VM4LMS - Virtual machines on education

Victor Jesus Farinha Agrela

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. João Nuno de Oliveira e Silva

Examination Committee
Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. João Nuno de Oliveira e Silva
Member of the Committee: Prof. Daniel Jorge Viegas Gonçalves

November 2020
I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.
Dedicated to my parents...
Acknowledgments

I would like to thank to my family, to my supervisor Prof. Joao Nuno Oliveira e Silva, to my girlfriend Weronika Samon and my friends for the support and help that I have been given through my years at the university and all my life.
Resumo

O e-learning revolucionou tecnologicamente a área da educação, utilizando métodos de aprendizagem na qual utiliza a tecnologia para melhorar o desempenho dos alunos. Neste trabalho são falados três destes métodos, o Learning management Systems(LMS), os laboratórios virtuais e as maquinas virtuais(VM). O LMS é um dos softwares mais implementados na educação. Estes LMSs têm sido utilizados na comunidade académica para a administração, documentação, avaliação, etc. de cursos. Na área dos laboratórios virtuais, estes permitiram a substituição dos laboratórios físicos, utilizando software para virtualização de ambientes e equipamentos. Para a simulação de computadores foi criado o conceito das VMs. Uma VM tem a capacidade de criar virtualmente o hardware necessário para criar um computador virtual. A criação deste ambiente virtual permite que um sistema operativo funcione como se estivesse instalado num computador normal.

Hoje em dia não existe nenhuma ferramenta permita a criação de laboratórios virtuais para a utilização VMs através de qualquer LMS. Só existem LMSs modificados na qual permitem a integração de VMs. Uma solução para este problema seria a criação de uma aplicação externa na qual pudesse permitir o uso de VMs nos LMSs através de laboratórios virtuais.

Neste trabalho, foi criado um programa, VM4LMS, na qual permite o uso de VMs nos LMSs através da criação de laboratórios virtuais. VM4LMS estabelece uma ligação aos LMSs através do um protocolo(LTI) que esta integrado em muitos LMSs. Na parte da virtualização das maquinas é utilizado um sistema de virtualização em cloud(OpenStack) ou o computador dos alunos(VirtualBox).
Abstract

E-learning appeared as a new pedagogical technology supporting the methods of learning. It became a preferable learning method that suits the modern requirements and is tailored to the needs of current students. At the moment, there are many softwares, such as Learning Management Systems (LMSs), that can be implemented in the academic community to serve various purposes (such as administration, documentation, evaluation etc) of courses online. Similarly, virtual laboratories allow for the substitution of the physical labs, using software for the virtualization of environments and physical equipment, therefore making them a very valuable tool. A virtual machine (VM) can create a virtual hardware in the host computers to be used by a new operative system (OS). Up to date, there are no programs that allow the use of VMs through the LMS. Therefore, in this work, I developed a program that connects the virtual laboratories (that use VMs) with the LMSs. In the most common LMSs, it is possible to create external applications that connect and interact with the LMS. In this thesis I present the construction of a program that creates virtual laboratories on which the VMs are used. The labs are created by the teacher and accessed by the students through the LMS.

Keywords: Virtual Laboratories, LMS, LTI, Machine virtualization, OpenStack.
Contents

Acknowledgments ........................................ vii
Resumo .................................................. ix
Abstract ................................................ xi
List of Tables ........................................... xv
List of Figures .......................................... xvii

1 Introduction ................................................ 1
  1.1 Problem statement .................................... 2
  1.2 Objectives .......................................... 2
  1.3 Results ............................................. 3
  1.4 Thesis Outline ...................................... 3

2 Background .............................................. 5
  2.1 Virtual laboratories .................................. 5
    2.1.1 Virtual laboratories without virtual machines. .... 6
    2.1.2 Virtual laboratories with virtual machines ....... 7
  2.2 Learning Management System ......................... 8
  2.3 Learning Tools Interoperability (LTI) ................. 10
  2.4 Virtualization ...................................... 11
    2.4.1 Local virtualization ............................ 11
    2.4.2 Cloud Virtualization ........................... 12

3 VM4LMS .................................................. 17
  3.1 Users .............................................. 17
  3.2 VM4LMS Requirements ............................... 17
  3.3 Program architecture ............................... 18
  3.4 Execution steps .................................... 19

4 VM4LMS Implementation ................................ 21
  4.1 Support Technologies ................................ 21
    4.1.1 Virtualization .................................. 21
    4.1.2 VM4LMS ......................................... 22
List of Tables

2.1 NVMVLabs ................................................................. 7
2.2 V Labs. ............................................................... 8
2.3 LMS. ................................................................. 10
2.4 Table caption shown in TOC. ................................. 13
2.5 Cloud comparison. ............................................. 16

4.1 LMS difs. .......................................................... 23
4.2 End points. ....................................................... 26
4.3 End points. ....................................................... 26

8.1 After Scenario Questionnaire. .................................. 53
8.2 System Usability Scale questions. .............................. 54
### List of Figures

2.1 Full Virtualization VS Paravirtualization VS Hardware assisted virtualization .......................... 12
2.2 Cloud services .................................................................................................................. 14

3.1 Local’s laboratory ......................................................................................................... 18
3.2 Cloud’s laboratory .......................................................................................................... 19

4.1 VM4LMS Database ........................................................................................................ 25
4.2 UML - Sequence diagram of a local laboratory ............................................................. 27
4.3 UML - Sequence diagram of a cloud laboratory ............................................................. 29

5.1 Professor’s view of VM4LMS .................................................................................... 32
5.2 Local laboratory ............................................................................................................ 32
5.3 Cloud laboratory .......................................................................................................... 33
5.4 Configuration of a cloud laboratory ............................................................................... 34
5.5 Configuration of the VM ............................................................................................. 35
5.6 Student's view of the introduction text. ...................................................................... 36
5.7 Student’s vision of the laboratory .................................................................................. 36
5.8 Student's view of files ................................................................................................. 36
5.9 Teacher’s view of the students files ............................................................................. 37
5.10 Students population .................................................................................................... 37
5.11 ASQ answers for the first task. .................................................................................... 38
5.12 ASQ answers for the second task. ............................................................................... 39
5.13 ASQ answers for the third task. ................................................................................... 39
5.14 SUS answers .............................................................................................................. 40
5.15 Teachers ASQ answers for first and second task. ......................................................... 41
5.16 SUS answers .............................................................................................................. 41
Chapter 1

Introduction

The technological revolution has brought the society new ways of simplifying different aspects of life. One of such facilitation was the creation of computer’s applications and platforms suitable for teaching and learning. The usage of technology in pedagogy has created a novel concept used around the world known as e-learning (electronic learning). The e-learning applications are now standardised and implemented into the university’s systems, being used as a method of teaching and tracking the progress of students [1]. The use of computing programs as a method of teaching has been studied by different institutions and the results obtained are promising [2] [3].

In the past years, e-learning has been developed in different areas using an array of different techniques. One of the most common applications used by the educational institutions are the Learning Management System (LMS) or Course Management System (CMS). LMS is a web-based platform that functions by acquiring practical and theoretical components through the participation between students and teachers. LMS has the capacity to provide varied types of content such as videos, documents and programs and can also be used to conduct tests, quizzes or as a submission portal [4] [5]. LMS is built with the anticipation to be user friendly; it possesses self-intuitive navigation and interface that catches the attention of the content uploaded or created by the teacher [6]. LMS allows the student to work anytime and from anyplace, supporting distance learning and allowing for the ease of accessing files. Technology as Learning Tools Interoperability (LTI) applications, provides tools that are created to interact with the LMS [7].

Another e-learning concept is the virtual laboratory. This is a type of lab are made to simulate real environments, where the student can practice remotely and with virtual simulations. One of the biggest use of these labs is in the field of networking. Where the networks are created virtually and composed of virtual equipment such as routers, switches, hubs, etc. On these networks are usually connected VMs. A VM emulates one computing system providing the same functionality as one physical computer. The emulation is run as one isolated system from the real machine, hence providing high security and isolation from outside. The software inside the VM can be previously installed and configured, and this allows the possibility of the laboratory to be easily customized. Another characteristic of the VM is the property of being multiplied, i.e., the same VM can be replicated and distributed in multiple users [8]. It is
important to note that most of the virtualization softwares support an application programming interface (API). The API allows for the complete manipulation of the VM from an external or internal application.

