Providing Learning Computing Labs Using Hosting and Virtualization Technologies

doi:10.3991/ijet.v6iS2.1651

A. González, C. R. García and S. Candela
University of Las Palmas de Gran Canaria, Las Palmas de G.C., Spain

Abstract—**This paper presents a computing hosting system to provide virtual computing laboratories for learning activities. This system is based on hosting and virtualization technologies. All the components used in its development are free software tools. The computing lab model provided by the system is a more sustainable and scalable alternative than the traditional academic computing lab, and it requires lower costs of installation and operation.**

Keywords—computing hosting technology, infrastructure technologies for engineering education, virtualiaztion technology.

I. INTRODUCTION

In the area of information technology services, the use of hosting and virtualization technologies have improved business profitability, facilitating the technology innovation of companies and incorporating new business opportunities in this area. However, in general, these technologies have not been used in academic contexts to provide the computing platforms required by students and teachers in their teaching and learning activities. The main objective of the system architecture explained in this paper is to provide virtual teaching computing labs using computing hosting technology, following the idea proposed by Hardaway [1], but with the difference that the hosting and virtualization services are provided by the academic institution itself. In order to provide high availability, the architecture has a distributed structure, is easily scalable and allows the users (students and teachers) access to computing resources (virtual environments) in order to perform their computing tasks. It also allows academic authorities to provide computing resources at lower costs and in a more sustainable manner.

The proposed system architecture achieves a high level of availability and functionality, quality of service and sustainability, typical of computer system hosting, using several computing models such as clustering, virtualization and cloud computing. Specifically, the architecture provides virtual computing environments using the following elements: a cluster network to achieve high availability computing, balancer software used to ensure an equitable load distribution on the cluster virtualization nodes according to an energy efficiency criterion, a shared storage system offering high availability and a data-center remote manager to facilitate the system administration. An important aspect of the system architecture is that its implementation can be carried out at a very attractive cost because, on the one hand, it does not require high performance hardware resources and, on the other hand, it uses open software tools.

This paper is structured as follows: section II describes the architecture's computing models and computing technologies, section III presents the objectives and requirements, section IV describes the architecture system and includes an implementation sample while section V is devoted to conclusions and future improvements.

II. COMPUTING MODELS USED VIRTUALIZATION AND CLOUD COMPUTING

Virtualization is conceptually very similar to emulation. With emulation, a system pretends to be another system. With virtualization [2], a system aims to offer the user two or more different systems. Most modern operating systems contain a simplified system of virtualization. For example, in context of multiprogramming computing, each running process is able to act as if it were the only element running. The CPUs and memory are virtualized. If a process tries to consume all of the CPU, a modern operating system will preempt it and allow others their fair share. Similarly, a running process typically has its own virtual address space that the operating system maps to physical memory in order to give the process the impression that it is a space potentially unlimited. Hardware devices are also often virtualized by the operating system. A process can use a socket or equivalent to access a network device without having to worry about other applications. Since we already use some form of virtualization every day, we can see that it is useful. Another important property provided by the virtualization is the computing isolation. This property prevents a bug, or intentionally malicious behavior, in one application from affecting others. Unfortunately, applications are not the only things that contain bugs. Operating systems also do, and they often allow one application to compromise its usual level of isolation. Even in the absence of bugs, it is often a good idea to provide a greater degree of isolation than that offered by an operating system.

The basic motivation for virtualization, and which therefore motivates our proposal, is the same as that for multitasking operating systems; computers have more processing power than any one task needs. Many organizations are now finding that they have numerous servers all carrying out single tasks, or small clusters of related tasks. Virtualization allows a number of virtual servers to be consolidated into a single physical machine, without losing the security gained by having completely isolated environments. Several Web hosting companies are now making extensive use of virtualization, because it allows them to give each customer their own virtual machine without requiring a physical machine to take up rack space in the data center. In some cases, the situation is much worse. An organization may need to run two or more servers for a particular task, in case one fails, even though neither one comes close to full resource usage. Virtualization can help here, because it is relatively easy the virtual machines migration from one physical computer to another, making it easy to keep redundant virtual server images synchronized across physical machines.

A virtual machine offers certain advantages, such as cloning, at a very low cost. If you are uncertain about whether a patch will break a production system, you can clone that virtual machine, apply the patch, and see what breaks. This is considerably easier than trying to keep a production machine and a test machine in the same state. Another major advantage is migration. A virtual machine can be migrated to another host if the hardware begins to experience failures, or if an upgrade is scheduled. It can then be migrated back when the original machine is working again.

Power usage also makes virtualization attractive. An idle server still consumes power. Consolidating a number of servers into virtual machines running on a smaller number of hosts can reduce power costs considerably. A virtual machine is more portable than a physical one. You can save the state of a virtual machine onto a removable device, for example a USB external disk or something like a memory stick, and transport it even more easily than a laptop. When you want to use it, just plug it in and restore. Finally, a virtual machine provides a much greater degree of isolation than a process in an operating system. This makes it possible to create virtual appliances: virtual machines that just provide a single service to a network. A virtual appliance does not take up any space, and can be easily duplicated and run on more nodes if it is too heavily loaded.

