

DESIGN AND PERFORMANCE EVALUATION OF A VIRTUALIZED MOODLE- BASED E-LEARNING ENVIRONMENT

Mario Manzo

IT Service Center, University "Parthenope" of Naples, Italy
mario.manzo@uniparthenope.it

Keywords: Moodle, Virtualization, E-Learning, Web performance, Distance Education.

In this paper, benefits and advantages arising from virtualization technologies, applied to the field of distance education, are discussed and highlighted. MOODLE (Modular Object-Oriented Dynamic Learning Management System), the most widely used LMS (Learning Management System), and a virtualization system, based on Xen hypervisor, are combined in order to create an effective e-learning environment. Commonly, the main problem about MOODLE

concerns the server workload, due to several factors, which results in a saturation of physical devices. To bypass this limit, the over-centralized load is deployed on a dynamic virtual machine. Experimental results prove that the proposed system can improve performances, produce low connection time and provide a powerful distance education environment.

for citations:

Manzo M. (2016), *Design and performance evaluation of a virtualized MOODLE-based e-learning environment*, Journal of e-Learning and Knowledge Society, v.12, n.3, 19-30. ISSN: 1826-6223, e-ISSN:1971-8829

1 Introduction

Virtualization (Chiueh & Brook, 2005, p. 1) involves allocation and administration of virtual version of hardware and software components. Success, diffusion and massive application can be attributed to some of its important features such as resources or performance optimization.

Even the field of distance education, and associated tools, is benefiting of virtualization. Generally, the attention is focused on the main tool adopted to spread teaching materials, LMS, and, specifically, the factors that limit its performances. The common problems are the workload generated by the activities of server hosting the service and the static nature of the hardware components. A sample solution could be a virtual architecture, for example working with MOODLE, designed and built to avoid excessive load in physical server. Different solutions about MOODLE and virtualization approaches have been proposed such as virtual cluster (Guo *et al.*, 2013, p.15), cloud computing (Morgado & Schmidt, 2012, p. 1), virtual distributed environment (Costea *et al.*, 2011, p. 1) or Learning Objects Virtualization (LOV) (Broisin & Vida, 2006, p. 629).

In this work, the goal is to demonstrate the potential of virtualization, applied to the field of distance education. For this purpose, a real case study, including MOODLE and a virtualization technology, working in academic context, with a resulting experimental procedure, is proposed.

The paper is organized as follows: the next two sections are dedicated to a detailed description of proposed virtual architecture and experimental results. While conclusions are reported in the remaining section.

2 Architectural details

This section provides details about the proposed virtual server architecture, hosting MOODLE. First, the management virtualization system, Xen hypervisor server, is introduced. Subsequently, configuration hardware and software, with related technical details, of the virtual architecture is described. Finally, a brief digression about the system login and backup procedure, containing important mechanisms for the security and strength of e-learning service, is provided.

2.1 Xen Server

System virtualization based on Xen hypervisor¹, with Citrix XenCenter version 6.2, is adopted. Xen is a virtual machine monitor for IA-32, x86-64, Itanium, and ARM architectures and is a type one hypervisor. Virtualization system hosts concurrently a set of virtual machines working with different

¹ <http://www.xenproject.org/>

services. Xen is designed to manage several guest operating systems, each one operating in a virtual environment or domain. Xen manages virtual resources working with all the virtual machines running on the hypervisor. Essentially, Xen stands between the physical layer, hardware components, and virtual machines. Domain 0 is the management domain and works in order to handle the system. It runs tasks about additional domains, management of the virtual devices, suspension, reactivation and migration of virtual machines. In the described system Xen works to manage different virtual machines hosting services such as e-mail server, distance education environment, digital library, etc. Finally, Citrix XenCenter is adopted to control virtual machines. Citrix XenCenter is a server virtualization framework which interact with Xen hypervisor and provides different tools such as administration, hosting, deploy and manage of virtual machines. The framework also provides tools for converting physical servers to virtual and virtual to virtual. Furthermore, it can support up to 500 virtual machines and 4000 virtual CPU for each host. Xen manages our general architecture composed of different blades. Blades host 8 or 24 CPU (Intel(R) Xeon(R) E5507 2.27 Ghz), with 48 GB of Random Access Memory (RAM) and physical disks of 4 TB overall space.

