

Comparison of LMSs:

Which is the most suitable LMS for my needs?

<http://dx.doi.org/10.3991/ijet.v8iS2.2758>

A. C. Caminero, R. Hernández, S. Ros, Ll. Tobarra, A. Robles-Gómez, R. Pastor
Universidad Nacional de Educación a Distancia, UNED Madrid, Spain

Abstract—The role of Learning Management Systems (LMSs) is very important when conducting educational activities, specially in the case of distant education – when its use becomes essential since all the interactions among the participants in the learning/teaching process are conducted through it. There are a number of tools to provide this functionality, both open-source (such as .LRN, Sakai, or Moodle) and proprietary solutions (such as Blackboard).

Given the wide variety of existing LMSs, it is worthwhile to devote effort in order to figure out *which LMS is the most suitable for my needs?* This paper strives at answering this question by presenting a twofold evaluation that compares three of the most widely used open-source LMS, namely .LRN, Sakai, and Moodle. First, a performance evaluation is presented that shows the different levels of use of the hardware resources (e.g. CPU, memory, I/O) of the machine the LMS is running on. Second, a performance evaluation from the point of view of the system administrators is also presented.

Index Terms—Comparison of LMS, suitable LMS, needs

I. INTRODUCTION AND MOTIVATION

In order to support learning/teaching processes, Learning Management Systems (LMS) have been developed. LMSs provide a number of tools, among others, communication tools such as video-conferencing [1], forums or email, evaluation tools such as questionnaires, or grading tools. The use of a LMS, being very interesting for any kind of learning, becomes totally necessary in the case of distance education, since they provide the resources needed to communicate the participants in the learning/teaching process [2].

Several open-source LMSs exist, such as Moodle [3], Sakai [4] or .LRN [5], but also proprietary LMSs such as Blackboard [6]. When deciding about the use of a LMS, an important point is the decision of which LMS should be used for each specific case. This decision must be made taking into account objective data such as the levels of use of the hardware on which the LMS will be hosted, and subjective information such as the experience and opinion of users of the system.

The main aim of this paper is provide hints to choose a LMS. For this, an extensive performance evaluation involving three widely-used open-source LMSs, namely Moodle, Sakai, and .LRN has been conducted. A twofold evaluation, from the hardware utilization and the system administrator points of view, is presented. For this work, these three technologies have been installed, and courses have been developed in each of them. Experiments involving different numbers of concurrent users performing different tasks (such as writing messages in

forums, or taking assessments) have been conducted in order to test the performance of each LMS. Experiments have been repeated several times and average results have been calculated, in order to avoid spurious events that may blur the results obtained in the experiments. Objective measures such the memory or CPU consumption have been collected. Based on these measurements, and along with user experience (from the system administrator point of view), this paper ranks the three technologies involved in this study.

This paper is structured as follows: Section II briefs the LMSs under study; Section III presents the testing tool used to conduct the experiments, and describes the test plans developed for this work; Section IV presents the experiments and results; Section V draws conclusions and suggests guidelines for future work.

II. SURVEY OF LMSs

The three technologies under review in this work are shown in Figure 1. They were chosen because of their wide-spread use and their open-source nature.

The .LRN platform [5] is written on Tcl and works under an AOLServer [7]. Sakai [4] is written in Java and works under the servlet container Apache Tomcat [8]. Meanwhile, Moodle [3] is written in PHP and can be run using any web server, such as Apache HTTP Server [9].

In order to perform efficient comparison between all the LMSs, their system configuration should be as similar to each other as possible. To this end, a number of implementation decisions were made. First of all, all the LMS use their default configurations – optimization of each LMS would be the aim of future work.

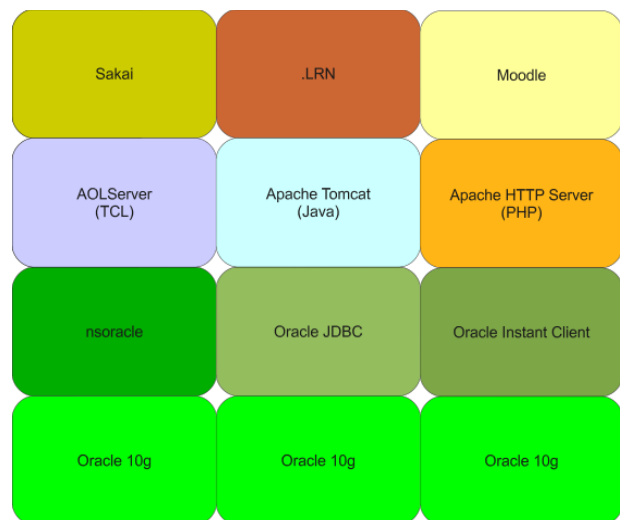


Figure 1. Software architecture of the tested LMSs.

Besides, we decided to use the same database for all the LMSS, and the chosen one was Oracle 10g. This database was chosen because it is supported by all the LMSS, so all of them would share the same database configuration, which means that none of them would be in better conditions than the others with regard to the database.

