An Implementation of the Cloud Based School

Vasilios S. Belesiotis, and Constantinos P. Alexopoulos

Abstract—Recently, cloud computing switched from being an experimental technology assisting everyday users through various cloud based applications. In a similar fashion, the use of cloud computing can improve the logistic infrastructure of the school computer labs and promote the introduction of innovative educational practices to the teaching of computer science related courses, as well as courses of different disciplines. Due to the economic crisis faced by many European countries, Greece included, the cost of continuously upgrading school computer labs, in order to follow the latest technological trends, is forbidden. Cloud computing provides an affordable solution to this problem by enabling the reuse of the existing computer equipment of the school computer labs and investing exclusively to the cloud computing infrastructure, an investment which is much lower than that of upgrading all school computer labs. This article presents a proposal of an implementation of the cloud computing based school. The conducted experiments indicated that this implementation could support the teaching process with respect to Computer Science, Information and Communication Technologies applications for Secondary and Primary Education schools.

Index Terms—Cloud Computing; Education; School Computer Labs; ICT; Open Source Software.

I. INTRODUCTION

Cloud computing technology, briefly cloud, has invaded rapidly into our everyday lives in the recent years. Numerous applications are designed to offer to user's services like email, data storage facilities and much more, through cloud infrastructures. Despite the multiuser nature of the cloud, all users have the feeling that they own the cloud infrastructure. The end user feels like working in his or her own environment, autonomously and without realizing that millions of other users have access to the same wide environment. Cloud technologies introduce a new computing model that allows users to share hardware, open source software and applications across the Internet with an easy, system-unique and non-transparent, end-user way [1].

A. Theoretical background

In spite of the various definitions of cloud computing [2]-[5], cloud is commonly considered to have the following characteristics: On-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. According to these characteristics cloud computing allows the user to access integrated content, services and applications anywhere and anytime by using an internet connection and paying exclusively for the resources he uses. Moreover, the user does not have to invest in expensive hardware and software infrastructure, because these are provided by the cloud provider in a scalable manner. The basic services provided by cloud are divided into three categories:

1) Software-as-a-Service (SaaS): This service is designed to provide the user with applications over the Internet.
2) Platform-as-a-Service (PaaS): This service provides the tools to design and implement the above applications quickly and efficiently.
3) Infrastructure-as-a-Service (IaaS): Here we refer to the hardware (processors, which includes CPUs and GPUs (graphics processing unit), memories, network infrastructures etc. and the operating system that supports the two services mentioned above. Users are able to self-provision this hardware pool using a web interface that operates as a management console for the overall environment.

The term “as-a-service”, which describes the three above mentioned services, is widely used in cloud computing. It describes an extensive variety of services and applications offered to cloud users that can be accessed via the Internet, instead of installations on their local computers [6].

B. Cloud Computing in Education

The education sector is a pioneer in embracing and applying new teaching methods and tools, but exhibits different rates of acceptance and integration. Computer and internet technologies are two factors that affected teaching methods worldwide. Traditional teaching methods are adapted in order to permit the usage of modern technologies, while new methods are developed. Modern education systems break the limits of the traditional classes giving to anyone, anywhere and anytime, access to virtual classes operated by large educational organizations like MIT (Massive Open Online Courses).

In the recent years Primary and Secondary Education syllabuses are reformed introducing new lessons, like Computer Science, and new teaching method based on ICT. The investment on purchasing new school computer labs (SCL) facilitates this reformation. Since 2000, especially Greek Secondary Education schools, massively acquired computer equipment for establishing new SCL, mainly through European Union programs addressed to the improvement of secondary education. SCL follow the client – server architecture. Two different kinds of software are installed on the workstations and server of the new laboratories: network management software and educational software. This architecture has a high complexity and an important drawback, the difficulty of maintaining hardware and software. Teachers, mainly those with the role of teaching Computer Science, without the appropriate
specialization on hardware and software maintenance, are usually called to support the operation of these labs often without success. The selection of this architecture was dictated by the need that all required software must be installed on every single computer, because the limited or no internet access discouraged the usage of www applications instead of the local ones [7]. This, however, is no longer the case, since the Greek School Network or EduNET, the national network of the Greek Ministry of Education, Research and Religious Affairs (GMERRA), interconnects all school labs and provides them, apart from internet access, many other services such as email, web site hosting, and e-class services.