The virtualization of these VMs can be performed through the usage of clouds and local virtualization technologies. It is a scalable system that allows a versatile manipulation over the VM, not only for the execution but also for the replication and maintenance. The usage of the clouds has many benefits, for example, the efficiency and effectiveness in the use of the computing equipment and the reduction of power consumption through adjustment of the equipment characteristic for each VM [9] [10].

The use of virtual laboratories as a method of teaching has been applied in different fields such as computer science, engineering, natural sciences and others [3], obtained promising results. In engineering, Caminero et al. [9] and Damiani et al. [11] provided a system based on templates that can be reused for the creation of virtual laboratories in the field of network services in operating systems. Guler et al. [12] described a project of virtual laboratories that has been tested in courses like computer networking, operating systems, database, programming and security. There are other types of laboratories, where the virtual and factual laboratories are combined, in the case of Liu and Orban [13], who provides a system in which the virtual machines interact with physical components.

1.1 Problem statement

Nowadays the resolution of exercises in virtual laboratories are seen as a good method for learning and improving skills. Not only can they save money on buying equipment and are easier accessed, but can simulate a range of possible problems [14].

The existing virtual laboratories were created for specific cases, working as an isolated software. The flexibility and adaptation of these programs are limited when used in other types of laboratories. In the field of the LMSs, there are no applications providing the connection between the LMS and the VM and the use of VMs lacks in the e-learning applications. Up to date, a software in which a user could create and manage VMs for laboratories through LMS does not exist.

1.2 Objectives

The lack of tools that allows the use of VMs on the education can be solved through the creation of an application that is connected to the LMS. Knowing that VMs can be applied in different learning areas due to their vast array of configurations, and high popularity, it would be a good idea to have an application that can be used in different types of fields. This can be facilitated by an user friendly interface that is easy to interact with and where the students can improve their knowledge and train their skills with the help of theoretical and practical lectures through virtual laboratories exercises. With multiples options for the evaluation created by the teacher. An application that can be run in any operating system and can be accessed everywhere.

The objective of this thesis is, hence, the development of a tool that would allow the use of VMs through a LMS. This tool would allow the teachers for the creation of virtual laboratories composed by
VMs. The laboratory should be then accessed and solved by the students. The application needs to create, erase and manage the VMs uploaded by the professors and delivered to the students. The evaluation system needs to obtain files from the VMs after the conclusion of the exercises. This tool should allow the students to execute virtual laboratories and work in different areas. The application needs to work in parallel to the LMS and to the cloud. The VMs are preserved by the system with the possibility to be downloaded or executed in the cloud. Each student needs to be considered as an individual student where every of them can access the VMs through their logging from the LMS.

1.3 Results

As a result, the final application obtained in this project should satisfy all the objectives discussed before. This program has been named VM4LMS. VM4LMS can be connected with any LMS that supports the LTI connection. The application is able to run the VMs locally (on the student PC) or in the cloud. The VMs are connected into a virtual network with access to the internet. For the evaluation system a timer was added for the execution of the labs. This timer is optional and configured by the professors when programming the laboratory. The current implementation allows the execution of local VM using VirtualBox or on the cloud supported by OpenStack and it was integrated into Canvas.

1.4 Thesis Outline

This thesis is divided into more five chapters. An introduction around some existing virtual laboratories and an overview about the technologies available to be used in this project such as virtualization methods and LMSs. A third chapter with a summary on the program's structure and the requirements that the application needs to satisfy. The fourth with a detailed explanation on the software and technology used for the implementation of the program. A fifth chapter with the creation and resolution of one exercise to demonstrate the functionalities of the program and a conclusion about the requirements achieved. The sixth chapter is a conclusion and future enhancements of the work.
Chapter 2

Background

This chapter provides with an overview of the technologies and techniques related to the virtual laboratories such as the virtualization, LMSs and cloud computing. The information in section 2.1 describes the virtual laboratories that have been developed in greater detail and the fields where these are used. In section 2.2 the functionality of LMS is characterised and the difference between the existing LMSs discussed. Section 2.3 explains with detail the possible methods of virtualization either in the cloud or locally.

2.1 Virtual laboratories

Virtual laboratories are able to simulate and create events which are impossible to see in real life or difficult to be achieved due to the complexity of the task. For example, the examination of the lines in a magnetic field, measuring the voltage at different points of the electric circuits or the simulation of robotics arms [3] [14]. These simulations are easily executed virtually, using an array of software available on the market. Most of the times the virtual software allows the manipulation of its experimental parameters [3] [14]. It makes the laboratory more interactive, interesting and attractive for the students who can play with parameters, thereby increasing their engagement and understanding, and therefore effectiveness of studying. Virtual laboratories have a certain advantage over the real laboratories, which is the possibility to be run at any place where a computer can be used. [14]. In the last two decades the virtual laboratories have been implemented in many fields [3] [15] [16] [17]. The creation and the expansion of these virtual laboratories comes from the advantages of these labs. The most important seems to be purely economical, as running a virtual device can replace a physical laboratory with costly equipment [3] [14].

There are many projects in which virtual laboratories are used, primarily by institutions such as universities, with the purpose of supporting the students with the learning process. Most of these institutions have implemented the labs in their system or as an external application. The virtual labs are mainly focused on programs that create experience simulations and can be used in different disciplines such as sciences, engineering, computing and others [3]. The research on the existing virtual laboratories are
going to be divided into two types of laboratories. The first type is the one that does not use virtual machines and the laboratory is based in a software. The second type involves labs that use virtual machines:

### 2.1.1 Virtual laboratories without virtual machines.

Woodfield et al. [18], present a study about a project called Virtual ChemLab. It is a simulated laboratory based on the reproduction of various chemistry experiments. These simulations are made to help the students in the factual laboratory. ChemLab shows animated steps crucial for carrying out experiments, such as: measuring the pH, centrifugation, flame tests, heating. The program contains a laboratory book where results can be recorded and a window where the mixture tubes can be pointed. Another chemistry lab is presented by Tetko et al. [19]. They have developed the Virtual Computational Chemistry Laboratory (VCCLAB), with the main purpose of data analysis and calculation. In physics Ding and Fang [20] presented a simulation program for diffraction grating where the students are able to configure the experiment parameters.

Tlaczala et al. [21], presents an European project calls VccSSe (Virtual Community Collaborating Space for Science Education). The program is focused on instruments virtualization and has an educational purpose in different areas of science. The exercise in this virtual laboratory is a combination of the virtual instruments and dynamics models of physical laws. Another project was the TEALsim [22] developed by Massachusetts Institute of Technology (MIT). It is an open source software and has an objective to represent the electromagnetic phenomena.

Many virtual Laboratories are built for fields of robotics and network. Jara et al. [23] present a project where, through the use of a tool named RobUALab, students are able to interact with a real plant or have a virtual experimentation environment where they can practice. On the networks field, Liu and Orban [13] present a virtual remote laboratory where students are connected to a cloud composed by real equipment's such as routers or switches and practice with it.

Vargas et al. [24], Tawfik et al. [25], Sancristobal et al. [26], Terkowsky et al. [27] and Barrios et al. [28] have created their programs with a LMS connection. This connections are executed through a modification of the LMS or a simple application that can access the LMS files for information.