III. EXPERIMENTS TOOLS

In a LMS under real usage, the system would have high numbers of users (hundreds or even thousands of faculty, students, and system administrators) interacting with the system at all times. In order to perform the experiments needed for our work, we need to study the usage of a real LMS. For this we have two options, namely (1) using real users, and (2) using a tool that simulates the use of real users. The use of real users for our experiments becomes impossible for large numbers of users, and for the need to conduct experiments in a repeatable and controlled manner.

Once the need to use a tool to simulate concurrent users becomes apparent, we have to choose the right tool to simulate a real LMS usage. A number of tools are available for our purposes, among others, Apache JMeter [10], ab – Apache HTTP Server Benchmarking Tool [11], Badboy [12], HP LoadRunner [13], or IBM Rational Performance Tester [14], among others. We have analyzed these tools and chosen one of them to conduct the experiments of this work.

The chosen tool is Apache JMeter [10], among others, because it is open-source and widely used, with a lively community that is of great help to tackle any difficulty; it has been developed for testing web applications, its Graphical User Interface (GUI) is friendly and easy to use; its learning curve is low, and deep knowledge can be acquired easily; and it is extensible, so third party developments can be used to cover functionalities not covered by the original tool.

In order to perform the experiments for this work, JMeter has been extended with JMeter Plugins [15], a third party tool. This way, JMeter can be used, for instance, to monitor the performance of a server in real time. To this end, its developers have created a monitor, called *PerfMon*, to be executed in the server, which will be entrusted with the data collection for JMeter.

This monitor uses an open-source library called *System Information Gatherer And Reporter (SIGAR)* which provides a portable interface which allows information gathering on memory usage, swap, CPU usage, filesystems... This information is available in most operating systems, but each OS has its own ways to provide it. SIGAR provides developers with one API to access this information regardless of the underlying platform.

Once the right tool is chosen, test plans have been developed to try the main functionalities of the LMS, such as reading and posting messages in the forums, downloading material, or taking questionnaires. Due to the heterogeneity of the LMSS under study, test plans for each LMS were highly different and the testing plan used for one LMS could not be reused for another LMS.

Test plans were created based on a number of scenarios, which have been chosen because from our experience these are the most used functionalities of a LMS.

Recall we work for a totally distance university, so the only way of interacting between students and faculty is by means of the university LMS. So, our experience with LMS tools is one of our most important asset for our work as educators. The scenarios chosen are login/logout, write in a forum, submit a task, download a file, schedule an event using the calendar, and take a questionnaire.

Each test plan includes all the functionalities mentioned above and is repeated ten times, where average results are calculated and presented in this paper. This way, we try to avoid spurious events that blur our experiments.

Once the scenarios object of our study were selected, the next point in our work was the definition of each scenario in each LMS, in order to reproduce its execution in an automatic way by means of JMeter. To this end, a tool called *JMeter HTTP Proxy Server* was used, which works as a proxy between the user and the LMS. The user browser requests are sent to the proxy, which in turns forwards them to the server. The use of this tool allows us the recording of the browsing session for its analysis and reproduction.

The JMeter proxy server was of great interest to record the browsing routines needed to implement each task, but there is still an important point, which is related to the dynamic information that is generated during the browsing session. This dynamic information allows the browsing control, record session data, and other elements used by the application. These elements are generated using varying parameters such as the date and time. For these reasons, the use of an extractor of regular expressions was needed to obtain such dynamic information. Figure 2 presents a screenshot of the regular expressions extractor for the time a HTTP request is performed, where we can assign a name to the regular expression, a reference name (*Nombre de referencia*) which will be the name of the variable which will keep the extracted value, the values that will be retrieved (*Plantilla*), and the regular expression itself. Also, Figure 3 presents a HTTP request that sends a variable named *time* whose value has been obtained using the aforementioned regular expression.

A. Test plans

JMeter test plans are made of several elements. The first element is a *ThreadGroup*, which tells JMeter how many users we are going to simulate, in which time intervals they are going to send their queries, and how many requests they are going to send. Inside the thread-group we can configure the default values for the HTTP requests, such as the server which will be the target of the HTTP requests, the port where the server is listening, different timeouts, or a proxy server if needed.

Figure 2. Regular expressions extractor for the time stamp of a request.

SPECIAL FOCUS PAPER
COMPARISON OF LMSS: WHICH IS THE MOST SUITABLE LMS FOR MY NEEDS?

