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

Moodle is one of the LMS, which is an Open Source. Our primary goal of this project is optimize the Moodle for improve the user response time. We are going to study detailed structure of Moodle, and its basic functionalities, available plugins, database server architecture. For finding Moodle database server performance bottleneck, we used some of loadbalancing, and real time query analysis tools. Then, we considered edX as our next LMS to study, because our institute is thinking to replace Moodle with edX, which is not decided yet. In this report, we presented architectural, functional, and different modules of edX.
## Contents

1 Introduction 1
   1.1 Motivation ......................................................... 1
   1.2 Learning Management System .................................... 1
   1.3 Feature comparison of LMSs ...................................... 3

2 Moodle LMS Introduction 6
   2.1 Moodle Major Functionalities ..................................... 7
   2.2 Architecture of Moodle Site ..................................... 8
   2.3 Available Users in Moodle ...................................... 8

3 Study of Moodle LMS 10
   3.1 Moodle three layer architecture .................................. 10
      3.1.1 Module wise explanation[11, 24, 33] ......................... 11
      3.1.2 Plugins in Moodle ........................................... 11
      3.1.3 LOC in Moodle .............................................. 12
   3.2 Moodle Database .................................................. 13
      3.2.1 Tables in Moodle Database .................................. 13
      3.2.2 Logical view of MySQL server architecture[32] .............. 14
      3.2.3 Database log .................................................. 15
      3.2.4 Security Level[11, 33] ...................................... 16
      3.2.5 Moodle LMS observations .................................... 17

4 Tools used in Moodle 18
   4.1 Jmeter .................................................................. 18
   4.2 mk-query-digest .................................................... 18
   4.3 phpMyadmin ........................................................ 18

5 edX Introduction 20
   5.1 Modules in edX ..................................................... 20
   5.2 edX Major Functionalities ......................................... 21
   5.3 edX database ......................................................... 22
   5.4 Preliminary experiments .......................................... 22
      5.4.1 quiz ............................................................ 22

6 Problem Statement 23
Chapter 1

Introduction

1.1 Motivation

Moodle LMS is an open source software, which is used in 239 countries all over the world. Moodle has a strong support of communities where the experts are involved in the discussion. Moodle LMS has performance issues with respect to user response time. 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 improve in many ways, i.e., by changing the Moodle architecture, hardware or software configuration, database, and so on.

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 take place asynchronous learning. e-learning is interactive learning, in which the learning content is accessible online, which offers feedback to students learning behavior[23].

E-learning is broadly inclusive of all forms of educational technology in learning and teaching[1]. e-learning is useful for students and teachers. It includes computer based learning like conducting online exams, and providing feedback to students, etc., which also tries to remove the time constraints and location constraints. It is suited to distance learning and flexible learning.

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

Figure 1.1: Comparison of LMS in traditional and e-learning[30]
distributed, 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 email or using discussion forums or chat rooms.

LMS is a software application for the administration, documentation, tracking, reporting and delivery of e-learning education courses or training programs[1]. Open and Distance learning mode is especially for working adults who need to support their families as well as manage their career while pursuing higher education.

**LMS Features**

The following features[13] 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 software’s are available in commercial as well as Open source software’s.

1. Open Source LMSs include Moodle, edX, SAKAI, WebCT, Bscw, Illias, eduplone, Claroline, ATutor, Dokeos and so on. These open source software’s have wide developer communities, which presents robust arguments for considering open source software, as a straightforward and potentially feasible competitor to commercial products[13, 14].

2. Commercial LMSs include Blackboard, Desire2Learn, KEWL, Blackboard Learning System, ANGEL Learning Management Suite, eCollege, etc.
1.3 Feature comparison of LMSs

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 exchanges, Emails, chat).

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

3. **Technical Specification Tools**:
   For better understanding, this is further divided into two kinds,
   (a) Hardware/software tools
   (b) Pricing/Licensing Tools

If the product doesn’t have cost, then that product has an advantage because we are mostly encouraging the open source software’s only.

![Figure 1.2: Summaries the features and capabilities[11]](image-url)
<table>
<thead>
<tr>
<th>Feature</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>Y</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 1.1: Learner tools[2, 11, 19, 26]

Moodle LMS is compared with other LMSs (edX, Blackboard, Desire2Learn, ANGEL, eCollege, Dokeos, Sakai) based on the above-mentioned features.

