
DEFAULT_FEEDBACK_EMAIL = 'raju@cse.iitb.ac.in'
SERVER_EMAIL = 'raju@cse.iitb.ac.in'
EMAIL_HOST = 'smtp-auth.iitb.ac.in'
EMAIL_HOST_USER = 'ldapusername'
EMAIL_HOST_PASSWORD = 'ldappassword'
EMAIL_PORT = 25
EMAIL_USE_TLS = 'true'
ADMINS = (('edX Admins', 'raju@cse.iitb.ac.in'),)

For LMS edit the following file and follow the above step
vim ~/edx_all/edx-platform/lms/envs/common.py
Bibliography


[30] Shri Ram Pandey and Shweta Pandey. Developing a more effective and flexible learning management system (LMS) for the academic institutions using moodle, 2009.


