Evaluating LMS/CMS performance

Marjan Krašna
Faculty of Arts
University of Maribor
Koroška cesta 160, 2000 Maribor. Slovenia
marjan.krasna@uni-mb.si

Branko Kaučič
Faculty of Education
University of Ljubljana
Kardeljeva ploščad 16, 1000 Ljubljana, Slovenia
branko.kaucic@pef.uni-lj.si

***Abstract**. During last few years we have started using LMS/CMS (Learning Management Systems/Course Management Systems) extensively. Students and teachers use them regularly in two way communication. Blended learning has become dominant form of education for ICT aware teachers. If we want to design the successful LMS/CMS we need to know some hard facts. We did not rely on the information from the producer of the LMS/CMS therefore we have performed our own tests. Results show that we should not trust their claims blindly. Test also proved that our hardware is completely capable to support TLS processes for our faculty.*

Keywords: e-learning, LMS, CMS, simulation, test

1 Introduction

As more and more students starts using LSM/CMS on daily basis we inevitably come to LMS's limitations. Questions that arise may be simple but answers are not. One question is particularly important - how to design the LMS? In the research of this topic we have found out that this question raises another set of questions [1]. It also address the problem of hardware, software, response time, number of students, number of concurrent users and number of concurrent requests.

Let suppose that LMS will be developed from the scratch. First thing to consider is the amount of users during the life time of the LMS. With the life time we denote the time between hardware acquisition and the span of the extended warranty of the same hardware. Typically this would be between three or four years. During this period we need to predict the number of students that will use our system and amount of concurrent users and concurrent requests. With these preconditions in mind we check for the desired LMS hardware requirements and we are done. This simple solution can be found on the web in matter of minutes. But there is a big grey area in this procedure. [2, 3] Can we know how many concurrent requests the specified hardware is capable to handle? Searching for this answer is a bit trickier and usually gives us only vague assumptions. It is interesting that we could get only the amount of memory (RAM) needed per concurrent user of the LMS. Little math calculations shows us horrified amount of memory we need to cover students of our faculty. Therefore we wanted to make our own analysis and verify their claims.

2 Preparing for the test

Before we could start with the test we need to find some information about the human feeling of response time [4]. What is fast, medium and slow response time? According to these data we could evaluate our tests and offer a positive user experience with LMS [5, 6]. Another question is how to measure the response time? If we take into account that we are in the internet environment we could define the response time as the time between the request of user and the final display of the page. But this time is not the same as the time measured at the server side. Which time is more accurate for measure is simple. But for user satisfaction the first one is more important.

The structure that influence the response time are:

- Size of the web page
- Bandwidth
- Time of full cycle
- Number of cycles
- Amount of time that server and client process the data.

Web page size was important in the dial-up networking environment. Today we have high speed internet connection and the time for transmit the web page drops dramatically. Most of the problem with the page size is today addressed to invisible code; large pictures; and downloaded components (e.g. non streamed flash video).

Bandwidth of the internet connection gives you a rough insight to the prediction of time the web page will be delivered to the client computer the client computer. According to the equation (1) we can calculate the time to transmit the web page.

$$t_{raw} = \frac{S}{v} \quad (1)$$

Where S is the size of the web page (in kilobytes) and v is the bandwidth of the internet connection (kilobytes per second).

Time to transmit the web page dropped dramatically with the high speed internet connections but it is still important constraining factor in the slow speed wireless networks (e.q. GSM transmission).

Time of full cycle is the time from the first request to the first response. This time can be measured with the PING command available in most operating systems. Time depends on the number of hops and dynamic events on the network. Typically time is measured in milliseconds but can grow up to second scope. The effect of this time has impact to every request/response event in the transmission of data.

Number of cycles depends on the objects (e.g. images, applets …) on the web page being transmitted from server to client. More components on the web page is present more cycles are needed for whole web page to transmit to the client. Modern web browsers enables up to four parallel transmissions of elements. Therefore the number of required cycles for transmission can be calculated with the equation (2):

$$n_{cycles} = 4 + \frac{3 * n_{objects}}{4} \quad (2)$$

Processing time depends on two factors. Speed of disk transfer and processing time on the processor. Disk transfer time occurs in all circumstances but processing time can be quite significant if we have dynamic web pages. The amount of memory on the server can dramatically reduce time of static web pages transmission in rapid succession since only first time disk transfer occurs. Every additional request is served directly from the memory. Since disk speed is very slow compared to the memory, the response of the system is dramatically shortened. But the memory of the system is finite, and the web page data needs to be replaced in the memory when different web pages are requested. In the case of the dynamic web pages the raw computer processor power is needed to prepare the data and transmit them to the client. The response is therefore proportional to the workload of the server.

2.1 Concurrent access

We have a problem defining concurrent access. In one aspect this can be the number of concurrent users using the server. On the other aspect it can be the number of concurrent requests. In the first time we have no problem with the definition but in the late we have. What are the concurrent requests? Are these the requests that happen in the same time or in the same second? The problem is that we cannot measure the requests in the same time; we can only measure the requests that occur in the same Δt.