In Table 1.3, we consider no cost product as Yes, and with cost product consider as No.

**Summary**

We conclude that Moodle LMS supports almost all features excluding two, which are book marking and company profile, details are shown in table 1.1, 1.3. If we 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[26].

This report discusses Moodle LMS introduction in Chapter 2. In detailed structure of Moodle is explained in chapter 3. In Chapter 4, we are going to discuss the list of performance testing tools which are used in our project. Introduction of edX is discussed in chapter 5.
## 2. Support Tools

### 2.1 Administration Tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<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>

### 2.2 Course Delivery Tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<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>

### 2.3 Content Development Tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility Compliance</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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total features</th>
<th>16</th>
<th>16</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<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 1.2: Support tools[2, 11, 19, 26]

## 3. Technical Specifications

### 3.1 Hardware/Software

<table>
<thead>
<tr>
<th>Feature</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<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>

### 3.2 Pricing/Licensing

<table>
<thead>
<tr>
<th>Feature</th>
<th>Moodle</th>
<th>edX</th>
<th>Blackboard</th>
<th>Desire2Learn</th>
</tr>
</thead>
<tbody>
<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 1.3: Technical specifications[2, 11, 19, 26]
Chapter 2

Moodle LMS Introduction

Moodle is an open source software e-learning platform, which provides educator’s tools to create a course website[14, 20]. It is a web-based Learning Content Management System (LCMS) i.e. a CMS, and LMS, which provides information and collaboration among the Moodle Users such as System Administrators, developers, course designers, etc. Moodle 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 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 moodle is to help educators by creating online courses with focus of interaction and collaboration of content and continuous evaluation by learner’s 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..
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[31]. The main advantage of this Open Source Moodle LMS is that, anyone can modify according to their requirements, and can also add new modules.

![Figure 2.1: Moodle Architecture](image)

6
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[11]

1. Moodle is Open Source software. This means that anyone can download, and modify Moodle source code according to 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 downloaded 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 share 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.

2.1 Moodle Major Functionalities

The various features[1] of Moodle are as follows

1. Assignment submissions

2. Discussion forums

3. File download

4. Grading

5. Chat

6. Online calendar

7. Online news and announcement (College and course level)

8. Online quiz

9. Wiki

10. Multi language support

11. Modular, i.e., can be extended by creating plugins

12. Report (Which can be used for tracing or analyzing the user logs)
2.2 Architecture of Moodle Site

Moodle site architecture is shown in Figure 2.2. This shows the available people, administrations, summary of activities, etc. This is useful for understanding Moodle site.

![Moodle Site Architecture](image)

Figure 2.2: Moodle Site Architecture[11]

2.3 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 unrolling 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.
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.

6. **Guest:** This role has lowest permissions. This role can have permissions to see the public course or teachers name, 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, and submit the assignments, project work, etc..
- They can collaborate between students while doing project. They can ask question to the instructor in forums, which will make the communication between teacher and student effective.

**Faculty View**

Moodle allows the instructor to conduct 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 permissions. Administrator is capable of adding or deleting courses, users, etc. Admin can change any profile.

![Figure 2.3: Moodle LMS Authentication levels for Users](image)
Chapter 3

Study of Moodle LMS

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 plugin architecture.

Standard Moodle distribution includes core modules and different types of plugins. These are helpful for users to start teaching and learning immediately.

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 wikis, quizzes, etc.
2. Users
3. User functionalities
4. Logs and Statistics
5. Libraries

3.1 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. User interface refers to communication between software and human beings. For example, in Moodle, this is what we see and click on 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 maximize 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 Module wise explanation[11, 24, 33]

Moodle architecture is divided into six 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 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 1.1, and 1.2

3.1.2 Plugins in Moodle

Moodle is a collection of many different plugins instead of just single complex application. 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

- Activities: The main student activities in a course
- Users: Useful tools for managing users
- Reports: Useful for administrators, teachers and general users.
- Gradebook
- Cache
- Admin tools: Provides utility scripts useful for admins to examine and modify a Moodle site
- Blocks: Useful tools to add to courses or home pages
- Themes: Used to change look and feel of your site or course
- Course formats: Change the structure (or) layout of course pages
- Filters: Filters can process and change text
- General plugins: If any plugin do not fit into particular plugin, then that will go to general plugins
- Editors
- Messaging
- Repositories
- Plagiarism

### 3.1.3 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 \textbf{1.488\text{Millions}}. Lines of code for Moodle directory wise shown in the Table 3.1.

