

Optimization of LMS for Improving user Response Time

Sabareeshwaran^[1], Muralidharan^[2]

Department of Information Technology

Dr. Mahalingam College Of Engineering and Technology
Pollachi

Abstract- Moodle is one of the LMS, which is an Open Source. Moodle has serious performance issues where number of participants attempting quiz at same time or collecting feedback from participants, etc. The entire optimization was tested on load testing Moodle system, identifying bottleneck and removing bottlenecks. The major bottleneck while performing load testing was acquiring DBMS lock to update the Moodle sessions table. The major project aim to come up with alternative logging strategy so that session update information is inserted as new record. And deletion is done in bulk so that the logging process works faster. The purpose of this section is to provide the reader with a general, background information and an insight into the web application of “Optimized Moodle Learning Management System” and is to specify the requirements for policy enforcement framework for moodle. When moodle is being used for conducting quizzes or exams in schools, only the quiz or exam related apps should get open. Any request to start any of the remaining apps should be blocked.

Keywords- Moodle, Open source, MYSQL, LSM

I INTRODUCTION

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.

1.1 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 behaviour.

E-learning is broadly inclusive of all forms of educational technology in learning and teaching 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.

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 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.

1.2 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 educator's tools to create a course website.

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.

Some of the reasons for using Moodle are 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, Easy to learn and provides high quality online courses. Instructors are sharing documents/lecture materials, graded assignments discussions forums, etc., among students, Provides good educational tools/plugin, and easy to install and Moodle allows users to post news items, assignments, electronic journals and resources, and to collect assignments, etc.

1.3 LMS Features

LMS softwares are available in commercial as well as open source softwares. Open Source LMSs include Moodle, edX, SAKAI, Web CT, Bscw, Ilia's, 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. Commercial LMSs include Blackboard, Desire2Learn, KEWL, Blackboard Learning System, ANGEL Learning Management Suite, eCollege, etc.

1.4 Moodle Three Layer Architecture

Moodle can be represented in a three layer architecture. 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. The design of such an interface requires collaboration between software. Users make this user-friendly and maximize overall usability.

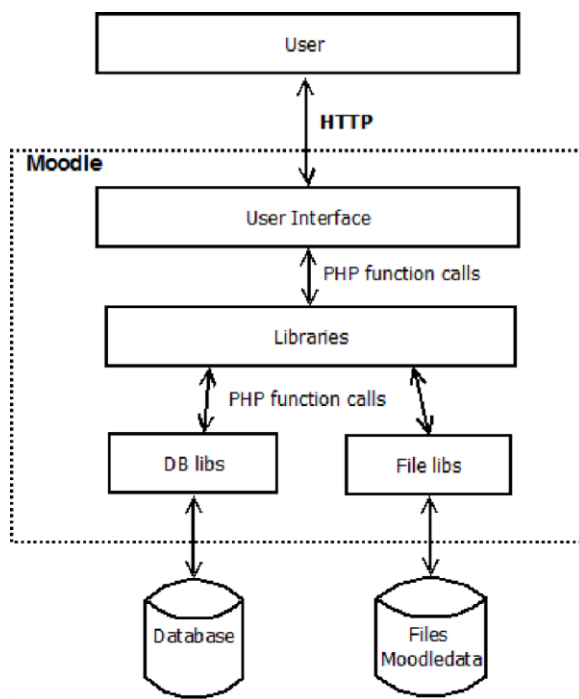


Fig 1:Moodle 3 layer architecture

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 moodle data directory.

II EXISTING SYSTEM

Moodle LMS contains some performance issues. According to DBP Moodle system administrator, DBP moodle supports 2500 users at a time for conducting online quiz with configuration of 96GB of RAM, 100 TB HDD. When we run moodle on normal PC then we found that moodle database is using database locks for updating records. We planned to remove those locks from moodle database. These updation is done on moodle session database table which is accessed by user frequently. While updating these, other queries are going to sleep state till

completion of updating session record in moodle session table. We decided to remove those locks so that we can improve Moodle LMS performance.