Another computing model integrated in the architecture that we propose is cloud computing [3]. Everyone has an opinion on what this feature is. For Srinivasan and Getov cloud computing represents a fundamental shift in the delivery of information technology services that has permanently changed the computing landscape [4]. For example, it can be the ability to rent a server or a thousand servers and run a geophysical modeling application on the most powerful systems available anywhere. It can also be the ability to rent a virtual server, load software onto it, turn it on and off at will, or clone it ten times to meet a sudden workload demand. Or it can be storing and securing immense amounts of data that are only accessible to authorized applications and users. For example, it can be supported by a cloud provider that sets up a platform configured by free software tools, including the operative system, Apache, MySQL, Perl, Python and PHP with the ability to scale automatically in response to changing workloads. Cloud computing can be the ability to use applications on the net that store and protect data while providing a service. It can be using a storage cloud to hold application, business and personal data. And it can be the ability to use a handful of Web services to integrate photos, maps, and GPS information to create a mash up in customer Web browsers, for example the Google Street View service [5].

One of the main features of Cloud Computing is that there is no need to understand the infrastructure behind it. It is a 'cloud' where applications and services can easily grow (scale), running fast and almost never fail, without knowing the details of how this 'cloud' works. This type of service is paid for according to some measure of consumption, not the hardware used by itself but, for example, in electricity consumption or CPU/hour. Other important features included in cloud computing architectures are:

- Auto-repair: in case of failure, the last application backup automatically becomes the main copy and it is generated again.
- Scalability: the whole system or architecture is predictable and efficient. If one server handles 1,000 transactions, 2 managed servers handle 2,000 transactions.
- It is governed by a Service Level Agreement (SLA) that defines a number of policies, such as what the expected times of return are and whether it creates more peak instances.
- Virtualized: The applications are not dependent on the hardware, on which they run, and several applications can even run on one machine or an application can use several machines at once.
- Multi-purpose: The system is built in such a way that it allows different customers to share the infrastructure without worrying about it and without jeopardizing their security and privacy.

The cloud computing represents a drastic shift, not only in terms of the new economics it brings to bear, but also in terms of the new possibilities it enables [6], being the e-learning a field where these possibilities arise.

III. OBJECTIVES

The skills and competences related to information technology is a goal that has increasingly been incorporated into curricula for all education levels, including university studies in all fields (engineering, sciences and humanities). To this end, academic authorities should provide numerous information resources in different settings that affect sustainability in all aspects of these studies. Therefore models that can resolve this issue in an appropriate way are a topic of interest.

The main objective of the architecture developed is to provide distributed, easily scalable environments that supply the computing platforms required for teaching and learning activities that take place in the teaching labs of several university degrees. This will allow teachers and students themselves (educational users from now) access to virtual environments to perform these activities. Additionally, and in conjunction with this main goal, we will present the following features and requirements:

- Ability to have one or more virtual machines available for a specified period of time. For example, when a student beginning a learning activity requests a virtual machine, it validates that the student is enrolled in the course and then a machine is provided with all necessary resources to develop his/her computer practice activities during a stipulated period of time.
- Ability to have the virtual machine at anytime and anywhere. For example, a student will can access to his virtual machine from his home or from anywhere of the campus. Basically, the goal is to incorporate the mobility computing paradigm.
- Ability to communicate with the user to provide data access to a particular machine. When the machine an-

swers a request, it also provides all the data required to access the necessary data, for example, the machine's IP address and password.

- Publication of several operating systems to ensure a better fit to user needs where access to devices can be made via either Web or remote desktop (VNC). Users also have client software to access the requested resource. For example, users can access their virtual machines using their own personal computers from home or from anywhere with Internet access.
- Balancing of virtual machines through a scalable number of physical nodes by balancing the system performance with energy consumption. The academic authorities establish usage policies based on performance criteria that guide consumption and load balancing of computing nodes of the system.
- High availability. If a virtual machine should fail, the system mast is able to recover the service as quickly as possible. For example, if a student damages a machine, it can be recovered from the last image recorded on the system and run on any available computer node.
- Easily to manage for the administrator, allowing full centralized control over the machines of the distributed system. The management and administration of virtual machine users can be carried out remotely via a web tool that provides different screens to carry out all the administrative tasks required.

Finally, a requirement that has been taken into account to achieve these objectives is the use of free software tools so that system costs are attractive and maintenance is facilitated.

IV. DESCRIPTION OF THE SYSTEM ARCHITECTURE

The architecture developed is a full scalable distributed architecture that allows users to access their own virtual computing environments at anytime and anywhere using the Internet to perform practices and other tasks anytime and anywhere. With this objective in mind, we have configured a virtualization cluster with the addition of a shared storage system. The cluster is composed of a set of two or more nodes, named virtualization servers, which may be increased numerically at any time, and that serve to provide virtualization technology. On these virtualization servers, a virtual machine balancer that is responsible for providing the highest possible efficiency and availability is run. This is achieved by constantly monitoring the systems comprising the cluster and migrating virtual machines to one or another node if necessary, according to a fully configurable balancing algorithm. This gives rise to a system with a high level of availability. Virtual machines will be linked to a specific node so that if, for any reason, the virtual machine goes down, it will be restarted on the same virtualization server or another.

Looking for energy savings, the system has been configured to boot the nodes in succession as they are being filled, so the load balancer will ensure virtual machines are booted in the first cluster node and when a configurable percentage of memory load has been exceeded, this can also give rise to migration in order to balance that percentage if there are nodes with available resources. Thus, for a possible extension of this project, it would boot cluster nodes as necessary.