From Table 3.1, 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.
Table 3.1: 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>

3.2 Moodle Database

Moodle database schema contains many tables (more than 310 approximately). These tables are collection of core database tables and 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 abbreviation ActiveX Data Object. This is the database abstraction library for PHP. When you create a table, MySQL stores the table definition in a .frm file.

For analyzing the Moodle database server performance, we used different tools like Jmeter, mk-query-digest, and http-load. Detailed explanation of these tools is given in next chapter.

3.2.1 Tables in Moodle Database

The present Moodle database schema has 314 tables and uses InnoDB by default. Other storage engines like MyISAM, Merge MyISAM, etc., are also supported.

Following database tables are categorized[17] based on type of data they store.

1. Configuration
2. Users and Profiles
3. Roles and Capabilities System
4. Courses
5. Groups
6. Logging System
7. Blocks System
8. Events
9. Backup and restore
10. Statistics
11. Tags
12. Grade Book
13. Question Bank
14. Messaging System
15. Moodle Network
16. Caching
17. Miscellaneous
18. Activity Modules
19. Blocks
20. Question Types

3.2.2 Logical view of MySQL server architecture[32]

This will help us understand the server in the form of MySQL components. Figure 3.2 shows the logical view of MySQL architecture. In the Figure 3.2, the topmost layer contains services most network based client/server tools or servers need: connection handling, authentication, security.

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. 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 do not communicate with each other, they simply respond to the request from server.

Moodle Database Storage Engines:

Storage Engines are defined as a particular set of methods, 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 provides 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 use 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 Database log
As Moodle uses its own abstraction layer to convert the PHP queries to SQL queries, it is quite interested 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 does not show you, how many rows of MySQL are examined for executing queries.
We can find out where MySQL spends most of the time by using server profiling. MySQL has two kinds of query logs. For generating these logs, we need to do some changes in my.cnf file which is shown in Appendix A.1.6

1. General query log

2. Slow query log

General query log[32]: These log have all queries which are received by the server and the queries that may not be 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 this log. These general query logs do not contain the execution time of queries or after completion of query information. It also does not maintain general log. Log file records 554 queries per second while attempting quiz.

Slow query log[3, 32]: 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 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 between one to ten seconds respectively (If the query time is between the specified limit, or more than that query will be stored in slow log file) [3]. Slow query log will helpful for optimization because it contains queries which are taking a long time to execute.

The slow log file contains following metrics,

1. Query time: How many seconds query bring to execute
2. Lock time: How long query waited for table lock in MySQL server level (not at the storage engine level)
3. Rows sent: The number of rows the query returned
4. Rows examined: The number of rows query examined

Some sample output is shown in Appendix A.1.7.

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 does 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 [11, 33]

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. LMSs are clients/servers 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 [12] as follows

1. Authentication: further classification of authentication attacks is
(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 attack to make e-learning services and data is unavailable 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 exploit 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.

3.2.5 Moodle LMS observations

Following are the limitations of Moodle

1. Moodle can be operated by only IT experts[21]. According to Ajlan AL-Ajlan and Hussein Zedan, Moodle is complex for normal users to use and more than 66% of them are teachers, researchers and administrators.

2. While installation, users need to know the minimum technical terminology. Thus, it's difficult for beginners to install. Further details refer Appendix A.1.5.

3. 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.

4. Moodle requires course manager for generating, and providing online materials to instructors and technicians, else, Moodle will stay idle[11].

5. Moodle does not maintain the SSL functioning all over the site[12, 22, 33].

6. 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].

7. Moodle database server and webserver need to maintain seperately otherwise, it leads to database crash (or) security problems. This can be shown in Appendix A.1.5.
Chapter 4

Tools used in Moodle

For finding the bottlenecks of Moodle database server, we are going to use following tools, which are available as open source.

4.1 Jmeter

JMeter is a tool, that 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 desktop application. It is also 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 webservers. The detailed jmeter tool setup, and how to conduct the experiments on Jmeter is shown in Appendix A.1.3.