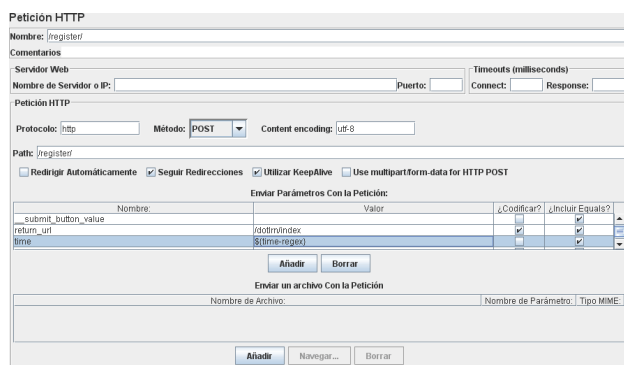


Figure 3. HTTP request using the result of the regular expression presented in

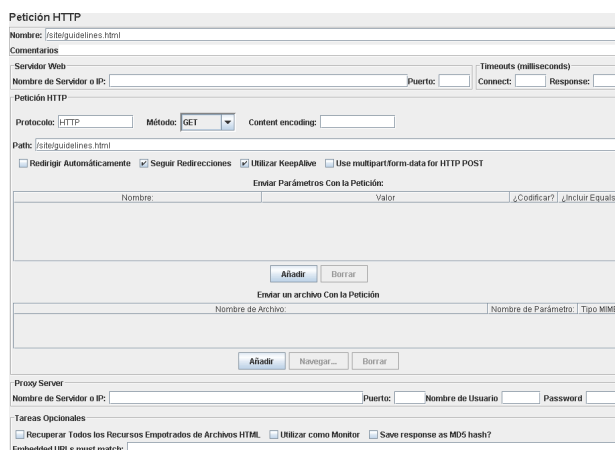


Figure 4. Properties for a HTTP Request.

Another important element is the *HTTP Cookie Manager*, which allows each thread (user) have access to his cookies shared among all his/her requests. Next element in our test plans is a *HTTP Request*, which represents one request that is made by a user, against a predefined web server, and aiming at a concrete path. A test plan is made of a set of requests that test the functionalities of a web application. This HTTP request will take the default values specified before, although they can be overwritten if necessary. Figure 4 shows the properties of a HTTP request.

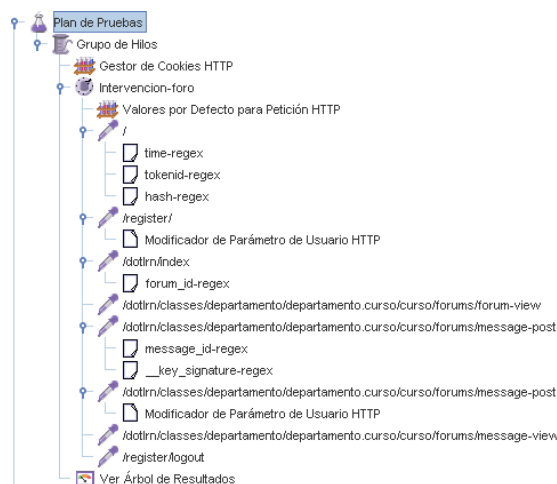


Figure 5. Test plan to write a message in a forum in .LRN.

An interesting element is the *HTTP User Parameter Modifier*, which keeps the route to an external XML file used to store the value of variables. This file keeps multiple couples *variable-value* that will be used by the threads. The last element in our test plans are *Listeners*. These are entrusted with the storage of the results of the HTTP requests and their presentation in different formats, such as graphs.

Figure 5 shows a test plan (*Plan de pruebas*) used to simulate the writing of a message in a forum in .LRN, which has one HTTP default value, several HTTP requests and regular expressions, a cookie manager, and a listener, this last one is a results graph (*Árbol de resultados*). For this test plan, we have to visit a number of paths during its executions, such as `/`, `/register/`, or `/dotlrn/index`, each with several regular expressions extractors and *HTTP User Parameter Modifiers* that will guide the test plan process.

Each LMS works in a totally different way for all the scenarios (login/logout, write a message in a forum, download a file...), so we could not reuse test plans between LMSSs, which increased the effort needed to perform the experiments of this work.

Finally, sample courses had to be created in each platform in order to run the test plans for each LMS.

IV. PERFORMANCE EVALUATION

The overall evaluation of the three LMSSs was twofold. First, we have conducted an evaluation comparing the use of the hardware resources, in which case studies varying the number of users have been executed. Second, we have evaluated each LMS from the administrator point of view. These evaluations are presented in this section.

A. Use of hardware resources

For the first evaluation, experiments have been conducted for 90, 100, and 110 concurrent users. We chose

TABLE I.
OF QUERIES PERFORMED.

# of users	Sakai	Moodle	.LRN
90	1890	2610	2160
100	2100	2900	2400
110	2310	3190	2640

TABLE II.
AVG EXECUTION TIMES OF TEST PLANS

# of users	Sakai	Moodle	.LRN
90	0:01:04	0:01:50	0:02:19
100	0:01:13	0:01:50	0:02:42
110	0:01:26	Collapse	0:04:22

TABLE III.
AVG QUERY COMPLETION TIME (MSEC)

# of users	Sakai	Moodle	.LRN
90	2834.333	3403.125	5449.625
100	3270.556	3652.4	6518.667
110	3646.556	Collapse	7465.714

these amounts of users because they are enough to create congestion in the machine running the experiments and draw conclusions – less users would not create congestion, and more users would overload the server.