To facilitate the management function of the system administrators, the architecture includes the use of Web services for managing and monitoring servers and for deployment and manipulation of storage systems available in the distributed system. Using these services, the manager may carry out the following actions, among others: create, delete, start, stop and migrate virtual machines, manage logical volumes to install several virtual machines, create some virtual machine templates, deploy virtual machines, manage user accounts and monitor the services of any machine in the distributed system.

The administrator may at any time migrate to virtual machines for maintenance purposes from one node to another via the Web interface, without causing a significant performance drop as seen by users. In performing these migrations, the system uses the cluster command to enable the machine to be available on a continuous basis. Similarly, each machine is set to be launched by a command load balancer that uses cluster architecture.

The component diagram that models the physical layout or topology of the architecture is shown in fig. 1. This diagram describes the hardware used; the hardware components installed and specifying the physical connections between hardware components and relationships. The main element in this diagram is the node. Each node represents a physical resource of the architecture, that is: computers, servers, external devices, and so on. These nodes are interconnected by lines defining a network structure. The cluster is identified in this diagram as a load balancer and is composed entirely of nodes that are virtualization servers. Therefore, the cluster is only composed of virtualization servers that keep user requests in physical memory, as seen in the diagram of components.

This model also illustrates the layout of the nodes in the system and its interaction with neighbors through the components. The user accesses the application through the local browser to interact with the Web portal, located in one of the nodes, and can access the various components in a transparent way. Data is on a server located in our database and this is where the Web application stores and retrieves the information necessary for the proper determination of the user's needs. Among the information stored in the database is the identification of system users and the various settings established by the users.

All relevant action is stored in a shared log system and accessible from all nodes in the system; it is essential to use a component that allows for the assembly of shared devices in each of the nodes of virtualization. This component is also necessary for the continued visibility of significant data in the host operating system in order to balance the execution of virtual machines as efficiently as possible.

It is on the virtualization servers where the requests of users are really running and where the system spends a great deal of time trying to keep the service operating with an appropriate level of load and high availability of virtual machines. This is achieved through the intervention of the components in each LBmonitor virtualization node synchronized with the sole purpose of enhancing the performance of this distributed system. Furthermore, this component is on a subsystem LBmonitor cluster which is what really affords high availability to the system and allows for it to be managed centrally.

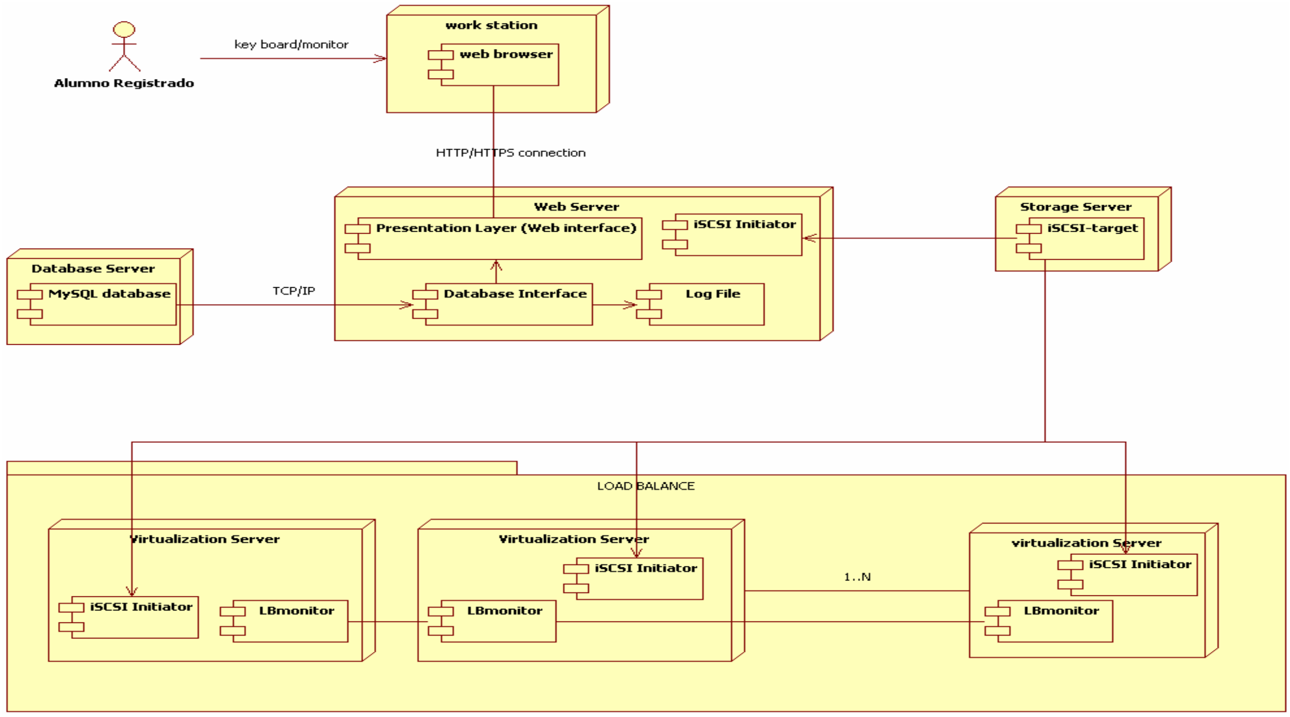


Figure 1. Component diagram of the architecture

A. Implementing the Architecture

This section explains how the design has been constructed; explaining how a fully scalable distributed system has been developed that enables users to access their own virtual machines for their practices and other tasks. To build this architecture we have used only open source resources that are easy to implement and incur no extra cost. This point describes the main used software components.