In the recent years, the Greek School Network (GSN) continuously improves its infrastructure and offers to schools high-speed and reliable internet connections via ADSL lines. At the same time, schools that are located in isolated parts of the country are connected to the Internet through satellite connections. Moreover, new teaching material is available through the Internet, stored in national educational content repositories, like Photodentron and Aesop, providing adequate teaching support for most courses taught in Greek schools. Moreover, Greek teachers have become familiar with the use of the Internet and its applications, the incorporation of modern teaching methods using digital learning contents and collaborative tools, mainly by attending training courses designed and operated by the GMERRA. On the other hand the extensive usage of modern appliances like smartphones, tablets and laptops, combined with the large number of applications designed for them, results to a vast part of the population to be experienced users of ICT technologies. Most students fall within this category, and, as a result, have embraced the usage of ICT technologies in their everyday life.

Cloud technologies have recently entered the Greek reality and their users are constantly increasing. New applications, as well as older ones redesigned to operate on the cloud infrastructure, are available to the end-user. For example, the Greek Public Sector, in the context of administrative reconstruction, has redesigned the web applications offered to the citizens. A typical example is the applications of the Ministry of Finance [8]. To cope with the increasing demand of cloud applications new cloud service providers have emerged from both the private and public sector [9]. Modern internet connections via Wi-Fi and Mobile Data Networks, in addition to numerous connected mobile appliances, have increased the demand for applications that benefit from cloud technologies.

In the field of education, cloud infrastructure was mainly created for research and teaching purposes, with the users being academic and research institutions [10]. Cloud infrastructure addressed exclusively for the Greek Primary and Secondary Education has not been developed, unlike the global community, where there is a growing interest in the use of cloud for the support of the teaching process. There have been a few exceptions mainly focused on the development of educational applications and their availability through the cloud or cloud file storage & sharing. However, to the extent of our knowledge there is no cloud infrastructure addressed for Greek Primary and Secondary Education schools, which enables the creation of virtual machines (VMs) with all the required software installed on them, connected to virtual labs, permitting students to access them from any place, not only from school Informatics and SCL. Only a theoretical study on the development of a framework under which the above mentioned infrastructure can operate and offer hardware and software services to the Greek schools has been proposed [11].

C. Research Questions

This work focuses on the following research questions: a) whether there is the possibility of having a Cloud based school under the Greek reality, b) whether there is a possibility to use a cloud IaaS service to create VMs and virtual labs and accessing them using the existing infrastructure, mainly that of the SCL and c) whether it is possible to use the existing educational software on the new computing platform (VMs and virtual labs).

In order to answer these questions, this paper describes an implementation of the cloud based school and investigates the usage of this infrastructure in a classroom under real conditions. In the following, the development phases of the cloud based school EduResearch project are analyzed both at the school lab and cloud level, proposing the required hardware and software to cover primarily teaching, while also outlining the management and training needs of schools. The authors present an implementation that is to a great extends independent from the technical infrastructure available at each educational unit. This is achieved by focusing on open source software, in order to reduce the cost of implementation, and allowing it to be manageable and updatable in centralized manner. The rest of the article is structured as follows: In section II we describe our implementation giving detailed information of the four phases that we followed. In section III we present the results after testing it in schools of the Secondary Education. In section IV is presented a discussion about the results of our test and finally, in section V, the future work on improving and enriching with new features the proposed implementation is described.

II. METHODOLOGY

The presented research project had an overall duration of nine months, from October 2016 and until June 2017. The project involved an analysis, design, implementation and testing phases.