2.2 Virtual machine configuration

The proposed virtual architecture, that hosts MOODLE, is composed of three layers: host layer, network layer and storage layer realized respectively through server, network and storage virtualization. The host layer is composed of 3 virtual CPUs, 8 GB of RAM. CPU and RAM are the core for performance optimization. Any fault affects massively on the goodness of the service and on the performance of other virtual components (hard disk, network card, etc). The storage layer is composed of 3 hard disk, adapted for the installation file, data file and MySQL database. Storage virtualization is a technology that provides flexibility in terms of data physical location and storage capacity. It is a software layer that collects the I/O requests and maps them to the available physical positions. In this way, data management occurs by optimizing the physical space available on the disks. Finally, the network layer is composed of a virtualized network card. In order to improve the network throughput, the workload between the connected network interfaces, belonging to different virtual machines and hosting different services, can be shared. The performance can be enhanced only when the traffic is not balanced between the network adapters. For example, the incoming data flows can be managed by using scheduling algorithms which direct the traffic to network adapters that, at a specific time, results to be less charge than others. Another factors which affects the network performance is CPU usage. When the requests of the

CPU workloads increase, the effective network performance can degrade. As all types of virtual machines, the hardware features can be changed for any need at any time. In the proposed system, 4 GB of RAM, during installation, have been allocated. Subsequently, due to growth in accesses and educational materials, the RAM has been increased to 8 GB. Xen is chosen for this end. Xen works on the system hardware directly and creates a bridge, virtualization layer, between the system components and the virtual machine, by placing the system components in a set of logical computational resources. Subsequently, Xen can dynamically assign, these resources, to the guest operating system. The operating system works in virtual mode interacting with the virtual resources in the same way as physical resources through a policy of paravirtualization (PV). PV acts directly on the hardware in order to manage the sharing of resources allocated to different virtual machines and outperforms performance of full virtualization. The operating system and hypervisor work synchronously and efficiently, avoiding the overhead imposed by the emulation of system's resources. The operating system adopted is SUSE Enterprise Server 11. Suse operating system is adopted due to its features such as scalability, performance and advanced boot process, in addition to other improvements. Furthermore, differently from Microsoft family operating systems, Suse is free and open, light and, at the same time, adaptable and offers a wide range of solutions for all needs. The virtual machine is constantly monitored through the software Citrix XenServer. Monitoring of CPU usage, memory usage with service of receiving notifications when becomes dangerously high or low, monitoring of network usage with related information of data transmitted, received rate and network traffic. All these data are made understandable, even for unskilled users, through the real-time creation of 2D graph based of single components (CPU, RAM, etc.). A careful monitoring phase, performed both based on generation of real time data and human supervision, is essential for the virtualization system with purpose to provide better performances than standard architectures.

2.3 Backup system and access to service

Another key issue concerns security and integrity of files (database and data), located on main server, ensuring the proper behaviour of the service. The files are duplicated through a backup phase. A twin machine is created, with the same hardware/software features, and a procedure provides, every day, for a synchronization respect to main machine (files and database). Also in this case, it is a virtual machine managed in the same way as the main machine. Essentially, only the files that differ from main machine, specifically modified during daily activities, are updated on twin machine. This approach is adopted in order to minimize the network traffic between the machines and, thus, to

reduce the overhead. The comparison and copy phase occur during the night when the activities are more light. The backup is performed by rsync command. rsync is a protocol created for Unix-like systems with incredible versatility for data synchronization. Indeed, it includes authentication mechanisms for server and database. Our backup system uses this mechanism to migrate from the main machine to the twin machine data and database using a protection ensured by a login procedure (super user credentials are adopted). This protocol ensures reliability in the case of inefficiency of the main machine such as failure of hard drive, network or further problems.