Besides, each experiment was repeated ten times and average results were calculated in order to ensure that conclusions were not based on spurious events. Measurements were collected on CPU utilization, memory usage, network input/output (IO), and disks IO. The LMSs were installed on a server with the following features:

- Processor: AMD Phenom(tm) II X6 1035T 3.10 GHz (6 cores).
- Memory: 3 Gb DDR3.
- Hard disk: SATA 7200 rpm.
- Network interface card: 10/100/1000 Mbps.
- Operating system: Debian Squeeze 64 bits, Linux kernel 2.6.32.

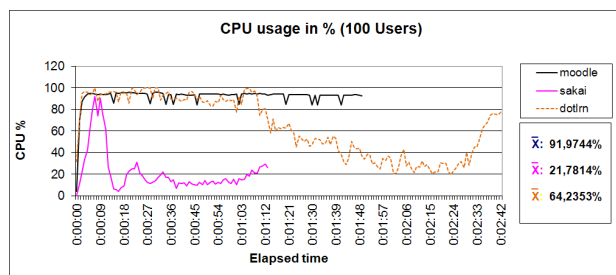
The first statistics presented here are related to the queries required by each test plan, where a query refers to a call to a Uniform Resource Locator (URL). Table I presents the number of queries for each case study. It can be seen that the LMS that requires more queries to complete test plans is Moodle, followed by .LRN and Sakai, respectively. This data depends on the way how each LMS is implemented, and will influence other results, as will be seen later.

Other interesting statistic is the test completion time, presented in Table II. Recall that these data represent the result of several repetitions of each experiment, where the average of all the experiments is calculated. It can be seen that Sakai is the fastest tool, taking less than half the time than .LRN (for

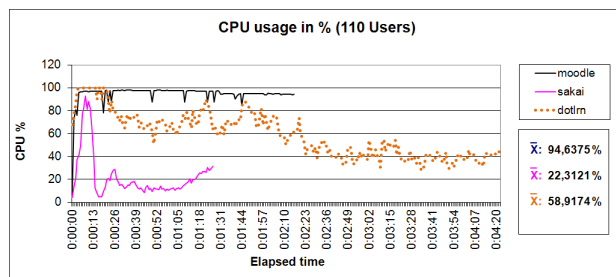
90 users) and around one third of the time (for 110 users). This data is related to the previous table, since the less queries a test plan requires, the faster the test plan is executed. With regard to Moodle, none of the tests with 110 users were completed, since the server becomes overloaded. This indicates the high requirements Moodle poses on the hardware it runs compared to the other LMSs (all the experiments with .LRN and Sakai complete successfully).

Figure 6 presents the CPU consumption of experiments for 100 and 110 concurrent users. Data on the experiments for 90 concurrent users are not shown since the server does not become overloaded, so no interesting results can be obtained from those experiments. In this figure, it can be seen that Moodle has the highest CPU requirements since it is over 95 % most of the experiments time, and yields an average of over 90 % for all the experiments. As opposed to it, both Sakai and .LRN have clearly lower CPU usage. This is especially true in the case of Sakai, which yields an average CPU consumption of around 22 % for all the experiments. Considering the results presented in this figure, we can conclude that, in the case that the server we plan to run our LMS has CPU limitations, Moodle should not be used – Sakai is the best tool from this point of view.

Next statistic we present in Figure 7 is the memory usage. As before, only results for 100 and 110 concurrent users are depicted. It can be seen that the LMS with the highest memory requirements is Sakai, followed by Moodle and .LRN, but differences are less significant than for the CPU usage. An interesting point is that the memory usage of Sakai remains with little variations as we increase the number of concurrent users, as opposed to Moodle and .LRN. This is, Sakai is less dependent on

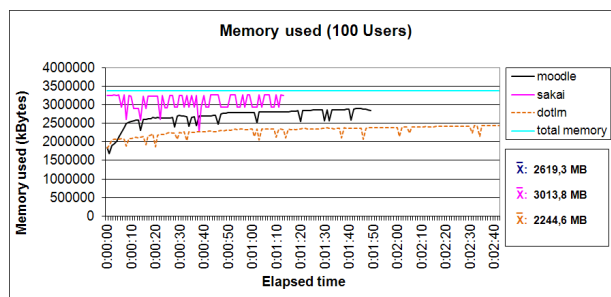


(a) 100 users

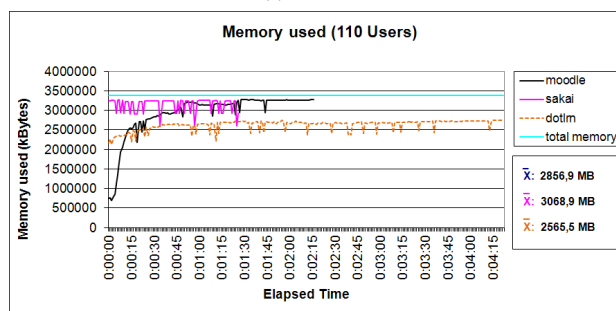


(b) 110 users

Figure 6. % CPU utilization.