4.2 mk-query-digest

mk-query-digest[4] 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, and memcached. For more details, look at this Appendix A.1.4

mysqltuner:

Analyzes your database server performance and, based on the statistics it gathers, gives recommendations which variables you should adjust in order to increase performance.

For sample output is shown in appendix.

4.3 phpMyadmin

phpMyadmin[5] 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 also 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 login into phpMyAdmin, the user needs to provide username and password of MySQL. This tool is useful for us better understanding of Moodle database, and for easy way to add or delete the database tables, and also we can check the size of each table, etc,. The detailed installation steps of phpMyAdmin tool is shown in Appendix A.1.8
Chapter 5

edX Introduction

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 an MOOC named as edX which hosts courses from various institute across the world. 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.

5.1 Modules in edX

The various modules aims to implement various features of edX independently and are integrated together to provide a complete MOOC. Basic Modules in edX are as follows,

- **edX Platform**: This contains both LMS and CMS-Studio. In LMS, students are its main users, and it provides the user with interface to pursue online education. CMS has instructor as its 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 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 types 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. This modules shown in Figure 5.1
5.2 edX Major Functionalities

The following features are provided by edX:

1. Interactive video lectures with subtitles and indexing on subtitles (Downloadable).
2. Study materials like books, notes, cheat sheets, etc (Downloadable)
3. Online test of different types like video embedded quiz, practice sessions, mid term exam, final exam, etc.
4. Virtual Laboratory with interactive interface for user to view the expected simulation.
5. Calendar based schedule
7. Wiki edits for implementing collaborative learning.
8. Progress reports and other kinds of embedded analytics.
9. Different kinds of assessment systems for submitted assignments(open response problems). It includes:
   - Peer Grading
   - Self Grading
   - Staff Grading
   - Machine Grading
10. Emails and Notification facilities for registered student.
11. edX organization provides certification by the institutes through protected test exams.
12. Registering and deregistering from a course.
5.3 edX database

The edX database system uses SQLite and Mongo DB. 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, course and student modules, sample wiki data, enrollment data, etc. This is a relational database system, where data is organized in different tables. Mongo DB 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. Mongo DB is a document-oriented database (also known as NoSQL database) effective for scaling and high performance[7]. For classification of edX database tables is shown in appendix A.2.1

5.4 Preliminary experiments

5.4.1 quiz

Database tables that are accessed while conducting the quiz, we have conducted experiments on Moodle and edX. The quiz is consisting of two question which are Multiple choice question and short answer question. Resulted database tables are retrieved from mysql_log file and edx.log file. The database tables are accessed in Moodle LMS, and further more details of database table names and number of times each table accessed is shown in Appendix A.1.8
Chapter 6

Problem Statement

1. Moodle database server performance: According to Moodle system administrator team, Moodle webserver do not have any problem (They concluded based on conducted experiments like clustering webservers, changing the hardware configuration, increasing users), and also CPU, Memory resources of Moodle is under-utilized. According to Moodle administrator team, database is bottleneck. Further more details refer Appendix A.1.2.

So we are planning to find the bottleneck of the database server. For this, we need huge data required which contains the user accessed information. Every year, IIT Bombay conduct the ten thousand teachers program, for that they are using the Moodle to conduct exams. In this time, we are planning to save those peek time log results. Based on those peek time log results, we will identify bottleneck of database server using tools mentioned in our report. And we will resolve those problems. If Moodle does not give any progress after applying optimization techniques then we have another alternate method, which is replace Moodle with edX.

2. edX Scalability: Scalability of 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. We will come to know that, IIT Bombay advances usage of edX. There is quite possibility that, edX also may have some performance issues that we don’t know now.
Chapter 7

Conclusion and Future work

7.1 Conclusion

We studied Learning Management System, and its features. We did the feature comparison of LMSs. We studied in detailed structure of Moodle, and database tables. We conducted some preliminary experiments like conducting online quiz for better understanding of how Moodle works. We discussed some performance testing tools. Initially we started project with Moodle at that time, there is no edX. Sudden requirement of our institute, we are going to adopt edX. Amount of time able to spend on edX is very less. We done successful installation of edX, literature survey on edX, in detailed structure of edX which includes functionalities, modules, technologies.