Concurrent access affects the response time. Users experience and satisfaction with the response time is therefore subjective measurement and according to some research can last up to few seconds [5]. For the properly designed server it is crucial to know how many concurrent users will have desired response time.

2.2 Response time and satisfaction

The satisfaction of the users experience with the web response is not just a matter of positive users' experience. It also provides financial benefits to the companies who sell their products on the web [7]. No matter of the purpose of web pages we can always receive the same response from the users.

Two questions are in order when we are talking about the response time: How fast is fast enough and how slow is too slow?

From the year 1968 the IBM have made research and found out that users need short response time to stay focused and productive [8]. Their research show that response time should be under the second and that is still the requirement in the software development. On the other hand they found out that response time that is more than ten seconds is unacceptable for the users.

In the research about the computer hardware buyers the negative effect on the buyer by the response time which lasts more than ten seconds was also confirmed [9]. But they found out that the response time should be faster at the end of the transaction (buying) than at the beginning. Their limit for the response time was shortened to eight seconds [10]. In the response time that is longer than eight seconds a significant number of buyers did not complete the transaction and consequently bring the financial loss to the companies.

Subjective effect of the response time is somehow different than the measured one. When users were asked to assess the fast and slow response they decide differently. To them one of the fastest web page was slow and one of really slow seems to be fast [11]. This impression was correlated to the number of elements valuable to the users on the web page. Results shows that users are prepared to wait around four seconds for each web element. Fastest web page has little elements and seems to be slow.

Time of tolerance is the time span between the first impression of the user that web response is slow and the time the user is willing to wait for the response. This time is around four times longer then the acceptable response time (around 40 seconds). After the time of tolerance is over users definitely leave the web page.

3 Testing MOODLE

We have picked the LMS/CMS Moodle for our distance learning server at Faculty of Arts University of Maribor, and for e-learning server at Faculty of Education University of Ljubljana. In the hardware requirements for the Moodle we have found some recommendations but none of them were based on factual testing. Administrators exchange their knowledge on web forums (Moodle forum) and some claim that server needs faster processor other claim that server needs more RAM. Since we cannot accept guesses and we need to provide the satisfactory response to all our users we decide to conduct our own test.

For testing Moodle we have decided to check the following features:

- How many users can be served by personal computer as server?
- What is the difference between different operation systems?
- When the response time becomes unacceptable?
- Which operation is the most complex and is capable for the optimization?
- How capable is our faculty's LSM/CMS system?

For the testing purpose we have developed the software that simulates the user's interaction with the LMS system. In the test we use computers in the classroom, local area network (100 Mbps) and two servers. One of our servers was P4 class desktop computer running two different kinds of Moodle. (P4, 1,5GHz, 1GB Ram, Windows 2003 server, IIS, MySQL and PHP / Linux, Apache, MySQL and PHP) and other was our faculty server (P4D, 3,2GHz, 2 GB RAM).

The reason we use desktop class computer was simple. We wanted to be sure to reach the limits of its capacity in small amount of concurrent users. The test also omits the processing on the client side. It was not our intension to include the client side processing even though we know it is also a vital part in the response of the system as a whole. All the data that was transmitted from server to the clients were discarded immediately on client side. To minimize the effect of the latency of the network we use all computers (client and server) on the same router (Linksys-Cisco type)

4 Results

In the first test we use 120 simulated users with the random time span between 10 to 15 seconds between the users action. The server (desktop class computer) usage in idle time was 0% CPU and 308 MB of RAM on Windows server 2003 platform. During the test we monitor the processor activity, RAM used and network activity. All these monitoring activities are built in the Windows systems. During the test the CPU was used at 100% all the time and response time was over the eight (8) seconds limits. Figure 1 shows the response times for this server.

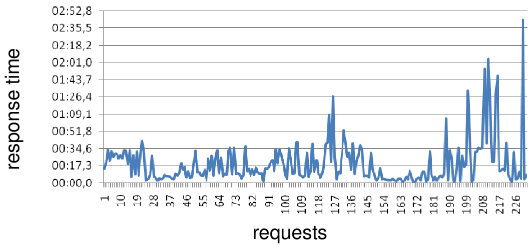

Figure 1: Response time - Windows

When we repeated the test on the same system with the Debian Linux we observed that in the idle state desktop class server used 0% CPU and 115 MB of RAM. During the test we have surprisingly found out that response time was even worse than on the Windows system (Figure 2).

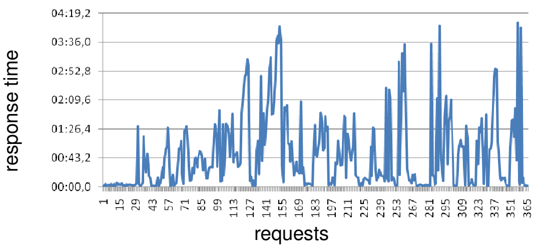

Figure 2: Response time - Linux

Each configuration was tested for a period of one week and the results did not significantly deviate between the tests. We need to remind that systems were not optimized in any way and they were used as installed from the installation media.