When we were running 400 users on server then we found the observations which described in figure Most of the CPU(433.9%) is spend on processing mysql requests. This gave us the basic way to find the bottleneck of Moodle LMS.

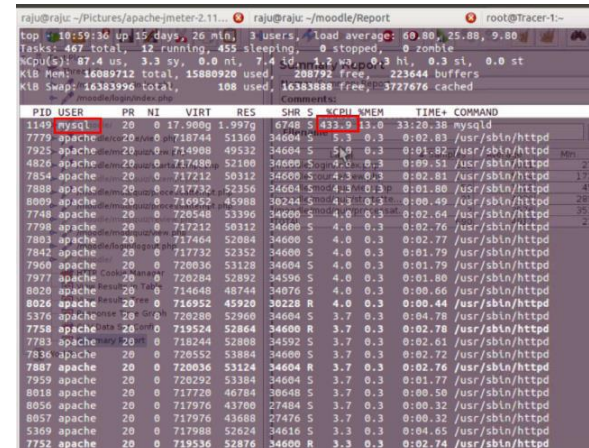


Figure 2: CPU usage

2.1 Existing Functionalities

Whenever user try to login to moodle, user session is created in moodle sessions table. In index.php internally calling the moodlelib.php for user-login details which is complete user-logging (\$USER). This function internally checks login user details which is described in moodle database.php file and manager.php. Initially manager.php generates session of every user (user can be either guest, admin, student, etc.) in moodle session table.

```

mysql> desc mdl_sessions_temp;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | bigint(10) | NO | PRI | NULL | auto_increment |
| state  | bigint(10) | NO |     | 0 | |
| sid    | varchar(128) | NO |     | | |
| userid | bigint(10) | NO |     | NULL | |
| sessdata | longtext | YES |     | NULL | |
| timecreated | bigint(10) | NO |     | NULL | |
| timemodified | bigint(10) | NO |     | NULL | |
| firsttip | varchar(45) | YES |     | NULL | |
| lasttip | varchar(45) | YES |     | NULL | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
  
```

Figure 3: Moodle session temp table

2.2 Logging Module

Each user will have his own username and password through which he can login into the moodle and take up a test or do any other activities regarding courses.

Log in

Username:

Password:

☐ Remember username

[Forgotten your username or password?](#)

Cookies must be enabled in your browser

Figure 4: logging module

2.2.1 Categorising

Using the import option in the moodle we can import the files required for the course in any format such as aiken,

Blackboard, exam view, gift format, etc. Here we have used aiken format questions which has been saved with encoding UTF-8 and imported in moodle in order to have the test.

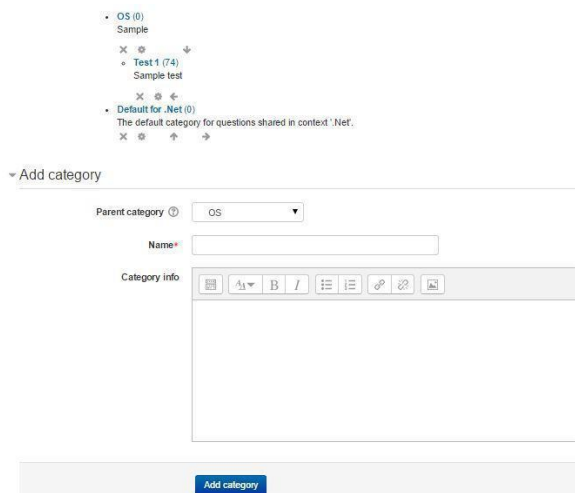


Figure 5: Categorising

III LMS OPTIMIZATION

3.1 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.

3.1.1 Tables

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 based on type of data they store.

3.1.2 MYSQL Server Architecture

This will help us understand the server in the form of MySQL components. Shows the logical view of MySQLs architecture. In the Figure 4.1, 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.