The access to MOODLE service is managed through a RADIUS (Remote Authentication Dial-In User Service) (Deshmukh, 2012, p. 238) system that bridges, with the user and password, with databases of students and staff. RADIUS is the standard de-facto for remote authentication. Authentication is based on username and password. If authentication succeeds, the RADIUS server sends configuration information to the client (IP address, subnet mask, TCP port number, etc.). One of the limits RADIUS is the authentication exclusively based on passwords: the password is transmitted in the form of hash (MD5 algorithm) or response to a request for identification (CHAP-password). In the proposed system, user credentials were controlled separately in an external Oracle SQL database for students and in a LDAP database for teachers and technical staff. The same authentication credentials were used to access to MOODLE, so that the accounts were created automatically during the enrolment for students and staff. This policy has been adopted in order to access to services (email, WIFI, multimedia classrooms, etc) using the same credentials. Essentially, it is a unified access to all services available (single sign on).

3 Experimental setup

The proposed system is submitted to a massive experimental procedure. Results are achieved through benchmarking tools (JMeter², ApacheBench³ (AB), Httpperf⁴, Dotcom-monitor⁵, Weighttp⁶) with different features in term of approach to experimentation (language development, load request optimization, etc.), setup parameters and working environments (Linux, Windows, etc).

MOODLE hosts courses structured in different units, with multimedia material compliant to the SCORM (Maratea *et al.*, 2012; 2013), followed by more than 5000 users. Therefore, the workload, everyday, is high due to several

² <http://jmeter.apache.org/>

³ <http://httpd.apache.org/docs/2.4/programs/ab.html>

⁴ <http://www.labs.hp.com/research/linux/httpperf/>

⁵ <https://www.dotcom-tools.com/>

⁶ <http://redmine.lighttpd.net/projects/weighttp>

accesses. The first set of experiments is performed using JMeter through the entity named HTTP Thread Group. For the experiments are chosen various MOODLE platforms belonging to different academic institutions (Uniss, Units, Unibas, Unitus) in order to verify the feedback of alternative server configurations. From now, this experimental configuration will be named first setup with reference to comparison platforms adopted. Table 1 shows some server features related to first experimental setup. Being the goal of this phase, and particularly of this first setup, a comparison with different MOODLE installations only some of server features are made available because a complete description is of little importance for experimental purposes.

Table 1
SERVER FEATURES ABOUT FIRST EXPERIMENTAL SETUP

Platform	OS type	Web Server type	Database
Uniss	Linux	Apache	MySQL
Units	Linux(Debian)	Apache	MySQL
Unibas	Linux(Ubuntu)	Apache	MySQL
Unitus	Microsoft Windows Server	IIS	MySQL

Experiments are conducted considering the daily activities of the MOODLE platforms. This issue could influence and justify even more the use of the virtualization approach. The values in table 2 show the connection time, in milliseconds, for 10 experiments. Each experiment expresses the average value of 10 iterations. As can be seen virtual architecture has a better connection times than its competitors.

Table 2
RESULTS ON FIRST SETUP AND JMETER TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	30.1	24.4	32.5	24.6	30.3	29.2	28.2	28.5	27.5	28.7
Uniss	63.2	54.7	50.3	48.3	49.7	49.8	50.2	49.5	48.6	48.9
Units	190.9	117.1	70.2	70.8	72.6	68.1	70.5	81.6	74.3	70.8
Unibas	299.3	288.6	220.9	270.1	368.6	336.7	577.4	607.3	603.1	651.3
Unitus	121.2	107.6	180.5	195.2	171.1	130.4	117.6	186.3	186.2	181.9

Even in the second case same Thread Group entity and configuration parameters are used. Differently, in this case various LMS (Atutor Unicas, Docebo Unimib, Dokeos Unict and Olat Unimc) are chosen with the aim to provide a comparison with alternative LMSs. From now, this experimental

configuration will be named second setup with reference to comparison platforms adopted. Table 3 shows some server features related to second experimental setup. Likewise to first setup only some features have been made available because a complete description is of little importance for experimental purposes.