The virtualization technology used is Xen Hypervisor [7]. This is a powerful open source that is standard for the virtualization industry and offers enormous capacity. It is efficient and offers secure virtualization features on a wide range of processors. It also supports a wide range of guest operating system such as Linux, Windows, Solaris and various versions of BSD operating systems. Furthermore, we used CentOS 5.4, given its usability and high performance. Virtualization Xen virtual machine in hardware (HVM) provides support for virtualization of unmodified guest operating systems without the performance loss inherent in the use of full virtualization.

With systems containing CPUs with virtualization support, a HVM approach provides higher levels of performance compared to full virtualization. If the processor does not support Xen virtualization the paravirtualization mode can be used although this will not allow for the running of unmodified operating systems and Microsoft Windows as guest operative systems on XEN. To set up the server virtualization installing the CentOS Linux system and adding the Xen virtualization technology; to this end the kernel of the systems was modified with extreme care. These servers, now server virtualization, must have hardware support for virtualization into their processors. This requires a small amount of disk space but the maximum RAM possible because this role is focused on maintaining virtual machines on that memory. The CentOS support was used for building a cluster of virtualization. This cluster is composed of two or more virtualization servers interconnected by a switch.

For high availability, clustering software needs to be obtained by using several tools. Specifically, we used the Red Hat cluster suit [8]. The cluster manager (CMAN) provided by Red Hat consists of two parts: the connection manager and the service manager. The first handles cluster members, manages and reports the transitions while the service manager manages group services that require some type of administration. CMAN is the foundation upon which Distributed Lock Manager (DLM), Global File System (GFS), Clustering Logical Volume Manager (CLVM) and FENCE are dependent and is generally available for any user space program. Rgmanager (Resource Group manager), running under CMAN (using DLM) to manage and provide failover capabilities for all

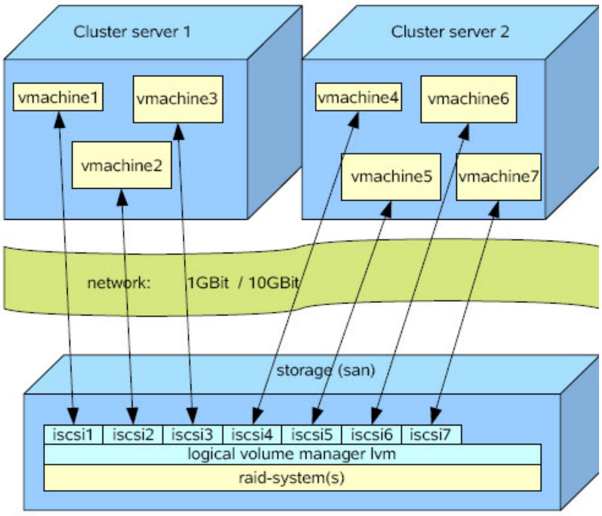


Figure 2. Architecture implementation

resources called services, resource groups or trees in a cluster resource. Conga provides an integrated set of software components, providing a centralized setting and management of clusters and storage systems. It has many features, among which we would highlight the following: providing a web interface to handle the storage system, cluster groups and services; to facilitate the deployment of software among several nodes of the cluster; to allow easy integration with clusters and accurate control of users' privileges. All these actions are reflected in logs, which provide information as to the state of the cluster.

To achieve a balance between system performance and resource consumption, we used balancing software called LBVM [10] that allows automatic migration of virtual machines. This allows virtual machines to be shared across physical servers on a predefined cluster and even to define balancing algorithms for the system to migrate a virtual machine in order to find better conditions in another physical environment based on the state of processor load, on the use of RAM memory or any other resource defined. LBVM consists of a set of Perl scripts that are installed on each cluster node virtualization; to adapt the operation of LBVM to the characteristics of our architecture we have made some changes in the Perl scripts. This virtual machine balancer, which is responsible for providing the highest possible efficiency and availability, runs on these virtualization servers. This is achieved by constantly monitoring the systems that form the cluster and migrating virtual machines to one or another location where required by a fully configurable algorithm. The virtual machines are added to the cluster after their creation process and it fills a particular node to a preset threshold RAM capacity. In our implementation, this threshold is fixed in relation to the memory needed for the host node from the beginning and was set at 1 Gb. While computers are turned on, the system continuously monitors and evaluates existing benefits using the balancing algorithm to determine whether a particular machine should be migrated.

To perform hot migrations with no downtime, the images of virtual machines must be located in a shared storage space and be accessible from all nodes in the system. This could be done in several ways, including using a SAN (storage area network) or with DRBD, if only two physical machines are used. In our case, we used iSCSI [9] software through open source iSCSI-Target, an application that works very well in business environments. Also, This shared area is used by the LBVM because the it needs to have continuous information as to available resources at each node. This information is provided by a script that runs periodically at each node and records the reading on files stored in the shared area. Thus, every node is aware of the status of their neighbors. In addition, in order to locate here the record of the status of system resources that form the cluster, balancer configuration files are also located, as well as images and configuration files for each virtual machine. Finally, it must be visible from every node. This is because every possible destination node needs to be aware of the existence and full settings of all the machines in order to migrate. Thus, this system offers a high degree of availability.