Table 2.1 presents a summary of virtual laboratories that have been implemented in different institutions that do not use virtual machines.
<table>
<thead>
<tr>
<th>Proposal</th>
<th>Field</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woodfield et al. [18]</td>
<td>Chemistry</td>
<td>Brigham Young University, USA</td>
</tr>
<tr>
<td>Tetko et al. [19]</td>
<td>Chemistry</td>
<td>Europe</td>
</tr>
<tr>
<td>Ding and Fang [20]</td>
<td>Physics</td>
<td>China</td>
</tr>
<tr>
<td>Tlaczala et al. [21]</td>
<td>Many</td>
<td>Europe</td>
</tr>
<tr>
<td>dos Santos et al. [22]</td>
<td>Physics</td>
<td>MIT, USA</td>
</tr>
<tr>
<td>Jara et al. [23]</td>
<td>Robotics</td>
<td>University of Alicante, Spain</td>
</tr>
<tr>
<td>Kim [29]</td>
<td>Robotics</td>
<td>University of Stellenbosch, South Africa</td>
</tr>
<tr>
<td>Liu and Orban [13]</td>
<td>Networking</td>
<td>Accenture Technology Lab, USA</td>
</tr>
<tr>
<td>Koretsky et al. [30]</td>
<td>Chemistry</td>
<td>Oregon State University, USA</td>
</tr>
<tr>
<td>Villar-Zafrí et al. [31]</td>
<td>Many</td>
<td>Universitat Politecnica de Catalunya, Spain</td>
</tr>
<tr>
<td>Abdulwahed and Nagy [32]</td>
<td>Engineering</td>
<td>Loughborough University, UK</td>
</tr>
<tr>
<td>Aziz et al. [33]</td>
<td>Gear Train Design</td>
<td>Stevens Institute of Technology, USA</td>
</tr>
<tr>
<td>Hashemipour et al. [34]</td>
<td>Mechanical and Manufacturing</td>
<td>Eastern Mediterranean University, Turkey</td>
</tr>
<tr>
<td>Vargas et al. [24]</td>
<td>Many</td>
<td>Spain</td>
</tr>
<tr>
<td>Tawfik et al. [25]</td>
<td>Electronic Circuits</td>
<td>Europe</td>
</tr>
<tr>
<td>Sancristóbal et al. [26]</td>
<td>Many</td>
<td>Spanish University for Distance Education, Spain</td>
</tr>
<tr>
<td>Terkowsky et al. [27]</td>
<td>Many</td>
<td>Europe</td>
</tr>
<tr>
<td>Barrios et al. [28]</td>
<td>Network</td>
<td>Colombia and France</td>
</tr>
</tbody>
</table>

Table 2.1: Virtual Laboratories without VMs.

2.1.2 Virtual laboratories with virtual machines

Caminero et al. [9] provide a study carried out over a system called TUTORES that has been developed by Universidad Nacional de Educación a Distancia (UNED). It aims to create and manage virtual laboratories. The system is focused in the cloud computing and virtualization principles. The virtual laboratories are based on reusable templates, with each template being composed of one or more VMs and its respective network.

Another virtual laboratory was built by Damiani et al. [11]. It is based only in open source technologies and composed by a XEN platform and a Gentoo Linux distribution made by University of Milan. Each student has their own Linux VM with all the administrator privileges. The access to the lab is provided on a browser, with an accessing link that opens the ssh shell of the VM. The structure of the program is composed of three elements. Firstly, an e-learning application that allows for the access to the course and laboratories. Secondly, a firewall that filters the connections and ensures that the VMs are secured and isolated and lastly the virtualization server that executes the VMs.

Guler et al. [12] introduces a virtual laboratory that has been used for courses in computer networking, operating systems, database, programming and security in the University of Michigan - Flint. It uses
the VMware lab manager that provides a user interface with an authentication for the access to the VMs. The managements of the VMs and network is executed by vCenter Server.

Li and Mohammed [35] and Stewart et al. [36] present virtual laboratories with the VMs installed in the student's computer. Both of them are focused in the virtual networks area. Their structures are very simple and the students only need the VMM and the virtual router installed in their computer. The VMs can be accessed via website.

Table 2.2 presents a summary of virtual laboratories that have been implemented in institutions and use VMs.

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Field</th>
<th>Institution</th>
<th>LMS integration</th>
<th>Execution environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vouk et al. [37]</td>
<td>Many</td>
<td>North Carolina State University, USA</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Doelitzscher et al. [38]</td>
<td>Many</td>
<td>Hochschule Furtwangen University, Germany</td>
<td>Yes - Private</td>
<td>Cloud</td>
</tr>
<tr>
<td>Wang et al. [39]</td>
<td>Network</td>
<td>Michigan Technological University, USA</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Bhosale et al. [40]</td>
<td>Network</td>
<td>VIT University Chennai, India</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Xu et al. [41]</td>
<td>Network</td>
<td>Arizona State University, USA</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Yan [42]</td>
<td>Network</td>
<td>Ningbo City College of Vocational Technology, China</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Wannous and Nakano [43]</td>
<td>Network</td>
<td>Kumamoto University, Japan</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Benzel [44]</td>
<td>Network</td>
<td>USC Information Sciences Institute, USA</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Guler et al. [12]</td>
<td>Many</td>
<td>University of Michigan, USA</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Damiani et al. [11]</td>
<td>Network</td>
<td>University of Milan, Italy</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Caminero et al. [9]</td>
<td>Many</td>
<td>Universidad Nacional de Educacion a Distancia, Spain</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Chan and Martin [45]</td>
<td>Network</td>
<td>La Trobe University, Australia</td>
<td>No</td>
<td>Cloud</td>
</tr>
<tr>
<td>Stewart et al. [36]</td>
<td>Network</td>
<td>Air Force Institute of Technology, USA</td>
<td>No</td>
<td>Local</td>
</tr>
<tr>
<td>Li and Mohammed [35]</td>
<td>Network</td>
<td>East Carolina University</td>
<td>No</td>
<td>Local</td>
</tr>
</tbody>
</table>

Table 2.2: Virtual Laboratories with VMs.

2.2 Learning Management System

The learning management system (LMS) is a software application that provides with all the tools required by the teachers to support their teaching process. It allows for the existence of distance learning courses
and home courses [4]. LMSs are used by many educational institutions around the world like schools and universities [46] [47] [48] [49]. The choice for using LMS is due to several advantages that this application offers. These tools are made for different objectives, usually their main purpose being to give the students mechanisms of learning on the way to prepare and perform their learning activities [48]. These tools can provide the students with different learning resources like PowerPoint slides from lectures, documents, programs or videos [49]. LMS offers the possibility to create groups, enabling their users for the formation of discussion forums or chats. The teachers can also track the progress of students by looking at the automatically generated statistics or through the reports from laboratories, quizzes, projects, assignments or other kind of exams completed by the student and submitted on the platform[47]. It is also possible for the teachers to receive the feedback on their performance (which is completed by the students) [46] [48].

There are various LMSs on the market. In this project the difference in the most commonly used LMS [50]: Moodle, Blackboard Learn, Schoology, Canvas is discussed.

- **Moodle**
  Moodle is a free open-source LMS written in PHP. It is highly customized and allows the developers to apply many modifications in the system. More importantly it is scalable (allowing a very large scale of users), who can be given various administrative roles; managing the roles and permissions individually. Moodle has personal dashboards in order to organize the curse trajectory and to manage the students. It has a good file management with easy drag and drop files in the server that, in turn, can be saved in personal clouds. The students have a notification system that can receive alerts. It has an excellent student tracking progress, allowing them to analyze their knowledge on different topics [51] [52] [50] [53] [54] [55] [56].

- **Blackboard Learn**
  Blackboard Learn is a software available as a Software as a Service (SAAS) and sold as a service. It is easily managed by the educators the creation of groups, evaluations and dead lines. It has a great evaluation tool, with many types of exams and with tools to verify the plagiarism [51] [52] [50] [55].

- **Schoology**
  Schoology is a cloud based LMS that supports course management. It is a simple tool for the educators. Schoology focuses their system in the use of socials communities and social learning. It allows the creation of private groups with the possibility of sharing documents. The interaction between student and teacher is really simple. It allows the creation of content like videos and audios through the application. Schoology possesses an APP where every types of role [50] [54] [55].

- **Canvas**
  Canvas is a web-based open-source LMS. It was created with a strong development in its open-source API and in a big pool of Learning Tools Interoperability(LTI) that interact with the LMS.
Canvas facilitates in the interaction of external tools in its system. It possesses a tool that helps the tracking of the student's progress, it shows the exercises done, participation and web pages visited in canvas. This tracking can also be done in teamwork. Canvas provides a method of learning by levels, this method is simple and easy to create, this method can also be used for exercises. The sharing of information is completely customized and its distribution through the different kind of users in the organization is also personalized [51] [53] [52] [50] [55] [56].