A. Phase 1: Analysis

The initial phase is focused on the analysis of the problem, the recording in detail of the current equipment of the SCL (hardware and software) and how these labs could be incorporated in the teaching practice.

1) Hardware Description:

Client-server architecture, the dominant case in Greek SCL, is the one to which computers are connected to a LAN, and through a router, usually provided by GSN, to the Internet. In most cases, the client-server architecture is not in function, due to technical problems with the server. In these cases, software and hardware operate as independent desktops connected to the Internet. File sharing, printer sharing or other key features of the client-server architecture

DOI: http://dx.doi.org/10.24018/ejers.2018.0.CIE.641
are not exploited, and the local network is used only to share
the Internet connection among the clients. Expensive
software installed on the server for antivirus protection of
the LAN or for restricting and controlling the use of the
Internet in most cases has never been used. Regarding the
hardware, in most cases the labs consisted of desktop
computers assembled before 2005 (Fig. 1). Intel processors
with Pentium M, Celeron architecture or prior and
maximum memory of 256 MB are the usual configuration of
these desktops. Some labs are equipped with desktops with
more powerful processors, like Intel Core 2 or Intel i3 and
memory up to 2 GB, but these desktops are mainly used by
the teacher of computer science or by the technical
supervisor of the lab. Rarely do we find powerful desktops
that are used by the students. In order to upgrade the
hardware of their labs, many schools decided to use the
LTSP technology (www.ltsp.org). Accordingly, they
purchased only a server with a powerful configuration and
use the existing desktops as thin clients, in order to improve
the total cost ownership and also empower their technical
infrastructure. The lack of experience from lab operators,
the incompatibility between educational software, designed
mainly for the Windows operating system, and the operating
system of the LTSP technology which is based mainly on
Linux, led to poor results. As a result, many SCL
supervisors abandon this solution. Recently, card sized
computers have appeared in some SCL. The low cost of
purchase and the support offered from the Greek Open
Source Community (ELLAK) results to 189 schools relying
to this technology to upgrade their labs. An illustrative
example with a card size computer based lab, in this case the
Raspberry Pi 2 (https://www.raspberrypi.org/), is analyzed
in [7]. An important drawback of this solution is the
requirements for a powerful server with advanced software
and extensive knowledge about this technology from the lab
supervisor and operators. In terms of the operating system,
the dominant operating system is Microsoft Windows of
various versions starting from Windows XP up to Windows
10. A limited number of labs are using various versions of
Linux basically because the users of the labs have no
experience on the Linux operating system. In addition, the
use of SCL for teacher training on ICT technologies, which
is based on software running under Windows operating
system, has served as an inhibitory factor in the use of
Linux.

<table>
<thead>
<tr>
<th>OS</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIN 98</td>
<td>35</td>
<td>27</td>
<td>4</td>
<td>66</td>
</tr>
<tr>
<td>WIN 2000 PRO</td>
<td>112</td>
<td>105</td>
<td>0</td>
<td>217</td>
</tr>
<tr>
<td>WIN 2000 SRV</td>
<td>8</td>
<td>10</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>WIN XP</td>
<td>310</td>
<td>193</td>
<td>215</td>
<td>726</td>
</tr>
<tr>
<td>WIN VISTA</td>
<td>44</td>
<td>56</td>
<td>1</td>
<td>61</td>
</tr>
<tr>
<td>WIN 7</td>
<td>95</td>
<td>89</td>
<td>20</td>
<td>204</td>
</tr>
<tr>
<td>WIN 2008 SRV</td>
<td>27</td>
<td>17</td>
<td>4</td>
<td>48</td>
</tr>
<tr>
<td>WIN 2008 SRV</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>UBUNTU</td>
<td>49</td>
<td>13</td>
<td>1</td>
<td>63</td>
</tr>
</tbody>
</table>

Fig. 1. The first diagram shows the distribution of processor’s generation
per school type per year, while in the table on that follows the distribution
of operating systems, by school type, A: Lower High School. B: High
School, C: Vocational High School. We see that a fairly high percentage, is
considered old for supporting the educational process [12]

2) Software Description:
The software (Fig. 1) that is installed in the school labs is
as diverse as their hardware. Software serving the same
needs comes from different developers, is based on a vast
variety of developing technologies, and operates under
different operating systems and operating system versions.
We have discovered that even software from the same
provider is sometimes installed in different versions, which
do not offer the same functionality, in different labs or even
in different desktops in the same lab. Finally, the required
software for supporting the teaching requirements of the
school courses is not present in all labs due to lack of proper
licensing.