We shall now consider the implementation of the user interface, in order to achieve a high level of usability for users, both technical staff (system operators and administrators) and academic users (students and teachers), the software application openQRM 4.6, [11] providing Web interface has been installed that allows the deployment of requests for virtual machines. This application represents the next generation and is an open source platform for the Data-center management. Its architecture is completely composed of plugins designed for rapid deployment and dynamic monitoring applications enabling flexible, high availability, Cloud Computing that supports different types of virtualization.

The role and relevance of use interface is illustrated in the following example: a user contacts the administrator to create a virtual machine. Then the administrator has to create the virtual machine manually by system commands through the console. Having created the petition, the system should contact the student. It would therefore be very useful to have an interface through which to perform the tasks of administration and to enable communication between the user and administrator. OpenQRM has been used for this purpose. This tool is an open source platform for centralized data control. Its architecture is completely composed of plugins designed for rapid deployment and dynamic device, flexible monitoring, High Availability and Cloud Computing and it supports different types of virtualization, including XEN. Through this tool, see fig. 3, the system administrator via the Web may, among other actions:

- Create, delete, start, stop and migrate virtual machines.
- Manage logical volumes to install several virtual machines.
- Create some virtual machines templates.
- Clone template virtual machines via snapshot (LVM).
- Perform deployment of virtual machine via Net File System (NFS) or iSCSI.
- Manage user accounts.
- Monitor the services of any machine in the distributed system.

We have had to make some modifications to openQRM because the virtualization system developed is not installed by default and, at the same time, this software is not configured to work on the cluster developed. Specifically, we have taken the following action:

- Changed management tools for virtual machines that are made on the balancer and cluster configured, such as starting, deleting etc.
- Performed migration on the cluster, because otherwise migration would not be registered by the balance and would give rise to an inconsistent state.

As a result, the administrator may at any time migrate virtual machines from one node to another via the Web interface, without causing a performance drop that users will perceive as significant. As mentioned, when performing this migration, the system uses the cluster command to enable the machine to remain available. Similarly, each machine created by means of the balancer command system uses the cluster command to create machines. This was done by modifying the scripts to run the web interface.

Any machine created will be recorded in the cluster configuration file /etc/cluster/cluster.conf, which defines parameters for management such as: domain as recovery in case of failure, type of migration to use path configuration file location etc. Each time this file is modified, the version of it is upgraded and spread to other nodes in the cluster which are then aware of the new machine created. This is carried out by providing the tool commands CMAN.

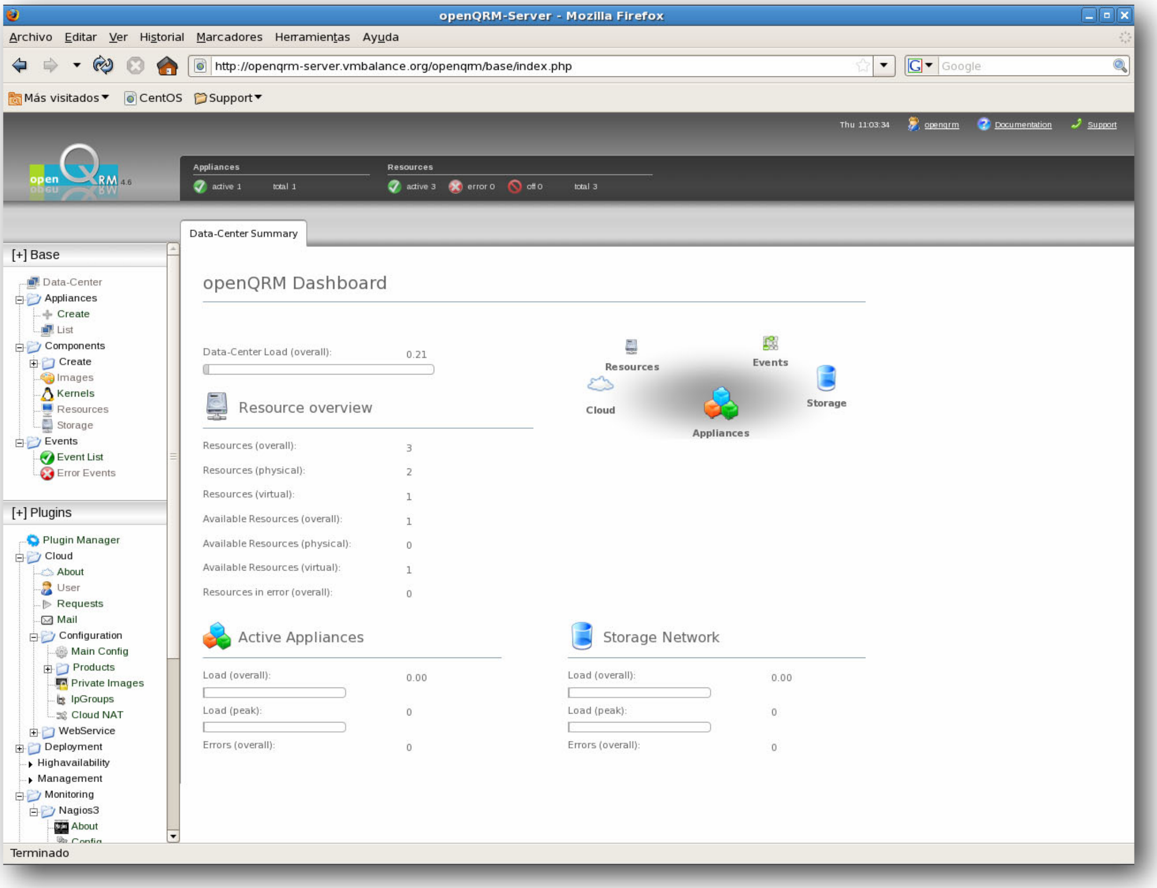
A domain is a node belonging to the cluster and in which a particular machine can boot in case of unexpected failure. Once the user making the request runs the command *lbm-c virtual_machine* the above-mentioned settings are made registering the new device in the balancer and the cluster is responsible for starting the machine in the corresponding node using the command *clusvcadm*. Registration in the balancer is added in your configuration file in order to balance the new machine. This will then be aware of the existence of a new virtual machine balancing system to be migrated, if successful in the relevant checks.