(a) 100 users



(b) 110 users

Figure 7. Memory usage.

memory availability than the other LMSs with regard to the number of concurrent users. Moodle increases its memory usage from 2.55 GB to 2.79 GB (around 8.95% increase) when increasing the number of users. In the case of .LRN, the increase is from 2.19 GB to 2.56 GB (around 14.41% increase). As opposed to them, Sakai requires 2.94 GB when the number of concurrent users is 100, which rises to 2.99 GB for 110 users (merely 1.94% increase). Such memory behavior also exists when raising from 90 to 100 users, and these results conclude that Sakai is less dependent on the actual number of users connected to the system.

Figure 8 presents the swap space used by each LMS. This statistic only refers to the 110 concurrent users experiments since no swapping took place in the other experiments. It can be seen that only Moodle presents

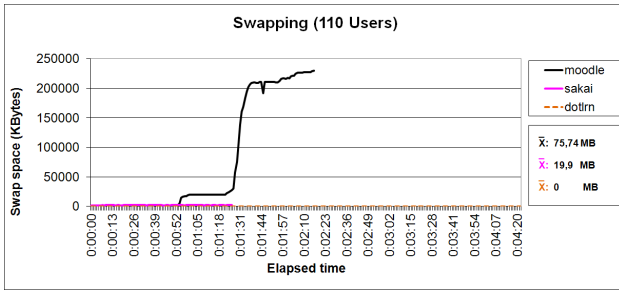
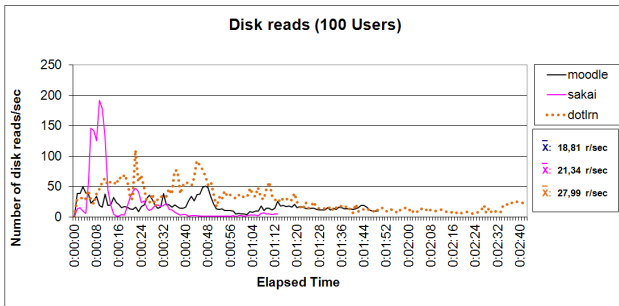
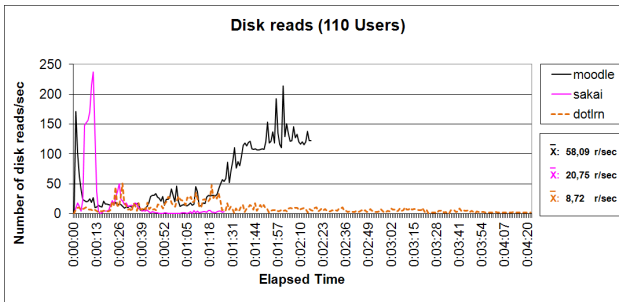


Figure 8. Swapping.



(a) 100 users



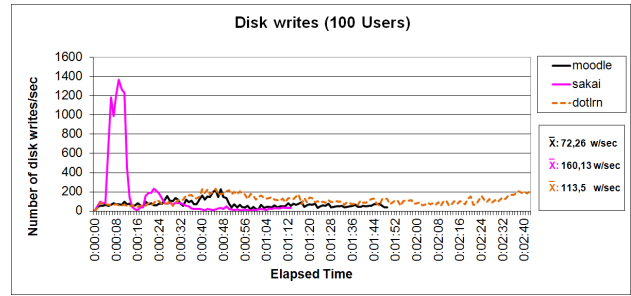
(b) 110 users

Figure 9. Disk reads.

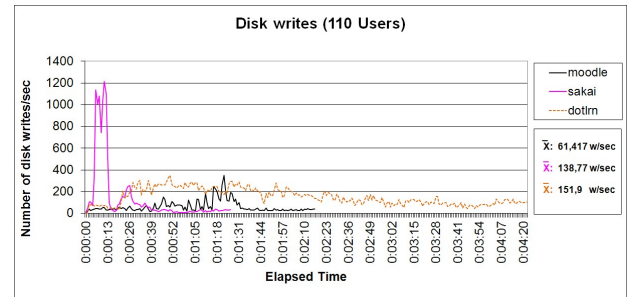
significant swap memory usage. This is related to the problems Moodle suffers under the experiments with 110 users (experiments never finish).

Next statistic to be presented is the number of disks reads and writes per second, and this is shown in Figures 9 and 10. It can be seen that Sakai presents a peak at the beginning of experiments and then disk reads decrease to become almost constant as experiments progress. On the other hand, .LRN presents several peaks at the beginning of experiments, but their size is smaller than those of Sakai. In turn, Moodle presents a peak for the 110 users experiments, which is related to the swap memory usage presented previously. This means that in this point Moodle starts paging until experiments finally collapse. Regarding disk writes, at the beginning of experiments Sakai presents a peak similar to the one found in the disk reads. The other LMSS present more constant values, with less significant peaks.