For the project needs the software usually installed in a
school lab is categorized as following:

1) General use software. This category includes Office
Suites, Multimedia software, Internet access
software, i.e. web browsers, software tools, i.e. data
compression tools.

2) Educational software. This category includes
software for teaching Computer Science courses,
Mathematics, Physics, Chemistry and History etc.
For instance, in the Greek educational system and for
teaching computer programming to students from the
fifth grade of the elementary school to the third grade
of the junior high school, Logo like environments are
used. Also, the Python programming language is
addressed to students of the second and third grade of
Vocational High School (EPAL in Greek) (Table I).

3) Administrative software. This category includes
software designed to facilitate the school’s
administrative requirements. Software for registering
the students’ absences and the grades, for compiling
the lessons’ timetable, for the management of the
school library are the most common ones. Moreover,
high schools are using web based applications,
operated centrally from the GMERRA, to support the
General Examinations, exams that induct students to
the Greek higher education institutions.

DOI: http://dx.doi.org/10.24018/ejers.2018.0.CIE.641
TABLE I: EDUCATIONAL SOFTWARE

<table>
<thead>
<tr>
<th>Software Type</th>
<th>Lesson support</th>
<th>School Administration</th>
<th>Teacher training</th>
<th>Premium school</th>
<th>Junior high school</th>
<th>High School</th>
<th>Vocational High School</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office (Word Processing)</td>
<td>Libre Office, Microsoft Office</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Spread Sheet Proc. Presentations</td>
<td><a href="https://www.libreoffice.org/">https://www.libreoffice.org/</a></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Sound Processing</td>
<td>Audacity</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.audacityteam.org/">https://www.audacityteam.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Image Processing</td>
<td>Gimp</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.gimp.org/">https://www.gimp.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>LMS (as web access)</td>
<td>Moodle</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://moodle.org/">https://moodle.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>CMS (as web access)</td>
<td>Joomla</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.joomla.org/">http://www.joomla.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Web Browser</td>
<td>Mozilla Firefox</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://mozilla.org/firefox">https://mozilla.org/firefox</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Utilities</td>
<td>7-zip</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://7-zip.org/">http://7-zip.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Questions Maker</td>
<td>Hotpotatoes</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://hotpot.uvic.ca/">http://hotpot.uvic.ca/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Idea Presentation</td>
<td>cmapTools</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://cmap.ihmc.us/">http://cmap.ihmc.us/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Informatics' Lessons</td>
<td>Scratch</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Greenfoot</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.greenfoot.org/">http://www.greenfoot.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Alice</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.alice.org/">http://www.alice.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Python</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.python.org/">https://www.python.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Java</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://oracle.com/java/">http://oracle.com/java/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Algorithmiki (Greek educational software for structure programming)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>AppInventor</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://appinventor.mit.edu/">http://appinventor.mit.edu/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Mathematics</td>
<td>GeoGebra</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://geogebra.org/">http://geogebra.org/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Physics-Chemistry</td>
<td>ChemSketch</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.acdlabs.com/resources/freeware/chemsketch/">http://www.acdlabs.com/resources/freeware/chemsketch/</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Interactive Physics</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.design-simulation.com/IP/index.php">https://www.design-simulation.com/IP/index.php</a></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

B. Phase 2: Design

Our project is designed to cover progressively all the needs of a modern Greek school in hardware and software in a simple, cost-effective, easy to operate and easy to upgrade way.