Thanks to the Web interface, users can log into the system for virtual cloud machine requests and access via their own portal from which they can view all information regarding their machine managed by openQRM. In a production system, the user can select several pre-configured options for operating systems that will equip their virtual machine. For example, CentOS systems have been selected as a machine for users to distribute faculty computer labs. This distribution has been assembled and pre-configured on the iSCSI shared storage area to serve as a template for any such request. What is achieved by creating this template is that each time a user requests that copy be made, an exact copy of it with the option of snapshot of LVM is generated. Any variation to the new copy assigned to the user will only be recorded on the copy, keeping the original template intact.

Once the request has been made, the virtual machine starts with the minimal components, then reboots and adds the iSCSI device copied and assigned to the user. In the implementation done, the creation of a new virtual machine requested by the user, which involves copying the root file system; it consumes about 5 minutes. Before this time elapses, the user will receive an email sent to his inbox telling him that his request will soon be implemented, along with the IP of his machine and the root password.

To facilitate access to private network hosts in a distributed system from any user machine, such as a desktop or a laptop using an SSH connection by providing a password, a small client application using language C has been developed. The resources that need to be installed on the user machine to connect using this tool are: Java, if you want to access the Web via virtual machine; Real Vnc Viewer, to interact with desktop applications across the network; the server must have VNC Server installed to provide visible access port and 0.60_q1.129 Quest Putty to create a SSH tunnel to the private IP machines located in the distributed system.

Finally, to keep abreast of the status of remote machines at all times, we used Nagios 3 as monitoring system. With this software with a GPL license, we always know the status of any service to any pre-configured machine.



Appliances: active 1 total 1
Resources: active 3 error 0 off 0 total 3

openQRM Dashboard

Data-Center Load (overall): 0.21

Resource overview

Resources (overall): 3
Resources (physical): 2
Resources (virtual): 1
Available Resources (overall): 1
Available Resources (physical): 0
Available Resources (virtual): 1
Resources in error (overall): 0

Active Appliances

Load (overall): 0.00
Load (peak): 0
Errors (overall): 0

Storage Network

Load (overall): 0.00
Load (peak): 0
Errors (overall): 0

Figure 3. Initial administration screen.

V. THE CASE OF A TEACHING LAB

To illustrate the above, we present here the implementation details of the architecture by focusing on the hardware elements used and the type of educational activity supported by this implementation. The teaching activities supported with this infrastructure are the practical activities of topics related to the subject of operating systems in the Computer Science degree and master at the University of Las Palmas de Gran Canaria. Specifically, among other subjects, we address here:

- Basic Principles about Operating Systems in which about 150 students use resources to learn how to use the resources that an operating system provides (system programs and system calls). From a point of view of specific requirements for the practical activities of this subject, the computing platforms must have the tools for editing, compiling and debugging "C" programs. Form the point of view of the technical support, basically installations and configuration tasks are required before the start of the practical activities.
- Operating Systems Administration in which about 150 students learn the basics of operating system administration (installation, managing users and resources, refining the system and accounting system). From a point of view of specific requirements for the practical activities of this subject, the computing platform must have different packets of software services and services and the tools to develop scripts in different languages (Korn Shell, Perl, PHP, etc.) Because of the students must execute all the tasks related to the life cycle of a operating system and services (installations, configuration and tuning) no technical support is required for these activities, but because of the risk of system malfunction due to mistakes made by students in their operations with the systems, supports for system backup and system recovering must be available.
- System Programming in which about 50 students learn to develop new operating system components (systems software and kernel modules). From a point of view of specific requirements for the practical activities of this subject, the computing platform must have installed the development system of the operating system, basically configured by the "C" language compiler, debugger and the source code of the operating system (kernel and services). Because of the students must execute all the tasks related to the life cycle of the development platform (operating system and development system) that are installations, configuration and tuning, no technical support is required for these activities, but because of the risk of system malfunction due to mistakes made by students in their system developments, supports for system backup and system recovering must be available.

Following the traditional scheme of computer lab established for the operational activities of the subjects mentioned in the School of Computer Engineering, the Department of Information Systems at the University of Las Palmas de Gran Canaria provides a joint lab with 30 jobs (PCs) with the ability to run the software tools required for such activities. As the requirements of different practical activities, installations and configurations of different tasks are sometimes incompatible; the tasks have to be organized in groups that are assigned to subjects with compatible requirements facilitating the lab support and maintenance activities. Due to the difference between the number of jobs and the number of students, they are divided into different groups who come to the lab at different times. Another aspect that affects the configuration of the task is the given calendar of subjects, some of which are taught in the first semester and others in the second, so it is necessary to plan the installation and configuration tasks in terms of this calendar.