Now, we present statistics regarding the network IO. Figures 11 and 12 show the received and transferred bytes for each LMS, respectively. In both cases, Moodle is the LMS that makes the most use of the network, both for input and output. These statistics are related to the number of queries performed by each LMS (presented in Table I), so the more queries a LMS requires, the more network I/O is necessary for that LMS. The results

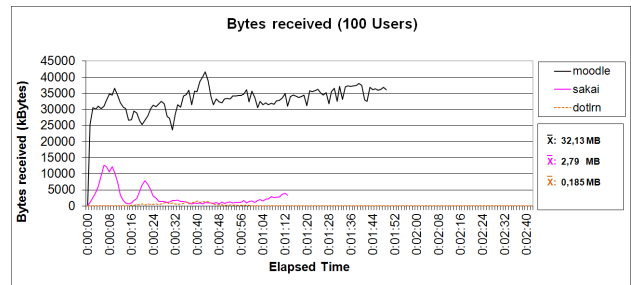


(a) 100 users

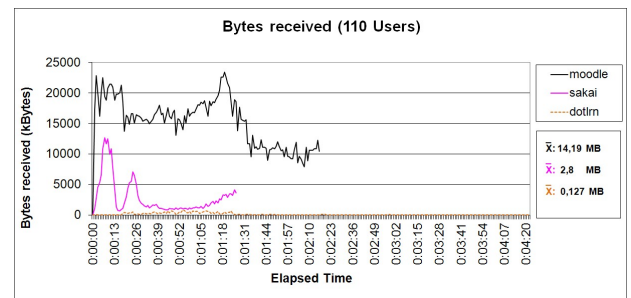


(b) 110 users

Figure 10. Disk writes.



(a) 100 users



(b) 110 users

Figure 11. Network input.

for Sakai and .LRN are similar for 100 and 110 concurrent users, but Moodle presents lower network I/O for 100 users than for 110. This suggests that Moodle is overloaded with 110 concurrent users, and cannot process all the users' queries, which means that many queries remain unanswered – resulting in lower network I/O.

The last statistic we present in this work is the number of queries dropped, and is depicted in Figure 13. This statistic refers to the TCP packets with the SYN code, which refers to the first packet of the connection start-up process. These SYN packets are stored in a kernel buffer, and when it gets filled, packets are dropped – and the connections they represent are not created. The status of the SYN queues can be considered as an indicator of the congestion of a server [16]. In this figure, the number of

dropped SYN packets is negligible for all the LMSs for 100 users, only .LRN presents some packet drops. When the number of concurrent users rises to 110, all the LMSs except .LRN present the same values as for the previous case. But .LRN shows an increasing tendency, which means that .LRN cannot empty the buffers as fast as required. For this, connections are not established as fast as they should, and make the test plan take longer time to complete (test plan durations are presented in Table II).

To summarize this section, we can conclude that:

- Regarding CPU power, Moodle requires more CPU than the other LMSs, so it should not be used in the case of CPU limitations on the machine we plan to run our LMS— in this case, Sakai is the best option since its CPU requirements are the lowest for the test plans executed.
- Regarding memory, Sakai consumes more memory, but its memory consumption is less dependent on the number of concurrent users since it remains almost constant for the test plans we executed. On the other hand, Moodle levels of use increase heavily with the number of users, and .LRN also suffers an increase, but lower than Moodle.
- Regarding disks, the reads and writes of each LMS suffer peaks, but their performance is reasonably good for the test plans executed. Moodle makes more use of disks, since the system starts paging before experiments for 110 users collapse, but this does not show problems of disks.
- Regarding network IO, Moodle requires more transfers than the other LMSs, and this may happen because the test plans for Moodle require more requests than those for the other LMSs. Apart from it, Sakai requires more data transfers but during shorter time than .LRN.

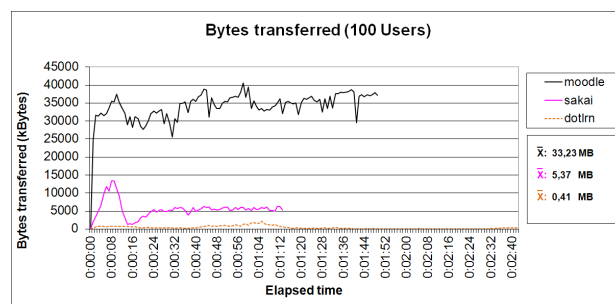
Based on these experiments, we created a rating presented in Table IV which summarizes this section, where the studied features of LMS have been rated from 1 (worst) to 5 (best). We can see that Sakai and .LRN present similar ratings, whilst Moodle performs clearly worse than them.

B. Administrators' survey

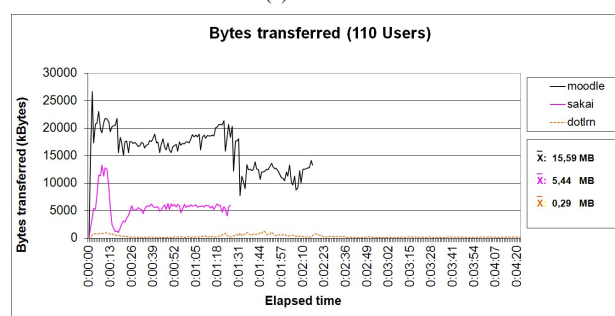
The second evaluation we performed is based on the personal opinion of the system administrator of the LMSs under study, and compares the ease of use in order to set up the LMSs for the experiments required in this work. This second evaluation is also important since it evaluates different parameters which are key for the administrators responsible of using the system. Since this paper strives at helping them how to choose the most suitable LMS, the opinion of a system administrator must be considered in this paper.