The solution we propose is the design of a cloud computing ecosystem in which all the necessary tools for knowledge transfer, interaction and collaboration exist. Due to the limited offer of cloud infrastructure in Greece we decided to design our ecosystem on the infrastructure of ~okeanos, operated by the Greek Research and Technology Network (GRNET)[10]. ~Okeanos is a new IaaS Service which gives to its users the ability to create their virtual computers through an advanced web interface without worrying about hardware or software. User can access their virtual machines or their virtual network using an internet connection and can have full control of them, including destroying or creating new ones with different configurations. Almost all Greek Universities and Research Institutes are using ~okeanos service because of its advanced features, the high educated and experienced personnel that operate the service, and the fully organized Helpdesk that supports all users, efficiently and in a timely fashion in the problems the face. More information about the ~okeanos service and detailed description of the capability of the infrastructure can be found in [10].

The design of our solution has two strands:

1) The cloud side, in which we design the hardware architecture of the virtual machines and their Operating Systems. The selection of educational software depending on the educational level is also described. Finally, an automated process is designed for creating easily VMs through images.

2) The user side, in which we design the process under which users from SCL access their VMs. The vast variety of hardware architectures and operating systems is an important factor in this case that has been taken into account.

In the following, we describe in detail the Design of our solution.
1) Cloud side design: Virtual machines

The main objective of this part of the design process is to find an easy way to create VMs. The web interface of ~okeanos allows us to create VMs in a simple way. Another important factor is minimizing the usage of the available hardware resources, like processors, memory and storage capacity. Every VM must meet the needs of the educational level it is targeting. This is dictated by the educational software that is used to the corresponding educational level. Also, a simple process for VM upgrade as the user changes educational levels is designed. This automation gives the user the ability to potentially have a life-long VM for educational purposes. With this design we overcome the static structure of the school lab making it flexible to meet different educational needs. After some experimentation using different operating systems and the most widely used educational software we came up with the minimum hardware requirements of the VMs which are: double processor architecture with 4 GB main memory and 60 GB storage capacity. The selected operation system is the ubuntuMATE OS, a free and open-source Linux distribution. UbuntuMATE is an official derivative of Ubuntu that uses the MATE desktop environment as its default user interface. After additional experimentation we reached the conclusion that the educational software that will be used in the project can run under this OS, while figuring out how to overcome some software and OS incompatibilities that we faced.

We created 12 VMs (Fig. 3), eleven of them are used for the experiments at schools (Junior High School, numbered from 1 to 6, High School, numbered from 7 to 11) and the remaining one, numbered 12, is used for testing software for other categories as depicted in Table II.

<table>
<thead>
<tr>
<th>School Level</th>
<th>Administration</th>
<th>Teacher Training</th>
<th>Primary School</th>
<th>Junior High School</th>
<th>High School</th>
<th>Vocational High School</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#12</td>
<td>#12</td>
<td>#1-#6</td>
<td>#7-#11</td>
<td>#12</td>
<td></td>
</tr>
</tbody>
</table>

A VM image for each category was created and through it the rest of VMs were built.

2) User side: Hardware & software

The main issue on the user side is that we have to cope with the variety of hardware and operating systems that are installed in the SCL. The software through which the users get access to the VM should run on different hardware and under different OSs. The positive aspect for the design of our project was the reliable internet connection offered by the GSN. We investigated different hardware architectures to make sure that all the SCL are equipped with desktops that are capable to operate the VMs. We experimented on laptops/desktops with processor architecture ranging from Pentium M to Intel i5. Also, we experimented on the use of Raspberry Pi 2 computers as clients for accessing VMs. The OS of all these machines was Windows, versions from Windows XP up to Windows 10 and various Linux distributions like Ubuntu, RedHat, etc. As software for accessing the VMs we selected X2Go (x2go.org) as an advanced Remote Desktop Solution. X2Go is an open source solution offering security, privacy and file sharing with good performance under low bandwidth internet connections.