Table 3
SERVER FEATURES ABOUT SECOND EXPERIMENTAL SETUP

Platform	OS type	Web Server type	Database
Atutor Unicas	Linux (Debian)	Apache	MySQL
Docebo Unimib	Linux (Ubuntu)	Apache	MySQL
Doceos Unict	Linux (Debian)	Apache	MySQL
Olat Unimc	Linux (Debian)	Apache	MySQL

The values in table 4 show best performances produced by virtualized architecture, while Atutor platform produces the worst results. Certainly, the improvements are stronger due to the combined use of MOODLE and virtualization approach.

Table 4
RESULTS ON SECOND SETUP AND JMETER TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	13.5	9.6	9.3	9.3	9.2	9.4	14.1	16.7	14.3	15.2
Atutor Unicas	64.3	70.1	70.5	79.4	63.1	73.3	78.5	77.4	74.2	74.1
Docebo Unimib	21.7	39.8	16.9	16.1	19.2	18.4	18.4	17.5	17.6	19.9
Doceos Unict	29.6	29.8	12.9	67.1	68.4	71.5	78.3	76.6	75.2	75.9
Olat Unimc	16.3	79.6	48.6	40.5	51.3	42.2	55.1	58.9	60.5	62.8

The third set of experiments is conducted by using AB tool and first setup. Each experiment, in table 5 and identified by number, is performed by scheduling 1000 HTTP requests submitted to machine in form of block (10 requests for block). The final result expresses the average connection time value, in milliseconds. The results are almost completely leveled with a slight improvement for the proposed architecture. The main reason could be connected to the block based approach which produces constant workload for the machines under test. Indeed, during experiments, each LMS receives block based requests which are satisfied in more or less constant time.

Table 5
 RESULTS USING FIRST SETUP AND AB TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	4.425	3.235	2.324	4.999	3.828	3.454	5.031	4.618	6.743	4.191
Uniss	5.625	5.625	5.952	5.752	5.787	6.614	6.130	5.816	7.347	5.797
Units	6.257	6.148	6.326	6.190	6.188	6.206	6.268	6.217	6.259	6.197
Unibas	10.512	7.805	7.798	6.344	5.976	6.487	6.057	5.709	6.316	6.205
Unitus	2.763	2.845	2.752	2.768	2.729	2.738	2.817	2.932	3.834	2.731

The fourth set of experiments is produced using AB tool and second setup. Each experiment, in table 6 and identified by number, is performed using the same earlier setup and lead to the same observation of previous case. The requests are serviced, by each LMS, in constant time producing uniform results.

Table 6
 RESULTS USING SECOND SETUP AND AB TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	3.326	3.236	2.323	3.999	3.727	3.363	6.031	3.617	6.733	3.191
Aturor Unicas	6.626	6.626	6.962	6.762	6.777	6.613	6.130	6.716	7.337	6.797
Docebo Unimib	6.267	6.137	6.326	6.190	6.177	6.206	6.267	6.217	6.269	6.197
Dokeos Unict	10.612	7.706	7.797	6.333	6.976	6.377	6.067	6.709	6.316	6.206
Olat Unimc	2.763	2.736	2.762	2.767	2.729	2.737	2.717	2.932	3.733	2.731

The fifth set of experiments is performed using Httpperf tool and first setup. Each experiment, in table 7 and identified by number, is performed by scheduling 1000 HTTP requests submitted to server machine. The final result expresses the average connection time value, in milliseconds, of the 1000 requests. The results show a slight improvement for virtualized architecture. The assessment of this case is even highly dependent to the tool adopted which generate swarm of HTTP requests without any preprocessing step different from previous cases of block based approach.