For this evaluation, a number of features of LMS have been rated from 1 (worst) to 5 (best), by the three system administrators who performed the installations and tests presented in this paper, where all of them dealt with the three LMSs. The set of features have been defined by the system administrators, so they are the features which they consider the most significant for this work. These ratings are presented in Tables V, VI and VII, where it can be seen that .LRN is one step behind Sakai and Moodle with regard to the system

admin opinion, mainly because it has a less active community and less tools available. Besides, .LRN is more difficult to install and configure – even if the default configuration options are chosen. Moodle and Sakai present similar results, and Moodle has slightly better score in the availability of modules, maintenance and community items, which shows that Moodle is the best supported LMSs (with little difference over Sakai). Table VIII presents a summary of these results, in which the rate that obtained the highest number of responses was chosen for each feature of each LMS. This highlights the differences between Sakai and Moodle, and .LRN.

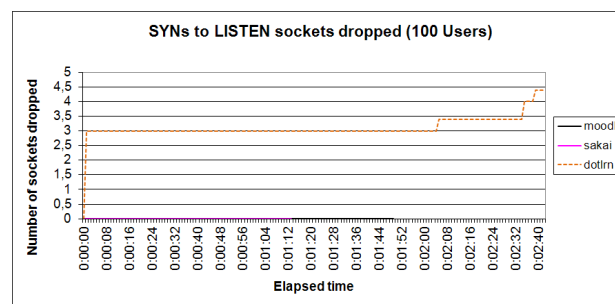


(a) 100 users

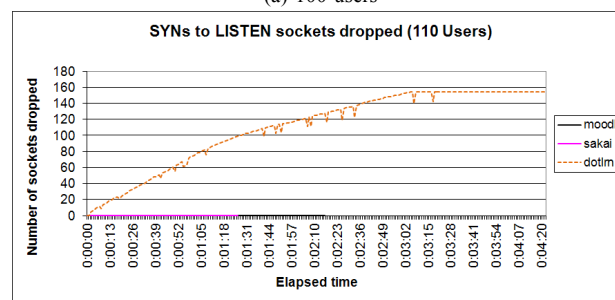


(b) 110 users

Figure 12. Network output.



(a) 100 users



(b) 110 users

Figure 13. Network queues.

TABLE IV. SYSTEM RATING

Feature	Sakai	.LRN	Moodle
CPU power	5	4	1
Memory usage	3	3	2
Disks IO	4	4	4
Network IO	5	5	2
Total	17	16	9

Considering both system and admin ratings, the overall ratings are presented in Table IX. It can be seen that Sakai is the leader in this ranking since it combines relatively good system performance and ease of use. As opposed to it, .LRN and Moodle have problems in either rating, which decrease their overall rating.

V. CONCLUSIONS

The main aim of this paper is providing hints in order to choose the most appropriate LMS for the necessities of a learning institution. For this, a twofold performance evaluation has been conducted. First, an extensive set of experiments were performed in order to find the use each LMS makes of the hardware resources of the machine where it is running. Second, an evaluation considering the opinion of the system administrators is also presented.

As a result, Sakai and .LRN are the two best tools from the performance evaluation point of view, since they make efficient use of the resources (CPU, memory, network...) of the machine running the experiments. From the system administrator point of view, Moodle and Sakai have similar ratings, mainly because they are sup-

ported by large and active communities, as opposed to .LRN.

The main conclusions of this work are the following guidelines. First, if you expect large number of users, the best option should be either Sakai or .LRN. Second, if you want ease of use, and the support of a large community of users, the best option would be Moodle or Sakai.

In summary, Sakai is considered the best tool because it obtains high ratings in both evaluations. This is mainly because it has a large community of users, it is easy to install and use, and it is kept up-to-date.

For future work we plan to study customized setups for each LMS – rather than their default configurations. Besides, we plan to develop metrics that define the performance of the LMS from the point of view of their users (students and faculty), in order to minimize the resources needed for the LMS to provide quality of service to its users.

ACKNOWLEDGMENT

Authors would like to acknowledge the support of the following European Union projects: RIPLECS (517836-LLP-1-2011-1-ES-ERASMUS-ESMO), PAC (517742-LLP-1-2011-1-BG-ERASMUS-ECUE), EMTM (2011-1-PL1-LEO05-19883), MUREE (530332-TEMPUS-1-2012-1-JO-TEMPUS-JPCR), and Go-Lab (FP7-ICT-2011-8/317601). Furthermore, we also thank Spanish Ministry of Science and Innovation for the Project TIN2008-06083-C03/TSI, and the Community of Madrid for the support of E-Madrid Network of Excellence (S2009/TIC-1650).