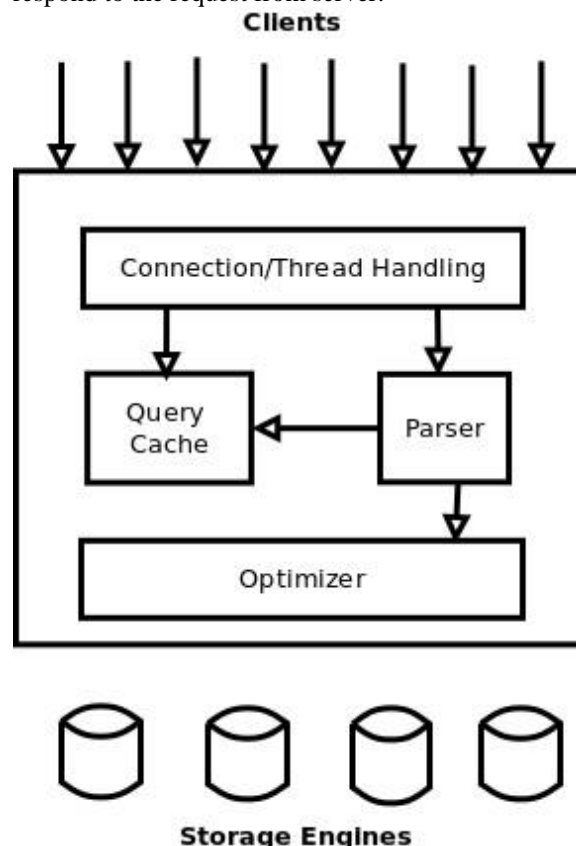


Figure 6: Logical view of MySQL server architecture

IV IMPLEMENTATION AND MODULES

For finding the bottlenecks of Moodle, we used the following tools, which are available as open source. Here, we discuss a brief introduction of some of the tools used in Moodle and its usage.

4.1 JMeter

JMeter is a tool, that is 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, or simulate a heavy load on a server and measure 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 web servers.

4.1.1 JMeter settings

Before going to start load testing, following configurations need to be setup for running large number of users. Create configure element which is a .csv data set of username and passwords of users. Accessing these username and password is done by creating variables with USER and PASS. Wherever credentials required define these

variables. JMeter reads data from csv file using these variables

4.2 Sesskey

```

1197 <collectionProp name="Arguments.arguments">
1198   <elementProp name="username" elementType="HTTPArgument">
1199     <boolProp name="HTTPArgument.always_encode">false</boolProp>
1200     <stringProp name="Argument.name">username</stringProp>
1201     <stringProp name="Argument.value">${USER}</stringProp>
1202     <stringProp name="Argument.metadata">=</stringProp>
1203     <boolProp name="HTTPArgument.use_equals">true</boolProp>
1204   </elementProp>
1205   <elementProp name="password" elementType="HTTPArgument">
1206     <boolProp name="HTTPArgument.always_encode">false</boolProp>
1207     <stringProp name="Argument.name">password</stringProp>
1208     <stringProp name="Argument.value">${PASS}</stringProp>
1209     <stringProp name="Argument.metadata">=</stringProp>
1210     <boolProp name="HTTPArgument.use_equals">true</boolProp>
1211   </elementProp>
1212 </collectionProp>
1213 </elementProp>
1214 <stringProp name="HTTPSampler.domain">10.129.50.155</stringProp>
1215 <stringProp name="HTTPSampler.port"></stringProp>
1216 <stringProp name="HTTPSampler.connect_timeout"></stringProp>
1217 <stringProp name="HTTPSampler.response_timeout"></stringProp>
1218 <stringProp name="HTTPSampler.protocol">http</stringProp>
1219 <stringProp name="HTTPSampler.contentencoding">utf-8</stringProp>
1220 <stringProp name="HTTPSampler.path">/moodle/login/index.php</stringProp>
1221 <stringProp name="HTTPSampler.method">POST</stringProp>

```

Figure 7 Sesskey

Sesskey is randomly generated number while accessing moodle for preventing security attacks like session hijacking. This sesskey is extracted by using regular expressions. Usage of sesskey is shown in figure 7.