7.2 Future Work

1. Next one or two months, we will findout the database server performance bottlenecks of Moodle, and we will resolve it.

2. We will investigate scalability of edX.

3. If time permits, we will look at into Moodle security issues.
Appendix A

Some details on Moodle and edX

A.1 Moodle

A.1.1 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, and PHP. In these we can’t modify code).

1. sudo apt-get install apache2 mysql-client 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 moodle db charset=utf8;
   - create user ’mdluser’@’localhost’ identified by ’password’;
   - grant all privileges on moodle db.* to ’mdluser’@’localhost’;
5. Setup Moodle server

A.1.2 Database server performance

According to DBP Moodle administrator team who conducted so-many courses with 3,000 users. Every November IIT Bombay conduct the online exams (Exam details, they conducted quiz with time limit of 20 minutes with in one hour at any time they can performed) for ten thousands teachers program. For
RAM 96GB
Processor Intel xeon
Operating System Ubuntu 12.04 LTS

Table A.1: DBP Moodle System Configuration

this, they need to conduct online exams with ten thousand users at same time but they are successfull till 3,000 users only. Later for based on their requirement they tuned hardware, and software configuration, and changed number of users from 3,000 to 5,000. At this time, Moodle goes down. For improving user response time, they performed different experiments like changed hardware configuration, and applied cluster concept on webserver but there is no progress. Observed that CPU, Memory is under utilized, most of the time spent on database server. Experiment is performed in following configuration,

For this reason, we are going to concentrate on database server. we studied database structure, technologies used in Moodle, we classified database tables, and some preliminary experiments for basic idea purpose. In the next month, we are planning to findout the bottleneck of database server.

A.1.3 Jmeter Tool Setup[10]

• 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

Different Moodle Modules

Moodle has a 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 A.2. These results are conducted based on the values of Table A.1. So, we need to optimize the database server, by applying the optimization techniques. Figure A.2 represents the Quiz module having high response ratio. The experiment compressed of five users, three quiz questions and the users accessed all modules simultaneously.