TABLE V. SYSTEM ADMINISTRATOR RATINGS: SAKAI

Feature	1	2	3	4	5
Up-to-date documentation				66%	33%
Easy installation and administration				66%	33%
Ease of use of GUI				66%	33%
Availability of tools			33%	66%	
Maintenance and updates				66%	33%
Active users community				100%	

TABLE VI. SYSTEM ADMINISTRATOR RATINGS: .LRN

Feature	1	2	3	4	5
Up-to-date documentation		33%	66%		
Easy installation and administration			100%		
Ease of use of GUI		33%	66%		
Availability of tools			100%		
Maintenance and updates			33%	66%	
Active users community		33%	66%		

TABLE VII. SYSTEM ADMINISTRATOR RATINGS: MOODLE

Feature	1	2	3	4	5
Up-to-date documentation				100%	
Easy installation and administration				100%	
Ease of use of GUI		66%	33%		
Availability of tools					100%
Maintenance and updates				33%	66%
Active users community					100%

TABLE VIII. SYSTEM ADMINISTRATOR SUMMARY RATINGS

Feature	Sakai	.LRN	Moodle
Up-to-date documentation	4	3	4
Easy installation and administration	4	3	4
Ease of use of GUI	4	3	2
Availability of tools	4	3	5
Maintenance and updates	4	4	5
Active users community	4	3	5
Total	24	19	25

TABLE IX. FINAL RATINGS

Feature	Sakai	.LRN	Moodle
Total System Ratings	17	16	9
Total Admin Rating	24	19	25
Total	41	35	34

REFERENCES

- [1] R. Pastor, R. Hernández, S. Ros, A. Robles-Gómez, A. C. Caminero, M. Castro, and R. Hernández, "A video-message evaluation tool integrated in the uned e-learning platform," in *Proc. of the Frontiers In Education Conference (FIE)*, Rapid City, USA, 2011.
- [2] S. Ros, R. Hernández, T. Read, R. Pastor, M. Castro, M. Rodríguez- Artacho, and A. Robles-Gómez, "The UNED's interoperable virtual campus service management architecture," in *Proc. of the Frontiers in Education Conference (FIE)*, Washington D.C., USA, 2010.

SPECIAL FOCUS PAPER
COMPARISON OF LMSS: WHICH IS THE MOST SUITABLE LMS FOR MY NEEDS?

- [3] Moodle, Web page at <http://moodle.org/>, Date of last access: May 13, 2013.
- [4] Sakai, Web page at <http://www.sakaiproject.org/>, Date of last access: May 13, 2013.
- [5] .LRN, Web page at <http://www.dotlrn.org/>, Date of last access: May 13, 2013.
- [6] Blackboard, Web page at <http://www.blackboard.com/>, Date of last access: May 13, 2013.
- [7] AOL server, Web page at <http://www.aolserver.com/>, Date of last access: May 13, 2013.
- [8] Apache Tomcat, Web page at <http://tomcat.apache.org>, Date of last access: May 13, 2013.
- [9] Apache HTTP Server, Web page at <http://httpd.apache.org/>, Date of last access: May 13, 2013.
- [10] Apache JMeter, Web page at <http://jmeter.apache.org/>, Date of last access: May 13, 2013.
- [11] <http://httpd.apache.org/docs/2.2/programs/ab.html>, Date of last access: May 13, 2013.
- [12] Badboy, Web page at <http://www.badboy.com.au/>, Date of last access: May 13, 2013.
- [13] HP LoadRunner, Web page at <http://www8.hp.com/us/en/software/solutions/software.html?compURI=1175451#UFyRTFYRhJU>, Date of last access: May 13, 2013.
- [14] IBM Rational Performance Tester, Web page at <http://www-01.ibm.com/software/awdtools/tester/performance/>, Date of last access: May 13, 2013.
- [15] JMeter Plugins, Web page at <http://code.google.com/p/jmeter-plugins/>, Date of last access: May 13, 2013.
- [16] C.-H. Tsai, K. G. Shin, J. Reumann, and S. Singhal, "Online web cluster capacity estimation and its application to energy conservation," *Transactions Parallel Distributed Systems*, vol. 18, no. 7, pp. 932–945, 2007. <http://dx.doi.org/10.1109/TPDS.2007.1028>

AUTHORS

C. Caminero, R. Hernández, S. Ros, Ll. Tobarra, A. Robles-Gómez, and R. Pastor are with the Dep. de Sistemas de Comunicación y Control Universidad Nacional de Educación a Distancia, UNED Madrid, Spain. {accaminero, roberto, sros, llanos, arobles, rpastor}@scc.uned.es

This article is an extended and modified version of a paper presented at the EDUCON2013 conference held at Technische Universität Berlin, Berlin, Germany from March 13-15, 2013. Submitted 13 May 2013. Published as re-submitted by the authors 25 July 2013.