4.3 PhpMyadmin

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 also need to install web server (Such as LAMP (Linux,Apache,MySQL,PHP)) because phpMyAdmin interface is entirely accessed on browser.

4.4 Optimizing Moodle LMS To Improve User Response Time

Web Applications are now required in each and every industry, including business, education, tourism, entertainment and many more. The Objective of our project is to optimize Moodle LMS to reduce average user response time by employing numerous front end and back end optimization techniques.

V EXPERIMENTS

5.1 Front-End Optimization Using Image Maps

In its simplest form, a hyperlink associates the destination URL with some text. A better alternative is to associate the hyperlink with an image, for example in navigation bars and buttons. If you use multiple hyperlinked images in this way, image maps may be a way to reduce the number of HTTP requests without changing the page's look and feel. An image map allows you to associate multiple URLs with a single image. The destination URL is chosen based on where the user clicks on the image. I have created a set of 10 pages with image map and another set of 10 pages without image.

5.2 Front End Optimization by Using Far Future Expires Header

Browsers (and proxies) use a cache to reduce the number of HTTP requests and decrease the size of HTTP responses, thus making web pages load faster. A web

server uses the Expires header to tell the web client that it can use the current copy of a component until the specified time. Moodle sends requests with an Expires Header which is set in past (20th Aug 1969 09:23 GMT). After changing it to 20th Aug 2015 09:23 GMT.

VI RESULTS

The following results represents the response time of start attempt page which is taking more time and one more page which is processing page while conducting quiz. And also mentioned that average response time of all the pages while attempting quiz. One of the column represents error rate which is contains the error rate of overall quiz. These details retrieved by running test case individually by changing the number of users.

6.1 Summary Report

This test case run on without optimized moodle by changing number of users.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
moodle	50	3468	1487	5418	779.16	0.00%	7.7/sec	41.48	5488.0
moodlelogin...	100	7885	1067	16867	6283.15	0.00%	4.6/sec	28.20	6345.6
moodlecourse...	50	8133	4552	10438	2183.65	0.00%	2.5/sec	17.68	7374.8
moodlemod...	50	6617	2813	8935	1720.43	0.00%	2.3/sec	14.47	6541.1
moodlemod...	50	11895	9263	16536	1374.03	0.00%	1.7/sec	11.38	6925.4
moodlemod...	1000	8580	3382	12445	1086.46	0.00%	5.1/sec	34.24	6901.3
moodlelogin...	50	3097	523	5301	1547.28	0.00%	2.4/sec	15.10	6258.9
TOTAL	1350	81.70	523	16867	2602.32	0.00%	5.0/sec	37.20	6792.8

Figure 8: 50 users run on 2GB RAM in without optimized moodle

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
moodle	75	7209	3427	10459	2593.44	0.00%	7.1/sec	38.14	5488.0
moodlelogin...	150	13173	462	28554	12818.93	0.00%	3.6/sec	22.35	6347.5
moodlecourse...	75	6808	4156	8697	1081.33	0.00%	5.1/sec	36.55	7376.3
moodlemod...	75	4214	348	8138	2547.62	0.00%	4.7/sec	28.35	6541.1
moodlemod...	75	15045	9915	21783	4088.32	0.00%	3.2/sec	21.54	6919.7
moodlemod...	1500	7596	428	13310	2324.88	0.00%	5.3/sec	35.90	6909.7
moodlelogin...	75	4171	842	8736	2464.07	0.00%	2.5/sec	15.35	6358.9
TOTAL	2025	8034	348	28554	5184.22	0.00%	5.3/sec	34.89	6796.0

Figure 9: 75 users run on 2GB RAM in without optimized moodle

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
moodle	100	7076	788	10635	3284.79	0.00%	6.5/sec	45.91	5488.0
moodlelogin...	200	15139	344	25888	13759.25	0.00%	4.0/sec	24.58	6347.5
moodlecourse...	100	6854	6272	10895	802.22	0.00%	3.8/sec	28.16	7374.8
moodlemod...	100	5978	3036	9187	1856.67	0.00%	4.1/sec	25.95	6476.0
moodlemod...	100	14368	7942	16095	1057.48	0.00%	4.1/sec	27.83	6919.7
moodlemod...	2000	9940	403	16135	3656.24	0.00%	6.1/sec	41.03	6909.7
moodlelogin...	100	4503	455	8783	3279.96	0.00%	5.0/sec	31.10	6358.9
TOTAL	2700	6971	344	25888	5452.45	0.00%	6.2/sec	41.39	6797.2