After analyzing the most used LMS in the market, it is possible to conclude the similarity in theirs tools for the course management, with some differentiation in the layout and administration. Next table shows some differentiation on the LMSs:

<table>
<thead>
<tr>
<th>LMS</th>
<th>Moodle</th>
<th>Canvas</th>
<th>Blackboard Learn</th>
<th>Schoology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open source</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Online-Web based</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mobile device</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Large enterprises</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Exam Engine</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Templates</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Data Import/Export</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Defined User Roles</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Grading</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>API Integration</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LTI Integration</td>
<td>LTI V1.3</td>
<td>LTI V2</td>
<td>LTI V2</td>
<td>LTI V1.1</td>
</tr>
</tbody>
</table>

Table 2.3: Differentiation of LMS.

2.3 Learning Tools Interoperability (LTI)

With the high utilization of LMSs in different areas by many institutions, the development of the LMS through the years became more innovated and flexible. This changes in the service are turning the system more manipulable with higher personalization by their users. One important feature on them is the capability to interact with external applications [7]. To keep the security of the LMS and to allow the use of external tools, the creation of the Learning Tools Interoperability (LTI) was essential. LTI is a tool that connects the LMS and the external tool when this one fulfills all protocols required [57] [7].

Through the connection the external tool gets a list of parameters with information about the course and the person who is assessing the application[7]:

- **user.image** - Image of the person.
- **lis.person_name_full, lis.person_name_given, lis.person_name_family** - Different choices for the name and surname.
- **user.id** - Unique ID that differs in all the users in the LMS.
• resource_link_id - Unique ID generated by the LMS that identify the link pressed by the user.
• roles - Role of the user in the LMS.
• context_id - Unique ID that identify the course.

With this information, LTI gives the developers more opportunities in the creation of programs for the students.

2.4 Virtualization

Virtualization is a software technique that through the sharing of one hardware, allow the execution of many virtual machines in only one machine. The virtualization can be done in the cloud or in a local computer [58].

2.4.1 Local virtualization

A VM works as an independent computer, it uses the hardware components (memory, cpu, network interface, etc.) of its host. This connection between the VM and the real computer is executed by the VMM. There are different techniques of virtualization, according to the needs of each application virtualization done by each VMM The techniques of virtualization are used according to the needs of each application, and each environment, there are mainly three techniques that are supported by the most of the VMM[59]:

• Full Virtualization

In full virtualization, the VMM creates a virtual hardware that is seen as a real hardware by the guest operational systems (OS), turning the guest unaware about the virtualization environment. One of the advantages of full virtualization is that the guest OS does not need to be modified to be able to run. This virtualization must be run in equipment with processors that support this type of virtualization, otherwise, its execution is slower[59] [58].

• Paravirtualization

In the paravirtualization, the VMM does not simulate the hardware in the manner as full virtualization does. Instead the VMM provides an API which creates an underlying hardware. This API is a software interface that allows the communication between the guest operating system and the underlying hardware. For this communication the guest OS needs to be aware that it is being virtualized and for that reason its Kernel needs to be modified [59] [58].

• Hardware assisted virtualization

Hardware assisted virtualization uses tools available in the host's hardware, this tools are used to differentiate the process that comes from the guest or host OS. This approach brought the possibility to build a VMM that can simulate the instruction made by the guest operating system that can create a conflict with the host operating system [59] [60].
There are many available options in the market for a software that creates the virtual machine. Here, the two most used VMMs, VirtualBox and VMware are discussed:

- **VirtualBox**

  VirtualBox, that was created by Microsoft and now is being developed by Oracle, is among one of the most used virtualization program for a desktop computer [61]. It supports full-virtualization [58]. VirtualBox supports multiples operating systems as a guest and as a host machine[58]. The possibility of pause and resume any virtual machine at any moment. Another important tool coming with these options is the chance of taking snapshots, this tool save the state of the virtual machine at the specific moment, enabling the user to return to that state in the future [61].

- **VMware**

  VMware is a well known company one of the leaders on virtualization software. It is possible to run the virtualization of one or more operating systems simultaneously in an isolated environment. It creates a complete virtual computer to be run in one physical computer, theses virtual computers can run different operating systems [58]. VMware sells its products in different types of services and prices. In this work it is compared three different software for local virtualization provide by VMware, the workstation player, workstation pro and fusion.

- **XEN**

  Xen is a free and open-source virtualization software. A VMM originally developed by University of Cambridge and now by Linux Foundation. It supports paravirtualization and full virtualization (only if using IntelVT and AMD-V technologies in the host). It only supports Linux as host OS [58].

### 2.4.2 Cloud Virtualization

Recently, the cloud computing has been developed and is looked as the technology for the next generation. It has been used in academic and industrial purposes. There are mainly three levels of services offered for cloud computing [62]:

<table>
<thead>
<tr>
<th>Model</th>
<th>VirtualBox</th>
<th>VMWare Workstation Player</th>
<th>VMWare Workstation pro</th>
<th>VMWare Fusion</th>
<th>XEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host OS</td>
<td>Windows, Linux, MacOS, FreeBSD, Solaris</td>
<td>Windows, Linux</td>
<td>Windows, Linux</td>
<td>macOS</td>
<td>Linux</td>
</tr>
<tr>
<td>Guest OS</td>
<td>Windows, Linux, MacOS, FreeBSD, Solaris</td>
<td>Windows, Linux, FreeBSD, Solaris</td>
<td>Windows, Linux, FreeBSD, Solaris</td>
<td>Windows, Linux, macOS, Solaris</td>
<td></td>
</tr>
<tr>
<td>virtualization types</td>
<td>Full Virtualization</td>
<td>Full Virtualization</td>
<td>Full Virtualization</td>
<td>Full Virtualization</td>
<td>Full Virtualization and Paravirtualization</td>
</tr>
<tr>
<td>Snapshots</td>
<td>yes</td>
<td>No</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Cost</td>
<td>Free</td>
<td>Paid</td>
<td>Paid</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Minimum Requirements</td>
<td>32-bit or 64-bit processor, 512 MB RAM</td>
<td>No Free 64-bit processor, 1GB RAM, 250 MB Disk</td>
<td>yes Paid 64-bit processor, 2GB RAM, 1.2 GB Disk</td>
<td>yes Paid 64-bit processor, 4GB RAM, 750 MB Disk</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4: Table caption.

- **Software as a Service (SaaS)**

  In SaaS, the provider is responsible for all the structure necessary to make the software available to the consumers via client interface or program interface. This software is run in a cloud infrastructure, and is maintained and supported by the provider [63] [62].

- **Platform as a Service (PaaS)**

  In PasS, the consumer is responsible for developing and deploying the application in the computing platform. In this case the consumer build the software using programming languages, libraries, services, and tools supported by the provider. The provider is responsible for the networks, servers, storage and other services [63] [62].

- **Infrastructure as a Service (IaaS)**

  In IaaS, the consumer has in a virtual form the control over the hardware, networking, and storage services. The software is developed and deployed by the consumer. All the cloud hardware is build by the provider [63] [62].
There are many options in the internet available to run virtual machines and others instances in the cloud. Two examples of this options are the OpenStack and OpenNebula, both of them are open source software. They provide the necessary tools to manage the instances in the cloud environment. The users can handle different virtual machines at the same time. Both services work as Infrastructure as a Service (IaaS), which means that the admins can add or delete instances from the cloud. They structure divided in different components [64]:

- **OpenStack**

  Openstack is an IaaS project that was initially developed by Nasa and Rackspace in 2010. It is a free and open source platform for building private and public cloud infrastructures. It is written in Python and Unix Shell. Openstack is scalable and compatible with many hypervisors, it has distributed architecture which makes it highly flexible [65] [64].

  Openstack divide all the architecture in core services in order to facilitate its use, these are the main components that support the infrastructure [64]:

  - **Nova** - It is the main part of the program. Nova provides services for the compute management of the servers. It can work with many virtualization technologies.

  - **Neutron** - It is a system that provide different networking services. It manages the networks and IP addresses. Neutron provides different networks models that adapts according to the user needs. Users are allowed to create their own network and connect on it servers and instances.

  - **Keystone** - Is a service that manages all users on Openstack and their access, this access can be previously configured by a user. The authentication can be done by multiple forms, such as username and password or through the API with tokens. It provide the tools to manage
the access of users to the existent instances in the cloud.