Table 7
RESULTS USING FIRST SETUP AND HTTPERF TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	51.1	18.9	52.1	18.0	51.3	18.7	51.6	18.9	50.9	17.2
Uniss	52.0	19.2	52.4	19.1	52.0	19.2	53.3	19.1	52.2	19.1
Units	69.2	14.4	69.4	14.4	68.8	14.5	69.0	14.5	68.9	14.5
Unibas	49.2	20.2	51.9	19.2	48.0	20.9	48.9	20.4	49.9	20.0
Unitus	22.0	45.1	22.5	44.4	22.2	45.5	22.0	45.8	21.8	46.0

The sixth set of experiments is performed using Httpperf tool and second setup. Each experiment, in table 8 and identified by number, is performed by scheduling 1000 HTTP requests submitted to server machine. The final result expresses the average connection time value, in milliseconds, of the 1000 requests. As for previous case a mixed trend can be noted, through which a specific profile cannot be established, with an improvement about proposed architecture and a worsening about Docebo platform.

Table 8
RESULTS USING SECOND SETUP AND HTTPERF TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	17.9	17.1	60.1	16.2	59.9	15.1	57.3	15.1	54.8	15.1
Aturor Unicas	25.3	39.5	24.9	40.1	25.3	39.5	25.0	39.8	24.7	40.7
Docebo Unimib	151.0	6.6	272.1	3.7	373.8	2.6	152.0	6.6	157.4	6.3
Dokeos Unict	57.9	17.4	60.3	16.4	60.4	16.4	58.8	17.0	58.4	16.8
Olat Unimc	56.9	11.4	58.3	19.4	70.4	13.2	57.1	16.0	55.3	17.9

The seventh set of experiments is performed using Dotcom-monitor tool and first setup. This type of experiment is chosen due to different from previous ones because the HTTP requests are sent from different countries and cities (20 cities of different countries). Each experiment, in table 9 and identified by number, is performed by scheduling HTTP requests submitted to machine and the final result expresses the average connection time value, in milliseconds, of requests submitted from different cities. The values are placed within a limited range except for Units and Unibas which present high times. This simulation is very interesting because very close to a real user demand' for e-learning services in terms of workload request.

Table 9
RESULTS USING FIRST SETUP AND DOTCOM-MONITOR TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	1111.4	1442.2	1538.6	1005.1	1131.3	1228.2	1624.2	1476.8	1004.3	1664.5
Uniss	1268.8	1552.5	1584.6	1345.1	1232.5	1428.5	1694.5	1486.7	1164.3	1744.5
Units	20836.68	34672.42	45373.58	66960.05	69441.63	69503.58	61756.42	56713.78	69491.06	69501.79
Unibas	42694.25	50324.85	54191.55	65409.30	67054.70	50806.60	24252.30	70010.65	70001.75	70010.25
Unitus	5722.1	3946.7	3511.7	3559.6	3620.5	5008.5	4004.1	7555.0	4894.1	7850.9

The eighth set of experiments is performed using Dotcom-monitor and second setup. The experiments, in table 10 and identified by number, are conducted as in previous case and the same evaluation can be expressed. One critical issue concerns some high values about Dokeos platform, from the column 2 to the column 10, respect to homogenous trend.

Table 10
RESULTS USING SECOND SETUP AND DOTCOM-MONITOR TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	1009.02	1171.1	1446.2	1442.2	1229.2	1329.1	1265.4	1109.0	1248.1	1334.28
Aturor Unicas	1569.52	1571.5	1336.2	1442.2	1559.2	1559.6	1535.2	1469.8	1498.0	1476.29
Docebo Unimib	1053.62	1315.7	767.65	1239.5	911.3	949.5	772.5	772.2	821.6	1451.00
Dokeos Unict	1645.62	44320.3	65516.7	70036.4	70004.00	70016.3	70011.1	70016.95	70007.43	70006.0
Olat Unimc	1480.00	1625.3	1633.1	1637.6	1558.2	1802.1	1712.5	1558.2	1480.0	1637.60

The ninth set of experiments is performed using Weighttp tool and first setup. Each experiment is performed by scheduling 10000 HTTP requests submitted to machine in form of block (100 requests for block). The final result expresses the time, in milliseconds, spent to perform the total requests. The results, in table 11 and grouped by experiment number, show an uniform trend except for Unibas and Unitus platforms which presents high computational times.