Figure 10: 100 users run on 2GB RAM in without optimized moodle

VII CONCLUSION AND FUTURE WORK

REFERENCES

7.1 Conclusion

We removed the locks successfully on moodle session database table. We did the correctness of our optimized solution using sanity testing. We used tool that is JMeter for load testing of moodle with different parameters, and configure elements like summary report. mk-query-digest tool used for finding number of times that particular sql query was accessed. In place of locks, we used one database table which is used for storing sessdata and at the end,

we cron temp table into original table. This improved moodle supports approximately 30% of extra users compared to already existed one.

7.2 Future Work

Whenever system administrator or instructor wants to check the reports of students then they are able to see either that particular day or all days logs. Instructor or system administrator is not able to see the details of particular range of days. Here, the problem is whenever teacher or instructor selects all days logs, then it retrieves old users details also.

This is unnecessarily retrieving old users(whoever not active in that particular course or semester) which is effecting performance or it kills the instructor patience. If this option can be done with optimized queries, then this will give good performance.

Moodle sessions are not necessary if we are able to redirect the users whenever they are going to open new tab with moodle page then we no need to maintain sessions. This can be done only when we are going to conduct quizzes or any other thing.

- [1]<http://dev.mysql.com/doc/refman/4.1/en/slow-query-log.html>. [Online; accessed 3 October 2014].
- [2]<http://www.maatkit.org/doc/mk-query-digest.html>. [Online; accessed 2September-2014].
- [3] <http://docs.phpmyadmin.net/en/latest/>. [Online; accessed 5 th-August -2013].
- [4]Sessionhandling.http://docs.moodle.org/26/en/Session_handling, Accessed on 21st April 2013.
- [5] Jmeter tool. <http://jmeter.apache.org/>, 2001. [Online; accessed 15-July -2013].
- [6] Jmeter tool. <http://jmeter.apache.org/>, 2001. [Online; accessed 2-October -2013].
- [7] A. Al-Ajlan and H. Zedan. Why moodle. In *Future Trends of Distributed Computing Systems, 2008. FTDCS '08. 12th IEEE International Workshop on*, pages 58–64, 2008. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4683115, Accessed on 15th July 2013.
- [8] Cansu Cigdem Aydin and G Tirkes. Open source learning management systems in e-learning and moodle. In *Education Engineering (EDUCON), 2010 IEEE*, pages 593 – 600. IEEE, 2013.
- [9] Sadanand Y Bansode and Rajendra Kumbhar. E-learning experience using open source software: Moodle. *DESIDOC Journal of Library & Information Technology*, 32(5), 2012. <http://ieeexplore.ieee.org/Xplore/home.jsp>, Downloaded on 20th June 2013.
- [10] Tao Chen, Arif Khan, Markus Schneider, and Ganesh Viswanathan. iblob: Complex object management in databases through intelligent binary large objects. In Alan Dearnle and RobertoV. Zicari, editors, *Objects and Databases*, volume 6348 of *Lecture Notes in Computer Science*, pages 85–99. Springer Berlin Heidelberg, 2010.
- [11] Ludmila Cherkasova. Scheduling strategy to improve response time for web applications. In *Proceedings of the International Conference and Exhibition on HighPerformance Computing and Networking, HPCN Europe 1998*, pages 305–314, London, UK, UK, 1998. Springer-Verlag.
- [12] Francisco Corbera, Eladio Guti  rrez, Juli  n Ramos, Sergio Romero, and Mar  a A Trenas. Developing an effective and efficient elearning platform using open source software. In *ACM SIGCSE Bulletin*, volume 40, pages 349–349. ACM, 2008.