- Glance - Provides image management services. It can use stores images and use them as a template. It is able to save backups from the servers and provides information about the deployed servers in use.

- OpenNebula

OpenNebula is an open source IaaS project, with the aim to be flexible, scalable and user friendly cloud management. OpenNebula is principally written in C++, Ruby and JAVA. It is mainly used in private cloud such as clouster, working as a virtualization tool that manages VMs, storage and network management. It supports many types of hypervisors for example the KVM, XEN and VMware. It is build in a centralized architecture and not divided in components as Openstack, it has a front-end and a cluster nodes to run the VMs. It supports hybrid cloud, where the combination of local and public cloud-based infrastructure can be done. All the control and monitorization over the VMs, can be done by different methods, as for example, Command Line interface, XML-RPC API, Libvirt virtualization API etc [65] [64].

The comparison between the Openstack and OpenNebula is relevant in terms of architecture. C12G Labs is the community support of OpenNebula, this community does not depend on public financing but also in its service by offering internal changes in its system for especial occasions. Openstack is supported by a foundation with over 4500 individual members around the world, this foundation is composed of many international companies such as HP, Dell, IBM, NASA, Cisco, etc. In terms of Hypervisors Openstack shows up with more compatibility than OpenNebula. It is also important to reeefer that both cloud systems supports VM migration.
<table>
<thead>
<tr>
<th>Cloud system</th>
<th>Openstack</th>
<th>OpenNebula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Distributed</td>
<td>Centralized</td>
</tr>
<tr>
<td>Cloud service module</td>
<td>IaaS</td>
<td>IaaS</td>
</tr>
<tr>
<td>Supported Platform</td>
<td>Linux</td>
<td>Linux</td>
</tr>
<tr>
<td>Programming Framework</td>
<td>Python and Unix Shell</td>
<td>C, C++,Java and Ruby</td>
</tr>
<tr>
<td>Databases</td>
<td>SQLite 3, MySQL and PostgreSQL</td>
<td>SQLite and MySQL</td>
</tr>
<tr>
<td>Hypervisors</td>
<td>Xen, KVM, VMware, HyperV, Xen Server, LXC, VirtualBox, QEMU, Vsphere, UML</td>
<td>Xen, KVM, VMware</td>
</tr>
<tr>
<td>Cloud types</td>
<td>Private and public cloud and hybrid cloud</td>
<td>Private, public and hybrid cloud</td>
</tr>
<tr>
<td>Compatalibity with public cloud</td>
<td>Amazon EC2, S3, EBS and OCCI</td>
<td>Amazon EC2,S3, Native XML/RPC,OCCI</td>
</tr>
<tr>
<td>Load balancing</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VM migration</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2.5: Comparison of IaaS Frameworks.
Chapter 3

VM4LMS

VM4LMS is a system that integrates LMS and VM. It allows the professors to modify the VMs before they are published in the LMS. These VMs can be reusable in another virtual laboratory. The execution of the VMs is done in the local student’s computer or in the cloud, depending on the teacher preferences. The students will access these labs through the LMS, when entering in the lab and with the help of the LTI, the LMS will subsequently send information such as user, discipline, etc. to the application which will redirect the student to the virtual laboratory. This application is able to create, launch, terminate and erase the VMs. It has the capability to extract files from the VMs, these files are selected by the teacher when doing the registration of the laboratory.

3.1 Users

VM4LMS requires three different types of users. Each user has a set of functionalities available:

- Admin - Has the responsibility to configure the LMS for the creation of users (students, professors) and to authorize the use of the VM4LMS a LTI application. The admin needs to configure the cloud authentication in the VM4LMS.

- Professors - Have the ability to create virtual laboratories, configures the VMs and have the access to all the files from the students resolutions. The professor has the control through the LMS on which students can access the laboratory. VM4LMS saves individually the VMs that is uploaded by each professor.

- Student - Are the one that solve the labs, they have the access to the VMs that have been configured by the professors. Each student is handled individually by VM4LMS.

3.2 VM4LMS Requirements

The following requirements are needed for the system to be capable of running virtual laboratories:
1. Compatible with the highest LMSs.
2. Allow the use of VM as tool for teaching and evaluation.
3. Allow the professor to fully configure the VM.
4. Allow students to run provided VMs on the local computer.
5. Allow students to launch VMs on cloud platforms.
6. Allow the extraction of files from the VM before they are terminated by the students.
7. Allow the professor to define a deadline for the execution of the VM.
8. Guarantee that only students enrolled in the course can access such VM.
9. Reduce the burden necessary for students to launch the VMs.

3.3 Program architecture

The program’s architecture can be divided into two scenarios for a sake of simplicity. One for the case where the VMs runs over the student’s local computer (fig.3.1) and another one when the VMs runs in the cloud (fig.3.2). In both cases VM4LMS receives information from the LMS’s LTI, and with that data redirects the student for the specific lab selected. VM4LMS verifies if the person accessing the application is a student, professor or another entity and redirects to the specific laboratory.

When the VM runs in the local computer the VM4LMS takes the control over the VMM through the VMM controller. This VMM controller is another web framework that runs at the student’s computer. It is connected with the VMM and with the VM4LMS, these connections are done through the APIs from all the sides. The VMM controller receives instructions from the VM4LMS, this instructions, such as installing a new image, switching off the VM or take files from the VM, are applied over the VMM. All the VM images for the laboratories and the files obtained from them are saved in the VM4LMS database. The VMM controller is the same for all the labs.

Using the cloud to run the VMs for the virtual laboratory makes the VM4LMS to work differently compared with the previous method. In this case the VM4LMS takes control over the VM with the help
of the cloud’s API. Most of all the clouds computing have their own database, these databases store all VM images needed for the virtual laboratories. For the files obtained from the VMs, these are stored in the VM4LMS database. When this laboratory is uploaded by the teacher, the VM4LMS connects to the cloud to save the VMs. In the case where the teacher wants to configure the VM when configuring the lab, VM4LMS uploads the VM, runs it and delivers it to the teacher through the web page.

![Figure 3.2: Cloud’s laboratory](image)

### 3.4 Execution steps

For the execution of a virtual laboratory, VM4LMS needs to be previously configured with the LMS and cloud. When configured the professor can use the VM4LMS for the creation of laboratories.

When a local laboratory is executed by the student (fig. 3.1), the student needs to run the VMM controller at his computer and introduce a code provide on the LMS browser. After that the student needs to download the VMs and introduce them on the VMM controller. With all these steps the student is able to start the laboratory by pressing “Start laboratory” on the LMS browser. When finished the lab VMM controller takes the files and erases the VMs from the VirtualBox.

For the cloud laboratory (fig. 3.2), the student just need to run the machine at the LMS web page. VM4LMS launches the VMs in the cloud and delivered it to the student.
Chapter 4

VM4LMS Implementation

In this chapter it is explained how and which technologies are used in the development of the VM4LMS. Is taken into detail all the communications between programs and how they interact.

4.1 Support Technologies

In this section it is explained how every program and method was chosen for the construction of all the VM4LMS environment.

4.1.1 Virtualization

For the right operation of VM4LMS, the requirements for the VMM are focused on its ability to be controlled by the local web framework. One of the possible connections between them is through the use of the API that is integrated in the VMM. This VMM API needs to be easy to install in the student’s computer and capable to be used in many OS.

For these reasons the option of using the VirtualBox was taken, because it is a simple program and not as much complex as the VMware. To use the Virtualbox’s API, the student needs to install the VirtualBox Software Development Kit(SDK) in the computer. In order to get the files from the VMs, the VirtualBox Guest Additions needs to be installed in the guest OS, this can be done by the teacher when configuring the VM or by the student when solving the problem.

On the Cloud software virtualization, Openstack was selected due to its big amount of hypervisors and its big community compared to the OpenNebula. Important components such as Nova, Neutron and Keystone are used in VM4LMS for the implementation of the VMs. Their SDK are used for different purposes:

- To upload, erase, launch and stop VMs.
- To create networks and assign floating IP to the virtual machines. The floating IP turn the VM accessible from the public network, this association is important for the ssh connection between the VM4LMS and the VMs.
• To give the students a remote console access of the VMs.

• To store the VMs for the virtual laboratories.