Testing faculty's Moodle server was done a bit different. Server is available to our students all the time. For the purposes of testing we did not disable other users. But we perform tests during the night time when the server was statistically idle. Installed software on our server was:

Apache 1.3.33
MySQL 4.1.21
PHP 4.3.11 and
Moodle 1.6.1+

All the data are on SATA disks in mirror configuration. During one year of operation we still have not found any performance degradation despite the fact that we have almost 4000 registered users.

At the beginning of the test when the server was idle it used 0% CPU and 500 MB of RAM. All the requests were served almost immediately. For the test purpose we create 600 concurrent users that create events in the random time span between 1 and 2 seconds. The test configuration that was used in desktop server configuration posed too little workload for this server and all the requests were served almost immediately (below the valid measurement time span). Statistically each response was served in under 0.3 second and therefore almost no concurrent events occur.

During the test we needed to monitor some additional parameters - concurrent requests. We have decided to measure the concurrent requests as requests within 1 second time span. For the user activity we use the predefined jobs. Each simulate user began his activity with the entrance to the course, and then randomly selected following activities: transfer of small size file (web page access), transfer of medium size file (SCORM package access), transfer of large file size (simulated video), flash video, review the forum; and at the end logout from the system.

As we predicted, the response time grew with the number of users. Most of the time multiple requests were served and during the test 17.000 requests were simulated. But even with 600 concurrent users we have found out that there were not more than 20 concurrent requests within one second (Figure 3). But despite the fact that server was fully loaded we could get normal user response from it.

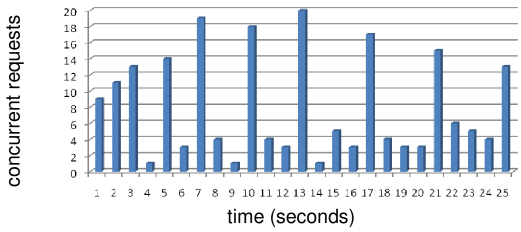

Figure 3: Concurrent requests on 600 concurrent users

5 Conclusions

For any large system it is necessary to make a system analysis. Wrong design of the system will influence the user's satisfaction and inherently provides a loss of money and/or social negative impact to the company's image. For a successful design process we need to know systems constraints. How many users will use the system and how many of them will use it at the same time.

We have found out that the recommendations about the Moodle on the web are not accurate. If we obey them we would need far more expensive server than we actually need. The amount of RAM proved not to be a real factor since computer never used more than 20% more of RAM as in the idle state. Most important factor is however the processor’s processing power. We have also found out that our faculty server is well designed for its operational life. In the future we will need to do additional tests since more and more e-learning material is available and users change their habits in using web learning material.

References

[1] Yueh HP (Yueh, Hsiu-Ping), Hsu S (Hsu, Shihkuan), Designing a learning management system to support instruction, COMMUNICATIONS OF THE ACM, 2008

[2] Gruser JR, Raschid L, Zadorozhny V, Zhan T, Learning response time for WebSources using query feedback and application in query optimization, VLDB JOURNAL, 2000

[3] Su LT, A comprehensive and systematic model of user evaluation of Web search engines: II. An evaluation by undergraduates, JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 2003

[4] Yan T (Yan, Ting), Tourangeau R (Tourangeau, Roger) Fast times and easy questions: The effects

of age, experience and question complexity on web survey response times APPLIED COGNITIVE PSYCHOLOGY, 2008

[5] Galletta DE, Henry RM, McCoy S, Polak P, When the wait isn't so bad: The interacting effects of website delay, familiarity, and breadth, INFORMATION SYSTEMS RESEARCH, 2006

[6] Savoia, A. Better Software magazine. Web Page Response Time 101: http://www.stickyminds.com/sitewide.asp?ObjectId=5030&Function=edetail , 2001

[7] Rose GM, Meuter ML, Curran JM, On-line waiting: The role of download time and other important predictors on attitude toward e-retailers, PSYCHOLOGY & MARKETING, 2005

[8] Ben Shneiderman, Response time and display rate in human performance with computers, ACM Computing Surveys, Colume 16, Issue 3, 1984

[9] Nina, B., Anna, B., & Allan, K. Integrating User-Perceived Quality into Web Server Design. 9th International World Wide Web Conference. 2000

[10] Zona research. Keynote Systems. Resource Library: White Papers: http://www.keynote.com/docs/whitepapers/zona_need_for_speed.pdf, 2001

[11] Sevcik, P. J. NetForecast Articles. Understanding How Users View Application Performance: http://netforecast.com/Articles/BCR%20C22%20Performance%20Zones.pdf, 2002