Optimization of LMS for Improving User Response Time

M. Tech. Project Stage-2 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

June 2014
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**, **Abhilash**, **Silpa** for their 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** and **Firuza** for proof reading this report and providing their valuable feedback.
Contents

1 Introduction ............................................. 1
   1.1 Moodle Three Layer Architecture .................... 1
   1.2 Moodle Database .................................... 1
       1.2.1 Tables in Moodle Database ..................... 2
       1.2.2 Logical view of MySQL server architecture[8] .... 3
       1.2.3 Log file analysis ................................ 3
       1.2.4 MySQL configuration ............................ 4

2 Performance Testing Tools and Related work ............... 5
   2.1 Testing tools ....................................... 5
       2.1.1 Jmeter ........................................ 5
       2.1.2 mk-query-digest ................................ 5
       2.1.3 PhpMyadmin .................................... 5

3 Apply ................................................... 7
   3.1 Optimization Techniques ............................ 7
       3.1.1 Frontend ....................................... 7
       3.1.2 Backend ....................................... 7

4 Results .................................................. 8

5 Conclusion and Future work ................................ 9
   5.1 Conclusion .......................................... 9
   5.2 Future Work ........................................ 9
Chapter 1

Introduction

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 educator’s tools to create a course website[5, 7]

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.

1.1 Moodle Three Layer Architecture

Moodle can be represented in a three layer architecture as shown in the Figure ?? . 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 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[6]. 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.

1.2 Moodle Database

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 abbreviation 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. Detailed explanation of these tools is given in chapter

1.2.1 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. When you create a table, MySQL stores the table definition in a .frm file.

These tables are categorized[6] 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
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
1.2.2 Logical view of MySQL server architecture

This will help us understand the server in the form of MySQL components. Figure ?? shows the logical view of MySQLs architecture. In the Figure ??, 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.

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**
   - 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 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.

1.2.3 Log file analysis

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.

1. **General query log**
2. Slow query log

**General query log[8]:** These log 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 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[2, 8]:** 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)[2]. 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 the 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

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.

1.2.4 MySQL configuration
Chapter 2

Performance Testing Tools and Related work

For finding the bottlenecks of Moodle, we used the following tools, which are available as open source.

2.1 Testing tools

Here, we are discussed a brief introduction of some performance tool in Moodle and its usage.

2.1.1 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 ??.

2.1.2 mk-query-digest

mk-query-digest[3] 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 ??

2.1.3 PhpMyadmin

phpMyadmin[4] 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 MoodleDB, 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 ??
Chapter 3

Apply

3.1 Optimization Techniques

3.1.1 Frontend
Disbling themes, sessions,

3.1.2 Backend
mdl session database table
list of those SQL queries

- Using Memcache
- Deleting content of DB table
- Storing session into file
Chapter 4

Results
Chapter 5

Conclusion and Future work

5.1 Conclusion

5.2 Future Work
Bibliography