The remote console access is obtained through the Nova's SDK. Nova enables the use of Virtual Network Computer(VNC), this VNC permit the sharing of a remote console or a remote desktop of the VMs. The access is obtained through a URL, this URL is open in the browser [66].

4.1.2 VM4LMS

VM4LMS is the main tool in this project, the program needs to be capable of:

• Managing the users individually (students and professors) that access the system and blocking the intruders.

• Supporting a big amount of student's requirements for their virtual laboratories.

• Supporting the LTI, this will help in the connection with the LMS.

• Interacting with the cloud operating system, in order to upload, deploy, manage and erase VMs from the cloud.

• Sending instructions and receiving information from the Web Framework that is running on the student local computer.

• Managing a Database, for storing files, VMs and others information for the labs.

• Getting files from the VMs when the lab is finished.

Therefore, the idea of building a RESTful API came up as a good solution to fulfill the requirements. There are many Web Frameworks available that can be used as solution, such as Bottle, Django and Flask. Django shows up as a large framework with many features built in it to help in the creation of complex web applications, is supported by a big community and it comes with an admin panel for the system administration. Flask is a small framework, very customizable, solid to scale, simple to use and has a big community. Bottle is another small framework, it is composed of only a single file module, its community is very small and it is used for simple cases. After this analyze, Flask was the choice taken for being simple to use and can perfectly satisfy the requirements.

For the Database used in VM4LMS, there are many options such as SQLAlchemy, peewee, Tortoise ORM, etc. In this project SQLAlchemy is used because it stands out due to its big community. SQLAlchemy is a Python SQL toolkit and a Object Relational Mapper(ORM). It gives the developers the full ability of the SQL. The VM4LMS's database is used for:

• Storing the material needed for the virtual laboratory, such as VMs and others.

• Storing the files obtained from the resolutions given by the students.

• Checking and registering new students or professors in the system.
The library Paramiko is used for the interaction of VM4LMS with the VMs running in the cloud. Paramiko allows the connections between two entities through the SSH protocol. This connection is encrypted and authenticated. The transaction of files are executed through another library called SCP. SCP uses the paramiko authentication to transmit the files under the scp protocol.

The Web Framework that is executed in the student’s local computer needs to accomplish some requirements for the right execution of the virtual laboratory:

- Simple to use, small and can be use in any OS.
- Be able to communicate with the VM4LMS, receiving instructions and sending data.
- It needs to control the VMs that are running on the VMM through its API.

With all these parameters and with the previous discussion about the Web Frameworks. Bottle was the option since it is a single file which can be easily downloaded and installed by the students. It is easy to configure and is very customizable.

The interaction with VirtualBox is made with the library pyvbox2. With this library the local Web Framework connects with the Virtualbox’s API. The API is used to install, run, stop, delete VMS and take files from it.

### 4.1.3 LMS

The LMS in this project is used for the creation of disciplines, registration of students/professors and the integration and configuration of the external LTI application VM4LMS. Many LMSs were taken into consideration:

<table>
<thead>
<tr>
<th>LMS</th>
<th>Community</th>
<th>LT1</th>
<th>C. Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moodle</td>
<td>+++</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Blackboard Learn</td>
<td>++</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Canvas</td>
<td>++</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Schoology</td>
<td>+</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 4.1: Differences between the LMSs

From the LMSs studied, Canvas was the option taken. It was chosen not only for its abilities on the course administration but also for its big development in the API and its interconnection capacity.

### 4.2 LMS integration technology

For the VM4LMS the integration with LMS is used for different purposes. One of them is for the creation of users, courses and laboratories. The second one is the security, only registered users in the LMS can access the Lab, this access is verified by the LTI.
Every time the users access the laboratory, the LMS through the LTI sends information to the VM4LMS (Appendix A). From this information it is used three parameters:

- **user_id** - It is an individual code that differs the users.
- **lis_person_name_full** - It is the name of the user.
- **roles** - It describes the type of user that enters in the virtual lab.
- **resource_link_id** - This is an individual code that differs with the links. Each link is considered as an individual lab.

One of the biggest advantages of the LTI is the protection on the data exchange between the LMS and the application.

### 4.3 VM4LMS Backend

This section describes the functionalities of VM4LMS and how were implemented. The structure of the program can be divided into two parts, the database and the end points.

#### 4.3.1 VM4LMS Database

The database of VM4LMS was built to satisfy all the requirements of the program and it is divided into different tables:

- **Student** - Saves the name and the user_id as primary key (PK) of students, both of them provides by the LTI. This table store all the students that enters in VM4LMS.
- **Laboratory** - Is used to store lab parameters, such as the introduction text, time, professor_ID and the link_id provided by the LTI.
- **Local_Machine, Cloud_Machine** - Save all the machines needed for the labs (in the case of the cloud, it only is store the ID of the VM in the OpenStack). It also saves the users and passwords for every machine.
- **Laboratory, Local_Lab, Cloud_Lab** - Is used to store the labs, the PK of this table is the link_id provided by the LTI. Local_Lab and Cloud_Lab are attributes inheritance from the table Laboratory.
- **CVM_Designated, LVM_Designated** - Save which VMs are used for each lab.
- **Cloud_Exercice, Local_Exercice** - Are used to store the students that have done the labs.
- **Active_CVM, Active_LVM** - Store the VMs that are being used by the students.
- **Files** - Stores the paths of the files that are taken from each lab.
- **File_Output** - Stores the files obtained from the students.
Figure 4.1: VM4LMS Database
4.3.2 VM4LMS end points

The end points are build to satisfy all the requirements of the system. Each end point has its own objective.

<table>
<thead>
<tr>
<th>End Point</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/launch</td>
<td>The first end point used when accessing the lab. It returns the lab according to link accessed</td>
</tr>
<tr>
<td>/downloadfile</td>
<td>This end point returns the files obtained form the resolution of the labs.</td>
</tr>
<tr>
<td>/upload</td>
<td>It makes the Upload of the laboratory, saves the VM and the extra information</td>
</tr>
<tr>
<td>/getcode</td>
<td>Check the secret key written by the student and sent by the local Web Framework. This code is given when accessed the lab and it verifies if it is the student the one who is doing the lab.</td>
</tr>
<tr>
<td>/install</td>
<td>It install and launch the VM image in the student's computer.</td>
</tr>
<tr>
<td>/end</td>
<td>Makes a request to the local web framework to takes the files from the VM, turn it off and remove the VM.</td>
</tr>
<tr>
<td>/launch_server</td>
<td>It launches the VM in the cloud and return the VNC console to the student.</td>
</tr>
<tr>
<td>/test_server</td>
<td>This end point is used when the teacher runs the VM in the cloud when configuring the Laboratory</td>
</tr>
<tr>
<td>/end_server</td>
<td>It takes the files and shut down the VM in the cloud.</td>
</tr>
<tr>
<td>/clean_lab</td>
<td>It cleans everything that is related to that lab.</td>
</tr>
</tbody>
</table>

Table 4.2: VM4LMS End points

For the local Web Framework the end points are implemented for the communication with VM4LMS.

<table>
<thead>
<tr>
<th>End Point</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Install</td>
<td>Installs the VM image in the local computer.</td>
</tr>
<tr>
<td>/VMon</td>
<td>Turns on the VM</td>
</tr>
<tr>
<td>/VMoff</td>
<td>Turns off the VM</td>
</tr>
<tr>
<td>/Copyfile</td>
<td>Copies the file from the VM</td>
</tr>
<tr>
<td>/Remove</td>
<td>Removes the VM image from the VMM</td>
</tr>
</tbody>
</table>

Table 4.3: Local Web Framework End points
4.4 Local execution Runtime

The execution of the local laboratory can be explained in different steps:

Figure 4.2: UML - Sequence diagram of a local laboratory
1. The students log in the LMS and access the lab through the web page of the course.

2. After that the LMS through the LTI send a request to the VM4LMS for the lab, the request returns a web page that gives the access to the local web framework, a secret code for the lab and the VM image with its username and password.

3. The student downloads the local web framework and the VM image.

4. The student inserts the secret code in the local web framework and the path of the VM image that was downloaded.

5. The local web framework sends a request to the VM4LMS with the code to verify if it is the student that runs the lab, if it matches it is showed a message in the console.