C. Phase 3: Implementation

The basic ingredient of the proposed solution is the cloud. Since our solution is based on VMs, a cloud infrastructure with a reliable IaaS service is necessary. For this task, we decided to use ~okeanos infrastructure, due to our previous experience with this service for the deployment of software for administrative purposes in secondary education, as well as for its reliability. The next steps involved the creation of the VMs, the installation of the required software, and finally the evaluation of our solution in school labs under real teaching conditions. Following, we describe the implementation details of our approach.

Creation of VMs and users:

For the implementation and employment of our solution we obtained from ~okeanos operational center a repository containing the following resources: 800 GB file storage space, 1 TB hard disk storage space, 24 CPUs, 48 GB RAM, and 12 public IPs. These resources supported a total number of 12 VMs. The creation of VMs was completed using the web-based platform of ~okeanos. The configuration of the VMs followed the details presented in Section B. Depending on the school grade, the appropriate educational software was installed. An issue we had to overcome for this step was the following. For the reasons presented in Section B, a Linux-based operating system was selected. Although most of the educational software operates under Linux OS some of them are designed only for Windows OS. To overcome this incompatibility, additional software was installed on the VMs in order to emulate Windows OS under Linux OS, such as Wine (https://www.winehq.org/). Regardless of the extra computational burden, the results were more than encouraging. The installation process, as described above, was effort demanding and we decided to automate the whole procedure. Using the interface provided from ~okeanos, an image for each VM addressed to each school grade was created. Accordingly, the deployment for each grade was completed using the appropriate image. The images were uploaded to the ~okeanos OS repository and the user of the service can easily create a VM with the operating system and the required software. The desktop environment of each VM was redesigned in order to aid the students and the teacher to access only the functionalities required for their courses.

TABLE II. DISTRIBUTION OF VMs PER EXPERIMENT

Fig. 2. ~okeanos virtual machines
The client side configuration depends on the school lab infrastructure. In order to evaluate the design of our system, as described in Section 2.6.2, different clients have been tested to allow access to the VMs. Some of them are presented in Table III. X2Go, the selected software for remote access to the VMs, was installed. The credentials necessary for accessing the VMs are installed together with the software. Therefore, students executing X2Go are connected directly to their VMs, without having to remember IPs or usernames and passwords. Moreover, some extra tuning was made to X2Go in order to enhance performance in cases in which low bandwidth connections were detected.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel I5, Ram 8GB</td>
<td>Windows 10</td>
</tr>
<tr>
<td>Intel P5, Ram 4GB</td>
<td>Windows XP</td>
</tr>
<tr>
<td>Intel I3, Ram 4GB</td>
<td>Ubuntu Linux (<a href="http://www.ubuntu.com/">http://www.ubuntu.com/</a>)</td>
</tr>
<tr>
<td>Intel I3, Ram 4GB</td>
<td>Windows 7 / Ubuntu</td>
</tr>
</tbody>
</table>

In all the above, the appropriate remote access software X2Go was installed.

D. Phase 4: Testing

Before applying the proposed solution, in order to detect any problems of the deployment of our solution, a lot of experiments were conducted. In the following we describe the actual test on the site of two schools, 4th Junior High School of Vyronas and 5th High School of Vrononas (Athens, Greece). In both cases, the authors installed the client side software and gave the teacher the minimum information about how to connect to the VMs and execute the required software. This information is transferred to the students from the teacher, without any additional information about cloud VMs.

1) Junior High School Testing

The school lab was equipped with three desktops with Intel Core 2 architecture, 256 MB RAM running Windows XP OS, and three desktops with Intel Celeron architecture with 256 MB memory running Windows XP. The available internet connection of the lab is a DSL connection with a theoretical speed of 24 Mbps provided from GSN. Before executing our experiment, the speed of the DSL line was measured and found at 6 Mbps. Twelve students of the third grade divided into groups of two used the desktop computers to connect to the VMs. The installation of X2Go was completed with no difficulties and students used the icon of the application on the desktop client to activate it. The class that these students attended was Computer Science and the course was Introduction to Computer Programming. The students were examined on the understanding of program flow control and iteration commands using Scratch (Fig. 5). Students were provided with worksheets and were asked to implement a traffic light simulator. After some instruction from the teacher on how to implement the simulator students connected to the corresponding VM and started to work (Fig. 6) with the Scratch environment, which was familiar to them from previous lessons, running it on the VM. We didn’t notice any malfunctions in the software. The students worked as they did in previous lessons without noticing the difference between executing Scratch on the desktop or on the VM. At the end of the lesson, a short discussion with the teacher and the students validated our observations.
2) High School Testing