A.1.4 mk-query-digest sample output

Sample results of this tool is shown below

```
# Attribute total min max avg 95% stddev median
# =========== ========= ========= ========= ========= ========= =========
# Exec time 2s 120us 304ms 7ms 3ms 33ms 287us
# Lock time 193ms 30us 74ms 515us 490us 4ms 125us
# Rows sent 2.38k 0 481 6.51 7.70 38.31 0.99
```
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):
    # Profile
    # Rank Query ID Response time Calls R/Call Apdx V/M Item
    # == ============== =========== ======== ======== ======= ========= =
    # 1 0xE6F05D29E7F679E6 0.5581 22.4% 2 0.2791 1.00 0.00 SELECT
    information_schema.TABLES
    # 2 0xD65D7926B1674FD0 0.4232 17.0% 2 0.2116 1.00 0.00 SELECT
    information_schema.TABLES
    # 3 0x04248D594D4723A6 0.3790 15.2% 2 0.1895 1.00 0.03 SELECT
    information_schema.TABLES
    # 4 0x5CBC6BF26D494691 0.3493 14.0% 2 0.1746 1.00 0.00 SELECT
    information_schema.TABLES
    # 5 0xF0A8832C977319D 0.1954 7.8% 1 0.1954 1.00 0.00 SELECT
    information_schema.TABLES
    # 6 0xF704AFA9EB5B6F 0.1202 4.8% 3 0.0401 1.00 0.07 SELECT
    information_schema.COLLATIONS
    # 7 0x6790AAB18EE3769D 0.0865 3.5% 7 0.0124 1.00 0.06 SELECT mdl_filter_active
    mdl_context mdl_filter_active
    # 8 0x55A3D07A1362151C 0.0622 2.5% 3 0.0207 1.00 0.04 SELECT
    information_schema.CHARACTER_SETS
    # 9 0x6E76254FA412009 0.0592 2.4% 8 0.0074 1.00 0.03 SELECT
    mdl_block_instances mdl_block mdl_block_positions mdl_context
    # 10 0x3528B8BD10F8D070 0.0497 2.0% 16 0.0031 1.00 0.01 SELECT
    mdl_forum_posts mdl_forum_discussions mdl_forum mdl_user
    # 11 0x467A0AC5DE5ACC7A 0.0447 1.8% 1 0.0447 1.00 0.00 SELECT
    UNION mysql.db mysql.user
    # 12 0xA4B8CA0B0029F748 0.0194 0.8% 6 0.0032 1.00 0.01 SELECT
    information_schema.ROUTINES
    # 13 0x81B87C1DA9E93F44 0.0108 0.4% 4 0.0027 1.00 0.00 SELECT
    mdl_block_instances mdl_block mdl_block_positions mdl_context
    # 14 0xAEE7609E9F14B62 0.0090 0.4% 8 0.0011 1.00 0.00 SELECT
    mdl_filter_active mdl_context mdl_filter_config
    # MISC 0xMISC 0.1234 5.0% 309 0.0004 NS 0.0 <278 ITEMS>
```

A.1.5 LOC of Moodle

I wrote script for finding the lines of code for Moodle that code(loc.py) is shown below
flag = False
counter = 0
f = open(f, "r")
for line in f.read().split(\n):
    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
Moodle Quiz database

Figure A.3: Moodle Quiz database tables[17]
A.1.6 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.1.7 MySQL Log file results

When i run the Moodle on configuration shown in table A.1) 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 when query was logged, second line shows who executed this query, line three shows how many seconds this query took to execute, and how long it waited for table lock 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 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 slowly when load is increases. These types 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.

A.1.8 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 administration 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

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.4: Moodle Database deleted

**Preliminary experiment results**

Moodle database tables is shown for quiz activity as table name and number of times each table accessed

1. mdl_context–204
2. mdl_config–171
3. mdl_course–161
4. mdl_config_plugins–139
5. mdl_sessions–81
6. mdl_role–78
7. mdl_cache_text–60
8. mdl_enrol–41
9. mdl_quiz–40
10. mdl_filter_active–38
11. mdl_groups–36
12. mdl_quiz_attempts–31
13. mdl_modules–30
14. mdl_grade_items–28
15. mdl_course_format_options–25
16. mdl_log–23
17. mdl_user_preferences–22
18. mdl_capabilities–21
19. mdl_cache_flags–19
20. mdl_message–18
21. mdl_course_modules–18
22. mdl_role_capabilities–16
23. mdl_quiz_overrides–14
24. mdl_event–12
25. mdl_course_categories–12
26. mdl_role_assignments–10
27. mdl_block_instances–10
28. mdl_repository–9
29. mdl_user_lastaccess–8
30. mdl_quiz_grades–6
31. mdl_quiz_grades–6
32. mdl_question–5
33. mdl_forum–4
34. mdl_forum_discussions –4
35. mdl_forum_posts–4
36. mdl_quiz_feedback–4
37. mdl_question_answers–4
38. mdl_question_hints–4
39. mdl_group_members–2
40. mdl_user_info_field–2
41. mdl_grade_settings–2
42. mdl_question_multichoice–2
43. mdl_question_shortanswer–2
44. mdl_event_handlers–2
45. mdl_events.queue_handler–2
46. mdl_external_tokens–1
A.2 edX

A.2.1 edX Installation Steps[9]

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)
-----------------------------------------------
export 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 -y install gfortran graphviz libgraphviz-dev graphviz-dev libatlas-dev libblas-dev pkg-config liblapack-dev

sudo apt-get --purge remove ruby-rvm
sudo rm -rf /usr/share/ruby-rvm /etc/rvmrc /etc/profile.d/rvm.sh
curl -sL https://get.rvm.io | bash -s stable --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
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 -0 http://pypicenter.org/packages/source/d/distribute/distribute-0.6.28.tar.gz
tar -xzf distribute-0.6.28.tar.gz
cd distribute-0.6.28/
python setup.py install
cd ..
rm distribute-0.6.28.tar.gz

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

Tables in edX:

The edX platform uses the django framework for development and production of the edX platform. It inherits the djangos legacy database. The edX platform consist of total 88 tables including djangos tables. These tables are classified as follows,

1. Authentication
2. Bulk_emails
3. Celery
4. Certificates
5. Circuit
6. Course
7. Courseware
8. Djcelery
9. Django
10. Licenses
11. Notify
12. Shopping Cart
13. Tracking
14. Verify
15. Student
16. Wiki
Bibliography