6. The student presses the button to start the laboratory which sends a request to the VM4LMS.

7. The VM4LMS sends a request to the local web framework to install the VM and to launch it after that, if any error happens it will be shown in the web page.

8. The local web framework installs and runs the VM. The steps of installation can be tracked through the console.

9. The student logs in the VM and does the lab.

10. When the student finishes or when the time ends, a request is sent to the VM4LMS to finish the laboratory.

11. The VM4LMS sends requests for every file from the VM.

12. The local web framework through the VMM’s API gets the file and sends it to VM4LMS.

13. VM4LMS, after getting the files, sends a request to the local web framework to stop and delete the VMM and after that, it is shown on the web page that the lab is finished.
4.5 Cloud execution Runtime

For the cloud laboratory execution the procedure is:

![UML Sequence Diagram](image)

Figure 4.3: UML - Sequence diagram of a cloud laboratory
1. The student logs in the LMS and access the lab through the web page of the course.

2. After that the LMS through the LTI sent a request to the VM4LMS for the lab, the request returns a
   web page that gives the access to the lab web page.

3. The student presses the button to start the laboratory which sends a request to the VM4LMS.

4. The VM4LMS sends through the SDK a request to the cloud to launch the VM. After launching it
   gets the VNC console and returns the URL.

5. The student logs in the VM and do the lab.

6. When the student finishes or when the time ends, it is sent a request to the VM4LMS to finish the
   laboratory.

7. VM4LMS assigns a floating IP to the VM to make possible the connection through SSH and SCP.

8. With the SCP VM4LMS takes every file from the VM.

9. After getting the files, VM4LMS stops the VM and deletes it. After that it is shown on the web page
    that the lab is finish.
Chapter 5

Evaluation

In this chapter, it is discussed how the program accomplished the requirements previously described in the section 3.2. Continuing with a demonstration of possible laboratories applications and its limitations. Concluding with a obtained result of surveys answered by teachers and students.

5.1 Functional requirement Validation

In this section it is explained how VM4LMS satisfies every requirement previously described in the section 3.2. Each requirement is explained individually about its integration in the system and its interface:

Compatibility with the LMSs.

VM4LMS is developed under the LTI standard technology. This technology allows VM4LMS to be used in all the LMSs that support it. After configuring the LTI in the LMS, every teacher is able to use the application. The configuration is done through three parameters: consumer key, shared key and the xml configuration file. All these parameters need to match either in the VM4LMS and in the LMS.

VM as method of teaching/evaluation.

The VMs are available through the laboratories. When creating a laboratory the teacher needs to file a group of parameters that complements the exercise. These parameters are divided in two sections, exercise and virtual machine. The section exercise is the same regardless to the type of lab:

- Text - Place that describe the lab, there can be written notes or instructions about the lab.
- Timer - Optional timer for the laboratories.
The section virtual machine allows the professor to choose between the local and cloud laboratory. The parameters for each lab differ depending on the option. For the local laboratory the parameters are:

- **VM option** - Possibility to upload a VM or to choose one from the database.
- **VM name** - VM's name to be save on the database.
- **username** - Username that is showed to the student to access the VM.
- **password** - Password that is showed to the student to access the VM.

For the cloud laboratory it is needed:

- **VM option** - Possibility to upload a VM or to choose one from the database.
- **Flavor** - It defines the memory Ram, number of cpus and memory capacity that each machine will use.
- **The disk format** - It is needed for the selection of the hypervisor.
• VM name - VM’s name to be save on the database.

• username - Username that is showed to the student to access the VM.

• password - Passwotd that is showed to the student to access the VM.

• Cloud-init(optional) - Code that allows the configuration of the VM when it is booted in the cloud.

• Output files path - Files that are taken from the VM after finishing the laboratory

<table>
<thead>
<tr>
<th>Virtual Machines</th>
<th>Exercise</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of lab:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear VM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cloud laboratory: (definition)

| VM No 5          |          |         |
| VM No 6          |          |         |
| VM open          |          |         |
|                 |          |         |
| Update new machine |          |         |
| Cloud-init       |          |         |
|                 |          |         |
| Fname            |          |         |
|                 |          |         |
| Date format      |          |         |
|                 |          |         |
| VM name/description |        |         |
| Username         |          |         |
| Password         |          |         |
| Cloud-init (optional) |     |         |
| Configure the machine |   |         |
| (optional)       |          |         |
| Output file path |          |         |
|                 |          |         |
| Add file path    |          |         |

Figure 5.3: Cloud laboratory

**Total configuration of the VMs for the laboratory.**

In both types of the labs the teacher has the opportunity to upload a VM image previously configured. In the case of the cloud laboratory, the teacher has the opportunity to run it in the cloud and customize it in the web page.

**Capable to take files from the VM in the cloud or from the VMM**

Every time that a laboratory is finished, the VM4LMS takes the files that have been configured by the teacher on every machine. This method allows a better evaluation by the teachers over the students.

It is also important to referee that the students are also able to see download their files.
Possibility to use a timer in the execution of the virtual laboratories.

The set of the timer on the virtual laboratories are programmed when configuring the laboratories, this parameter is optional. The counter start every time that the machines are running.

Executing the VMs in the cloud.

Running the VMs in the cloud do not only overcharge the student's computer but also makes the laboratory more practical and easier to be executed. The cloud brings the capacity to run several machines for each student at the same time, allowing this, laboratories more personalized by the teachers.

Using the student’s local computer for the execution of the VMs.

The ability to execute the virtual machines in the student’s computer allows them to work on the VMs in places where there is a low access to the internet or in laboratories where it is needed to connect the VM with real equipment.

Only the student registered on the course can enter the Virtual Laboratory

The VM4LMS takes into the consideration the users who have accessed through the LMS. When accessed the lab VM4LMS takes the information from the LTI and returns the respective laboratory. Considering that only the students can access the page course in the LMS, then only they will be able to enter the lab.

5.2 Example of a cloud exercise

In this chapter it is demonstrated the steps needed for the creation and execution of the cloud laboratory that is used on the survey. The laboratory has as objective the development and execution of the "hello world" in c.

Setup of the laboratory

The first step for the professor is writing down an introduction text, with a description about the lab what is it about(figure 5.4).
The second step is configuring the VM. At the section "Virtual machine" the professor needs to upload the VM and writing its respective username and password. Selecting the flavor (Ram, memory and number of cpus), disk format. At the end it is important to write file path where the student need to write the code (figure 5.5).

<table>
<thead>
<tr>
<th>Type of lab:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud VM</td>
</tr>
</tbody>
</table>

Cloud laboratory:
add machine

<table>
<thead>
<tr>
<th>VM No.0</th>
</tr>
</thead>
</table>

VM option:
New VM

Update new machine:
Choose File: clinfo-0.4.6-..._64-disk.img

VM's name/description:
VM_1

Username:
userdummy

Password:
123

☐ Cloud-init (optional):

Flavor:
name=cirrus256 - RAM=256 - CPUS=1 - Disk=1

Disk format:
QCOW2

Configure the machine:
Run machine

Output files path:

- /home/username/hello.c

Add file path

Figure 5.5: Configuration of the VM
On the student view, when opening the laboratory the introduction text appears at first (figure 5.6).

![Figure 5.6: Student's view of the introduction text.](image)

On the section "Virtual machines" the student is able to launch the VM and do the exercise (figure 5.7).

![Figure 5.7: Student's vision of the laboratory](image)

When finished the laboratory, on the section "Submit Files", the student is able to verify if he has all the files to submit by clicking on "check files". By pressing "Submit files" the student submits the lab and get access to the files.

![Figure 5.8: Student's view of files](image)

At the end the professor is able to access all the resolution files from the students on the section "Results" (figure 5.9).
5.3 Surveys

In order to understand the utility of the VM4LMS, a set of questionnaires were made to students and professors.

5.3.1 Populations

After the conclusion of VM4LMS, questionnaires were answered by students and teachers in order to obtain an opinion about the concept of the thesis and the program. It was taken 25 answers, 23 from students and from 2 teachers. All surveys were obtained at the Instituto Superior Técnico.

