Balancing Dynamic Overload in Moodle
E-Learning Servers by Virtual Means

Eduard Mihailescu

(1) Faculty of Electronics, Telecommunications and Information Technology, Technical University of Iasi, 11, Carol I Str., 700506, Iasi, ROMANIA
E-mail: meduard@etti.tuiasi.ro

Abstract
This paper summarizes a PhD research project that has contributed towards the use of virtual means for balancing hardware and software overloads of elearning servers (i.e., Moodle), when coping with extended computational tasks in science simulation environments. We provide experimental surveys and design methodologies. The theory we support is endorsed by an elearning project, which employs Moodle LMS and in-house tailored modules, for networking and biomedical engineering at the Department of Computer Networks and Distributed Systems at "Gh. Asachi" Technical University of Iasi. This project has been in progress for since 2008 and is due to be completed in the first half of 2011.

Keywords: LMS (Learning Management Systems), Moodle, Server Management, Virtual Means.

Introduction
The past twenty years have seen a marked increase in research around elearning and web-based teaching at various levels. As described in literature, LMSs (Learning Management Systems) provide support for the mainstream infrastructure of computer aided education. Recent developments in broadband internet and the employment of FOS (Free and Open Source) learning tools gain momentum and a comprehensive approach, binding teaching, learning, assessing and student management seems to emerge. Yet, technical surveys and feedback from server management staff emphasize drawbacks and restraints in the usage of elearning, mainly due to data surges and circumstantial overload of the software and hardware networks, especially when coping with simulation environments.

Theoretical perspectives
Learning Management Systems (LMS) are widely encountered in the daily activities of education by computer means. Most of them, basically, attempt to integrate collaborative teaching, class attendance, student project management and individual learning data from a school or university into a single computer system that can serve all the particular needs of the ones involved. Students, professors and staff on different organizational levels process the same information (hierarchical clearance on a need-to-know basis is required) and can update it. When one educational department finishes with the information, it is automatically routed via the LMS system to the next department in the university’s chain of activities. LMSs connect with different other programs from third-parties (virtual libraries, Ministry of Education’s network, partner universities worldwide etc) and achieve integration of data and educational activity. Dating from midst of the 90’s “(Dougiamas and Taylor, 2003)”, LMSs have evolved out in the educational software
environment and are derived from content management tools (CMS, Content Management Systems). Modern systems have reached to cover almost all aspects of the teaching procedures: curricula management, class attendance management, student’s homework and projects, embedded communication means like instant messaging, email and video conferences, various testing and quizzing tools and others. These learning platforms typically handle the teaching, learning, assessing and student accounting for an educational institution or even government organizations. Sometimes, LMSs are referred as Virtual Learning Environments (VLE). For integration and standardization purposes, some basic features that a LMS should comply to are briefly accounted hereafter: a) administration: the learning software must allow administrators to manage user registrations and profiles, define roles, set curricula, chart certification paths, assign tutors, author courses, manage content; b) coherent adherence to learning standards: the LMS has to comply to major e-learning standards like SCORM “(SCORM)” and the ones issued by IEEE LTSC (Learning Technology Standards Committee)”(IEEE)”; c) modular architecture support: although this is not compulsory for a LMS platform, it is desirable to have the possibility to natively integrate modules like: an evaluation engine that enables authoring within the product and includes assessments as part of each course, class management, embedded communications and others.

A rapidly growing force in the software world is that of Open Source Software (OSS), where the propriety and usage of the computer program is covered by an Open Source license such as the popular GNU Public License (GPL). Unlike typical commercial software, OSS licenses explicitly allow anybody to freely use, modify, redistribute and even sell the software under the condition that the open source license is maintained. In general this means that user modifications are absorbed into the main software project, and so the software evolves to embody the values of user community, even as that community itself evolves. This type of system has already proven very successful in developing much of the basic software that makes the Internet possible (Linux, Apache, Bind and Sendmail are among the most well-known examples of the thousands that exist)”(Dougiamas and Taylor, 2003)”.

As emphasized in a previous paper “(Mihailescu, 2009)”, Moodle in on the edge of the open source wave and leads among non-proprietary learning management systems. Is the acronym for Modular Object-Oriented Dynamic Learning Environment and was first released several years ago by Martin Dougiamas, who developed the system, and Peter C. Taylor, who built the first web site running this LMS, both from The Curtin University of Technology, Perth, Australia “(Dougiamas and Taylor, 2003)”. Nowadays, Moodle is continually being improved upon by various groups of researchers and developers worldwide. Moodle has been conceived to be compatible, flexible and easy to modify. It has been written using the widely accepted PHP language, which runs smoothly on most computers with a minimum of effort. “Moodle is built in a highly modular fashion and uses common technologies such as shared libraries, abstraction, and Cascading Style Sheets to define the interfaces (while still working on old browser technology)” “(Dougiamas and Taylor, 2003)”.