Table 11
RESULTS USING FIRST SETUP AND WEIGHTTP TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	6783	21238	22312	15616	14128	10185	15438	18599	12120	17409
Uniss	9083	24418	25612	17916	16828	11985	16668	19499	16920	18009
Units	11098	18982	18137	18444	11460	18052	19178	17522	17188	12631
Unibas	40916	38664	35633	35573	36472	37237	35838	38184	36994	37660
Unitus	8607	142941	139388	139134	7226	13594	144112	133723	148322	7740

The tenth set of experiments is performed using Weighttp tool and second setup. Each experiment is performed using the same earlier setup. Unlike from the previous case, the results, in table 12, show an uniform trend without any inconsistency respect to the table columns.

Table 12
RESULTS USING SECOND SETUP AND WEIGHTTP TOOL

N. exp/ Platform	1	2	3	4	5	6	7	8	9	10
Virtual Moodle	8766	20031	20300	24017	8384	23247	24207	23747	24407	8241
Aturor Unicas	3446	24221	24301	24037	8184	24242	24262	24242	24902	8243
Docebo Unimib	8446	21231	22301	20037	8574	27642	23237	24007	24602	8873
Doceos Unict	1276	24223	21202	21117	8214	24952	23037	24232	27402	8743
Olat Unimc	2466	21471	20002	24017	8184	24112	24007	24247	22102	8741

Conclusions

In this paper, MOODLE is designed to work using virtualization technologies. The main goal was to manage and allocate different computing resources, for example virtual CPU, smoothly and dynamically, to overcome in a shortest time the user needs. The effectiveness of this solution, in a distance education environment, is demonstrated through comparisons with standard server technologies, hosting MOODLE or different LMS, using benchmarking tools. Experimental results, performed in different conditions, parameters and work environments, demonstrate that virtual architecture can suitably enhance and stabilize overall system performances by creating advantages and benefits with the purpose to improving central server management. Future work is in trying to improve the virtual machine architecture in order to optimize MOODLE performances.

REFERENCES

- J Broisin and P Vidal (2006), *A single sign-on mechanism for authenticating users across a distributed web- based learning environment*. In Information Technology Based Higher Education and Training, 2006. ITHET'06. 7th International Conference on. IEEE, 629–638.
- Susanta Nanda Tzi-cker Chiueh and Stony Brook (2005), *A survey on virtualization technologies*. RPE Report (2005), 1–42.
- Sergiu Costea, Vlad Dumitrescu, Răzvan Rughiniș, and Mihai Bucicoiu (2011), *MiriaPOD A distributed solution for virtual network topologies management*. In Roedunet International Conference (RoEduNet), 2011 10th. IEEE, 1–5.
- R. Deshmukh (2012), *Interactive Remote Authentication Dial in User Service (RADIUS) Authentication Server Model*. In International Conference on Wireless and Mobile Communications. 238–241.
- Xin Guo, Qing Shi, and Danjue Zhang (2013), *A Study on MOODLE Virtual Cluster in Cloud Computing*. In Internet Computing for Engineering and Science (ICICSE), 2013 Seventh International Conference on. IEEE, 15–20.
- Antonio Maratea, Alfredo Petrosino, and Mario Manzo (2012), *Automatic generation of SCORM compliant metadata for portable document format files*. In Proceedings of the 13th International Conference on Computer Systems and Technologies. ACM, 360–367.
- Antonio Maratea, Alfredo Petrosino, and Mario Manzo (2013), *Generation of description metadata for video files*. In Proceedings of the 14th International Conference on Computer Systems and Technologies. ACM, 262–269.
- Eduardo Martins Morgado and Renato Schmidt (2012), *Increasing MOODLE resources through cloud computing*. In Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on. IEEE, 1–4.