From these students, 19 have not finished the course and 4 are currently doing the PhD.
5.3.2 Questionnaires

The surveys were based on two standardized satisfaction questionnaires, the System Usability Scale (SUS) and the After Scenario Questionnaire (ASQ) (Appendix B). The SUS is used to evaluate the usability of the program while the ASQ to assess how difficult a user perceived a task in a usability test.

The student’s questionnaire is divided in four parts. The first three parts are composed by three tasks. For each tasks the student needed to answer one ASQ questionnaire. After concluding all three task with their respective ASQ questions a SUS questionnaire was answered:

- For the first task the student needed to turn on one VM through the LMS. For that it is necessary to logging in the LMS, access the course page, open the laboratory and then launch the machine.

- The second task was interrupting the exercise. The way to interrupt was up to the student, it could be done by multiple ways such as refreshing the web page, logging out and in the LMS, closing the web browser, etc.

- For the third task the student needed to write a program, the classic hello world in c and conclude the exercise by submitting the lab.

The teacher questionnaire is divided in three parts, two tasks with a respective ASQ and the SUS at the end. The first task was creating a laboratory and launching a VM. While the second one was verifying the submitted files from the students on the hello world laboratory.

5.3.3 results

For the first task, the students got a good impression on how easy was opening the laboratory and how user-friendly was the interface. In some situations when launching the VM, it took more time than normal. The variation of this time was caused due to the OpenStack, Canvas LMS and the VM4LMS be running in one computer with low resources.

![Figure 5.11: ASQ answers for the first task.](chart)
When interrupting the exercise many students found interesting the ability of the program for being able to keep the VM in the same condition as they had before the interruption. For other students the program should suspend the VMs even though knowing that the laboratory has an option for it.

![Image of bar charts](image1.png)

**Figure 5.12: ASQ answers for the second task.**

On the third part the student needed to write a program and submit the laboratory. The students did like the VM4LMS shows up which are files taken from the VM and gives the possibility to download the files.

![Image of bar charts](image2.png)

**Figure 5.13: ASQ answers for the third task.**

On all the three ask's questionnaires, some of the students answered Not applicable especially on the third task. Most of the time this answer was given due to lack of documentation/messages about how to use the program.

On the SUS questionnaires the students gave a punctuation about the program. They did like the idea, gave different opinions about how the VM4LMS could improve. Two students gave the idea that VM4LMS should give the ssh access to the VMs so the students could access them through another program instead of the browser. It is important to report that all the students knew what was a VM and
have work with them. Overall the students liked the idea, giving the opinion that it could be used in
different subjects where specific software or OS is required in certain labs. Another good opinion was
that it would bring equality for all the students because everyone would use the same resources.

![Figure 5.14: SUS answers](image)

For the teachers surveys, their answers were positive. The tasks were done easily with the only
bad point on the lack of help/information about how to fill the parameters. Both of the teachers had
ideas about how they would use this labs on their subjects, giving different opinions about what type of
exercises could made with this system. Some of the ideas were configuring VMs in a network, analyzing
a network traffic, diagnosis of network problems and creating client-server programs. Both of them would
work with multiples VMs and would like to have possibility to configure the network.
Figure 5.15: Teachers ASQ answers for first and second task.

Figure 5.16: SUS answers
Chapter 6

Conclusions

The use of VMs as a learning method is limited and one of the reasons is the lack of applications that would enable it access through the LMS. Therefore, there is a great interest in developing a program that could solve this problem by creating an application that could manage virtual laboratories (that uses VMs), accessible through LMS. To do that, different technologies and methods were studied. In addition, different LMSs were analysed in terms of administration and compatibility with external applications.

The research in the LMSs has found that the LTI is used in most of them and that it allows for the interaction with external entities. Other virtual laboratories that have already been made were also taken into consideration and their objectives, problems and the limitations were analyzed. Considering data gathered, and selecting the most appropriate technologies, VM4LMS was developed as a solution to the above mentioned problems.

VM4LMS is a web framework (Flask application) that can be easily connected to the LMSs that supports LTI. It allows the students, who are registered for the LMS’s courses, for execution of virtual laboratories. It permits for execution of multiples VMs per laboratory and can take files from the VMs at the end of each laboratory. The execution of the VMs are taken either in the cloud or in the local student’s computer. In the cloud virtualization, OpenStack is used as a cloud software. For the local computers method, the VirtualBox runs the VMs images meanwhile another web framework (Bottle application) is running in parallel taking the control over the VMs. In both types of labs the VMs are fully customized by the professor before publishing the lab. VM4LMS did satisfy with its principal objectives, the delivery of VMs to the students through the LMS. VM4LMS can be used for areas such as networking, programming, database, security and software.

6.1 Future Work

In order to make VM4LMS more powerful and practical, additional options of configuration need to be implemented in the system.

- Configuration of networks:
Openstack allows the use of different types of networks configuration, it allows the creation of subnets and routers. VirtualBox is also very customize in this area. These configurations are important for the networks laboratories. In the actual version of VM4LMS, for the local laboratory it is created a NAT network and in the cloud a network with a subnet and a router connecting the network to the public network.

• **Methods of evaluation:**

  Different methods of evaluation are possible to be added in the VM4LMS. Methods such as running commands on the VMs, uploading files or even saving clones of the VMs. All this methods are easy to be implemented since most of the connections between the VM4LMS and the VMs are already programmed.

• **Better user interface:**

  The user interface could be improved, turning the lab more symple and understandable. With more helpful information about about how is the exercise done and what is the porpoise.


Appendix A

Parameters sent from the LTI to VM4LMS

INFO in views: {
    "context_id": "4dde05e8ca1973bcca9bffc13e1548820ee93a3",
    "tool_consumer_info_version": "cloud",
    "tool_consumer_instance_guid": "794d72b707af6ea82cfe3d5d473f16888a8366c7.localhost:3000",
    "oauth_signature": "l/71CasXnAVsjqCpZdriI5PfkoU=",
    "context_title": "Disciplina1",
    "lti_message_type": "basic-lti-launch-request",
    "custom_canvas_workflow_state": "available",
    "lis_person_name_full": "admin@email.com",
    "context_label": "Disciplina1",
    "user_id": "535fa085f22b4655f48cd5a36a9215f64c0627838",
    "oauth_consumer_key": "365219856327596345986325",
    "custom_canvas_user_id": "1",
    "launch_presentation_locale": "pt",
    "custom_canvas_api_domain": "localhost:3000",
    "custom_canvas_enrollment_state": "active",
    "tool_consumer_instance_contact_email": "canvas@example.com",
    "tool_consumer_info_product_family_code": "canvas",
    "oauth_callback": "about:blank",
    "lis_person_name_family": ""
}
"oauth_nonce": "6IWwQsI6J3FrU6mzWrbNHBucrP5nWVopS5eJGUqIU",
"oauth_timestamp": "1570040277",
"oauth_signature_method": "HMAC-SHA1",
"oauth_version": "1.0",
"lis_person_contact_email_primary": "admin@email.com",
"tool_consumer_instance_name": "thesis",
"resource_link_id": "ae06e3eb883588f0a1c5897b98830dc93f47d8",
"resource_link_title": "link",
"roles": "Instructor,urn:lti:instrole:ims/lis/Administrator",
"custom_canvas_course_id": "1",
"lti_version": "LTI-1p0",
"lis_person_name_given": "admin@email.com",
"launch_presentation_return_url": "http://localhost:3000/courses/1/external_content/success/external_tool_redirect",
"launch_presentation_document_target": "iframe",
"custom_canvas_user_login_id": "admin@email.com"
## Appendix B

SUS and ASK questionnaires:

1. Overall, I am satisfied with the ease of completing the tasks in this scenario
2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario
3. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks

<table>
<thead>
<tr>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>Not applicable</td>
</tr>
</tbody>
</table>

Table 8.1: After Scenario Questionnaire.
<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I think that I would like to use this system frequently</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. I found the system unnecessarily complex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. I thought the system was easy to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. I think that I would need the support of a technical person to be able to use this system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. I found the various functions in this system were well integrated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. I thought there was too much inconsistency in this system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. I would imagine that most people would learn to use this system very quickly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. I found the system very cumbersome to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. I felt very confident using the system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. I needed to learn a lot of things before I could get going with this system</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2: System Usability Scale questions.