**Project-related perspectives**

At the Faculty of Electronics, Telecommunication and Information Technology from the “Gheorghe Asachi University” from Iasi there is a pilot project going on which implements Moodle for the biomedical and networking engineering laboratories, coordinated by Prof. H.N. Teodorescu, m.c. The engineering approach consists of a Red-Hat enabled server which runs VMWare Virtual Server and a LAN (Local Area Network) of GUI – operated (Graphical User Interface) workstations which access educational data from the elearning server. An instance of the control panel of the elearning application is shown in Figure 1.
We have also developed specific interfaces for different levels of users of the elearning platform, which basically are the following:

a. the programmer interface deals with all the technical issues of the software environment and is not to be used by the teachers or students

b. the teacher’s interface allows each teacher to customize the course and laboratories to the certain need of his/her students, very user-friendly; provides assessment and class attendance tools; does not require advanced knowledge of software engineering, thus being used also by teacher of other specialities; not to be used by the students (Figure 2)

c. the student’s interface provides tools for individual learning, project management and other related instruments.

The software experience that I would like to share is from the perspective of hardware and software management of the network resources using virtual means (virtualization of computers inside computers). In the last decade, schools and universities willing to purchase/implement an elearning software system could not decide for certain hardware modules, due to the fact that on-the-shelf LMSs were delivered as a core and poorly customizable. It is obvious that an engineering learning platform requires different software tools (i.e. mathematical simulation software) in comparison with software dedicated to the history department (i.e. databases and library blocks). An educational entity was compelled to purchase the entire bundle of programs and the appropriate high expensive hardware; nevertheless, many modules remained not used and put weight on the university’s computer infrastructure. On the opposite, in certain situations, for instance when a large number of students are using a science simulation environment (Figure 3), there is a certain overload and a computational surge on the elearning server that hosts Moodle LMS, which frequently leads to computer crash and loss of working instances and educational data.
To overcome this drawback, the project team has developed a non-expansive and versatile approach, which basically consists in deploying the elearning server in a virtual environment.
called “virtual server” and dynamically balance the overload of computational power through the redistribution of hardware resources inside the virtual machine. This goal has been achieved using the VMware Server, a widely accepted reference in computer virtualization. According to literature “(VMware Server 2)”, it is a hosted virtualization platform, which is being installed like a common application on the existing server hardware. It works in the manner that it partitions a physical server into multiple virtual machines. “A virtual machine is a tightly isolated software container that can run its own operating systems and applications as if it were a physical computer” “(VMware Server 2)”. A thin virtualization layer partitions the physical server so one can run multiple virtual machines simultaneously on a single server. Computing resources of the physical server are regarded as a common bench of resources that can be allocated to virtual machines on controlled basis. VMware Server isolates each virtual machine from its host and other virtual machines, leaving it unaffected if another virtual machine crashes. Your data does not leak across virtual machines and your applications can only communicate over configured network connections. VMware Server encapsulates a virtual machine environment as a set of files, which are easy to back-up, move and copy “(VMware Server 2)”. 

Research methodology

In order to quantify the overload of data and measure the surge in computational power that occurs when a large number of users use simulation tools in elearning servers, we have employed ab- Apache HTTP server benchmarking tool. “ab is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving” “(ab)”. 

Our research employs an interpretive methodology that automatically provides easy-to-read graphs and charts, as shown hereafter (Figure 3).
Visual Outcomes
For a clearer image, we hereafter present screen footage with the behaviour of the e-learning server before and after being overloaded with computational requests.

Figure 5. Benchmark Records for Memory before Virtual Elearning Server Being Overloaded

Figure 6. Benchmark Records for Memory before Virtual Elearning Server Being Overloaded

Figure 7. Benchmark Records for Processor Load before Virtual Elearning Server Being Overcharged
Figure 8. Benchmark Records for Network Traffic after Virtual ELearning Server Being Overcharged

Conclusions

The key aim of this paper is to present an affordable solution to manage eLearning server overloads when coping with large amounts of computational requests, consequently to the extended usage of science and engineering simulation tools. It is not a solution – especially in these days, when financial means are on a high stake – to employ and pay large hardware infrastructure in educational environments just in case there will be a need someday. Our project is emphasized on employing freeware simulation tools that allow the balancing of dynamic overload in Moodle LMS by virtual means, thus achieving a proper management of the educational network and focus on further educational goals.

Acknowledgement

The author would like to thank to Prof. H.N. Teodorescu m.c. and to all other colleagues involved in the Moodle project for the coordination and the support provided.

References

Journal Articles

Conference Proceedings:

Internet Sources:
ab - Apache HTTP server benchmarking tool: http://httpd.apache.org/docs/2.0/programs/ab.html
IEEE: http://www.ieeetls.org:8080/Plone