The High School lab was equipped with Raspberry Pi 2 Model B computers with a server with an Intel i3 processor with 8GB RAM. The available internet connection was the same as in the previous case with its actual speed measured at of 8 Mbps. We followed the same procedure with the Junior High School case, installing the appropriate versions of X2Go for Raspberry environments. This lab has been used in the past for other experimental project such as [7]. Fifteen students from the third grade (Fig. 8) organized in groups of three attending the class of Economic and Computer Science Studies took part in the experimental study. The lesson was about Computer Programming and the programming environment (Fig. 7) they used is “Algorithmiki” [13]. This educational software is designed to help the students understand the design of algorithms and to introduce them to structured programming using advanced programming techniques. A drawback of this software is that it operates only under Windows OS and there is no Linux version. Also, since the interface is in the Greek language, Greek language support had to be installed on the VMs. In order to overcome the OS limitations of the software, a version of Wine, a software that emulates windows environment in Linux machines, was installed on the VMs. Greek language support was also installed by following the instructions from the Greek Linux Community found in http://www.linux.gr. We followed the same procedure as in the previous case to inform the teacher and the students how to use the new environment. Although in this case the working environment was more complicated due to the extra software required for the experiment, the results were very encouraging. The VMs performed very well, regardless of the additional complexity and computational burden, and the students worked without any delays. We tried a more resource demanding software designed for audio processing, Audacity (https://www.audacityteam.org/), which was previously installed on VMs. The students were asked to process some audio files, stored in the VMs. Again, the VMs responded very well despite the resource heavy nature of the application. No delays or malfunctions were observed and students worked smoothly in a collaborative way enchased by the homogeneity of the working environments provided by the VMs.

III. RESULTS

The employment of our pilot project in schools demonstrates the feasibility of the use of cloud to improve the hardware infrastructure of SCL. Under this architecture, current infrastructure is used efficiently for connecting users with VMs and all the required software for teaching can be installed on cloud without major changes.

With respect to the students, we noticed that they did not have any difficulties to work with VMs. On the contrary, made positive comments as the homogeneous work environment gave them an opportunity for better collaboration and data exchange. Even though the applications were running online, the bandwidth was sufficient to support the increased demand of data transfer between clients and VMs. Some minor delays observed in the same labs did not affect the development of our experiments. Also, the performance of some teaching software was improved because of the better architecture of the VMs compared with that of the local client. Specifically with respect to Raspberry Pi 2 clients, the performance of the applications was improved dramatically as the applications were not running locally. Students didn’t notice any change in the environment of the applications they used except from some extra steps that they had to take in order to connect to the VM.

After the experiments, thorough discussions with the teachers of the classes indicated that the proposed solution would be particularly useful because of the uniformity of the working environment, the simplicity of its application, and the lack of extra burden for its employment.

IV. DISCUSSION

This work demonstrates our proposal to employ cloud computing to alleviate the infrastructure of an SCL efficiently and cost effectively. Cloud computing can transform the computer lab from a static one, based on the Client - Server architecture with standard software, to a dynamic one with scalable hardware and software to meet the ever increasing demands of teaching lessons using ICT technologies. The problems of the traditional lab like hardware failures, software updates, and expensive technical support can be minimized if not eliminated. The dynamic nature of the virtual lab can break the barriers of the school building and the lab can be brought to students’ and teachers’ homes just by using an Internet connection. The new working environment encourages collaboration, not only between students of the same class, but from all over the world. Of course, to take advantage of the virtual lab,
new educational software has to be developed and the teaching practice of the majority of the lessons has to be altered. On the other hand, traditional learning material, like printed books, may be abandoned giving place to digital interactive books. Personalized education is another gain of the virtual lab. Teaching material adapted to the needs of every student, or team of students, can be installed at each VM. Finally, the usage of underpowered clients like Raspberry Pi 3 or later is not a drawback for the employment of our solution, but can be used to expand the access to the VMs from other spots in the school, rather than the school computer, like school libraries or classrooms, making pervasive computing a reality in the school in a very simple way.

V. FUTURE WORK

The next step of our project is the extensive use of the cloud based virtual lab from a larger number of schools, which will enable the evaluation of our solution under different hardware configurations and operating systems. Moreover, we will investigate the possibility to incorporate the virtual lab enriched with additional educational software in traditional and modern teaching methods. Finally, we plan to focus on the design of a training program addressed to the teachers on how to use a virtual lab, targeting specifically out-of-school usage.

VI. ACKNOWLEDGMENTS

We would like to acknowledge GRNET S.A, the operator of okeanos infrastructure, for granting us access to it. Special thanks go to Mr Yannis Mitsos, head of GRNET NOC, for his support to the completion of this project. Furthermore we would like to thank the teachers of the schools at which our experiments were conducted. Their support and relevant comments helped us to improve our infrastructure to meet as much as possible of their educational needs. Many thanks go to Mrs Eleni Robola, teacher of Informatics at a High School and to the students who participated in the experiments. Last but not least we thank Mr Yiannis Anastopoulos, supervisor of ICT support center of the 3rd Directorate of Secondary Education of Athens, for his valuable comments on the completion of this project.

REFERENCES


Dr. Vasilios S. Belesiotis was born in Athens, Greece in 1954. In 1978, he received a B.S. in Mathematics from the University of Patras, Greece. Moreover, in 1986 he received an M.Sc. in Computer Science and Operational Research from the University of Athens, Greece. Finally, since 2002, he holds a Ph.D. in Informatics from the University of Piraeus.

He served his military service in the Artillery as a second lieutenant. He worked a programmer in Soflex S.A. Athens, and since 1982, he works as a Secondary Education Informatics teacher for the Greek Ministry of National Education and Religious Affairs. His current position is a School Advisor for Informatics for the Regional Directorate of Primary and Secondary Education of Attica (since 2007). In addition, since 1997, he has been teaching in the Departments of Informatics of the University of Piraeus and Ionian University. He has published (author and/or co-author) over 40 conference and journal papers, and twelve books. His research interests include Didactics, Education of Informatics and Knowledge Representation.

Dr. Belesiotis has been a member of several technical committees and working groups on subjects related to Informatics and Education.

Constantinos P. Alexopoulos was born in Athens, Greece in 1968, received his B.Sc. in Mathematics from the University of Athens, Greece in 1989 and his M.Sc. in Atmospheric Physics and Meteorology from the Department of Physics of the University of Athens, Greece in 1991. Also received an MSc in Computer Science and Operational Research from The Department of Mathematics of the University of Athens, Greece in 1993.

He served his military service as sergeant of Artillery and during his military service worked as programmer for the Purchasing Department of the Greek Army. He worked as a programmer for Intrasoft S.A., a software solution developer and vendor company in Athens, as an advisor in software projects of the Greek Ministry of Education, as IT consultant of the A’ Directorate of Secondary Education of Athens and the Regional Directorate Of Primary and Secondary Education of Attica. Also Now he is working now as a teacher of Computer Science in Secondary education and his current position is Director of the 38th High School of Athens. He is author and co-author of many papers in scientific journals on computer science and he participated in many conferences as a speaker on the same subject. His research interests are on Cloud Computing, Image Processing and Computer Vision, Numerical Analysis and Network Security.

DOI: http://dx.doi.org/10.24018/ejers.2018.0.CIE.641