The alternative lab model proposed system architecture, provides virtualization cluster where practical activities are performed. Fig. 4 represents this alternative lab model. This virtualization cluster has been built with two servers; two dual-core processor intel Xeon inside 5120 to 1.86GHz and support VMX technology with 8 Gb random memory access (RAM). Another with 2 processors with similar features to the previous, 4-core intel Xeon inside E5345 to 2.33 GHz supports VMX too and with 8 Gigabytes of RAM. To support a variety of services such as email, Web service and the management of virtual machine by openQRM, a personal computer with processor Intel Core 2 Duo CPU E7400 to 2.80 GHz to 4 Gigabytes of RAM has been used. To provide the shared area with iSCSI-Target technology we used a Pentium D. Finally, a switch with 24 ports Gigabit Ethernet interconnects the physical machines involved.

For each subject there is a template or standard virtual machine that contains all the hardware resources, see fig. 5, and software tools required for practical activities. Currently, the used system templates are based on Linux distributions: CentOS 5.5, for the subject of Operating System Administration, and Fedora 14 for the subjects of Basic Principles about Operating Systems and System Programming. Each time a student is signed up in the subject, a virtual machine cloning virtual machine template is created.

VI. CONCLUSIONS AND FUTURE WORK

In conclusion, we can say that a computing hosting system to provide virtual learning computing laboratories has been developed using a set of free software tools (CentOS distribution, XEN, iSCSI and OpenQRM). Each virtual computing lab has an associated Web portal, offering the students a set of virtual computing components, and teachers, a set of tools to supervise the students' tasks and technical staff, a set of maintenance and maintenance

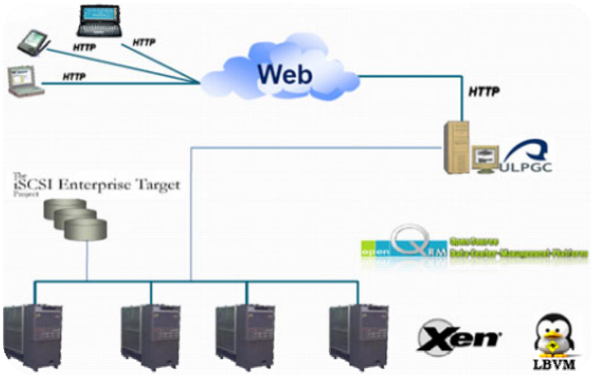


Figure 4. Overall vision of operating system virtualized lab

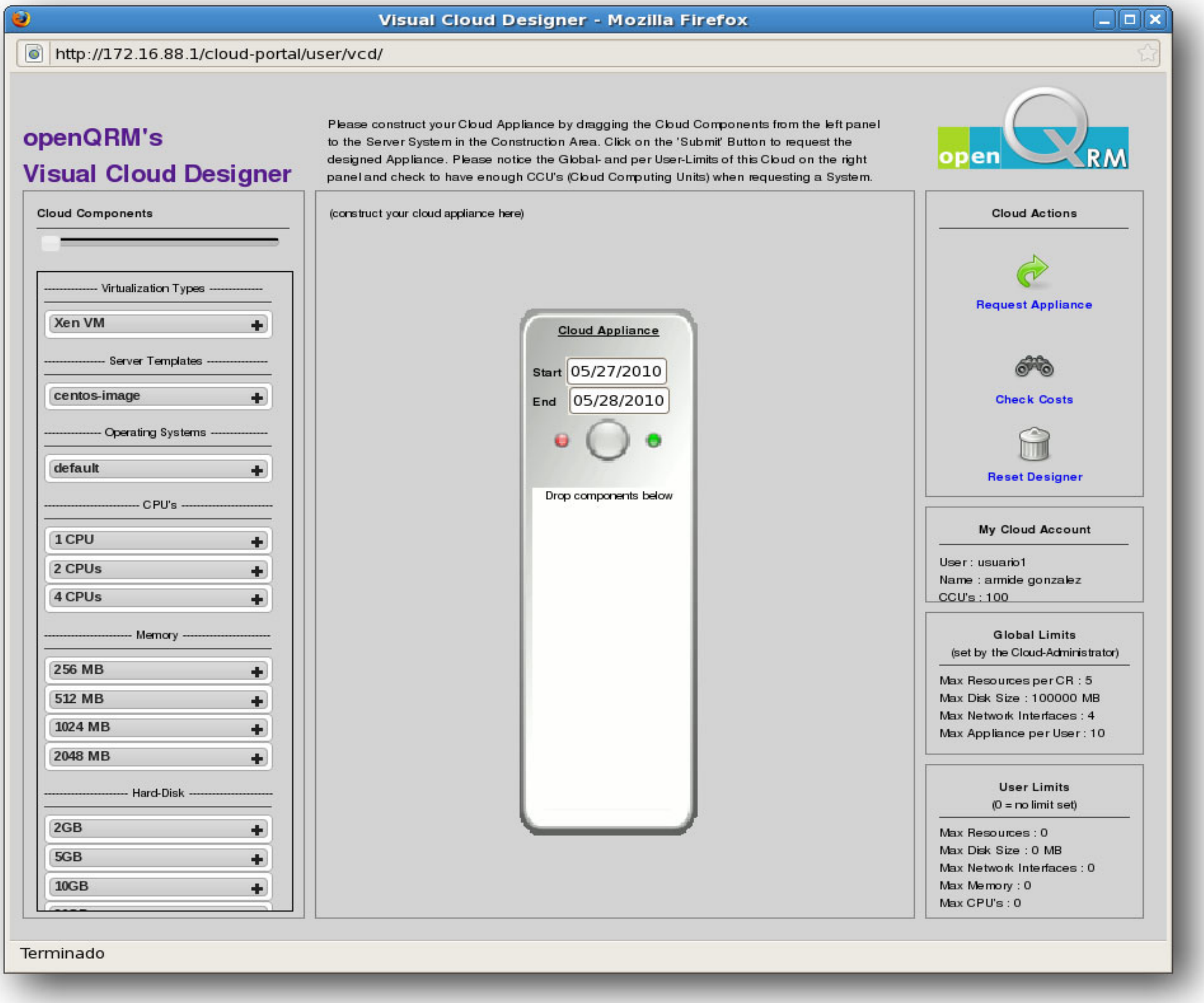
tools. This virtual lab model is a more sustainable and scalable alternative than the traditional academic computing lab, as it provides the computing resources required by the learning activities of the students with lower installation and operation costs. The hardware resources required for the hosting of the virtual computing lab are provided by a distributed system configured by a set of computers with no special requirements; the only special requirement is that they support the hardware virtualization technology. Additionally, the following functionalities have been achieved:

- Ability to have the virtual machine at anytime and anywhere.
- Publication of several operating systems to ensure a better fit to user needs where access to devices can be made via either Web or remote desktop.
- Balancing of virtual machines through a scalable number of physical nodes by balancing the system performance with energy consumption. The academic authorities establish usage policies based on performance criteria that guide consumption and load balancing of computing nodes of the system.
- High availability of computing resources (CPU and memory) to execute the computing learning tasks.
- Fault tolerance. If a virtual machine should fail, the system mast is able to recover the service as quickly as possible.
- Easily to manage for the administrator, allowing full centralized control over the machines of the distributed system.

This project has a wide range of future possibilities. In the immediate future, possible use in the School of Computer Engineering of the University of Las Palmas de Gran Canaria to observe the behavior of the system in a production environment. On the other hand, some improvements are needed, for example the balancer algorithm of virtual machine to boot nodes when needed, resulting in energy savings and theVirtual machine performance by balancing algorithms.

ACKNOWLEDGMENTS

The authors of this paper wish to express their sincere thanks to the Free Software Office, the Department of Computers and Systems and the School of Computer Engineering of the University of Las Palmas de Gran Canaria for their support in the carrying out of this work. They are also grateful to Heather Adams, ULPGC, for revising the English translation of the manuscript.


Visual Cloud Designer - Mozilla Firefox
http://172.16.88.1/cloud-portal/user/vcd/
openQRM's
Visual Cloud Designer
open QRM
Cloud Components
------------- Virtualization Types -------------
Xen VM
-------------- Server Templates --------------
centos-image
------------- Operating Systems -------------
default
-------------------- CPU's --------------------
1 CPU
2 CPUs
4 CPUs
-------------------- Memory --------------------
256 MB
512 MB
1024 MB
2048 MB
------------------- Hard-Disk -------------------
2GB
5GB
10GB
(construct your cloud appliance here)
Cloud Appliance
Drop components below
Cloud Actions
Request Appliance
Check Costs
Reset Designer
Terminado

Please construct your Cloud Appliance by dragging the Cloud Components from the left panel to the Server System in the Construction Area. Click on the 'Submit' Button to request the designed Appliance. Please notice the Global- and per User-Limits of this Cloud on the right panel and check to have enough CCU's (Cloud Computing Units) when requesting a System.

Start 05/27/2010
End 05/28/2010

My Cloud Account

User : usuario1
Name : armide gonzalez
CCU's : 100

Global Limits
(set by the Cloud-Administrator)

Max Resources per CR : 5
Max Disk Size : 100000 MB
Max Network Interfaces : 4
Max Appliance per User : 10

User Limits
(0 = no limit set)

Max Resources : 0
Max Disk Size : 0 MB
Max Network Interfaces : 0
Max Memory : 0
Max CPU's : 0

Figure 5. Screen for creating a virtual machine template.

REFERENCES

[1] D. Hatrdaway, M.J. Hogan and R.G. Mathieu, "Outsourcing the University Computer Lab", Computer, vol. 38 , n. 9, p.p. 100-102, 2005. doi:10.1109/MC.2005.308

[2] J.E. Smith and R. Nair, "The Architecture of Virtual Machine", Computer, vol. 38 , n. 5, p.p. 32-38, 2005. doi:10.1109/MC.2005.173

[3] M. Armbrust et al., "A view of Cloud Computing", Comm. ACM, p.p. 50-58, 2010.

[4] S. Srinivasan and V. Getov, "Navigating the Cloud Computing Landscape- Technologies, Sevices and Adopters", Computer, vol. 44, n. 3, p.p. 22-23, 2011. doi:10.1109/MC.2011.91

[5] D. Angelov, et al., "Google Street View: Capturing the World at Street Level", Computer, vol. 43, n. 6, p.p 32-38, 2010.

[6] Y. A. Khalidi, "Building a Cloud Computing Platform for New Possibilities", Compter, vol. 44, n.3, p.p. 29-34, 2011. doi:10.1109/MC.2011.63

[7] D. Chisnall. "The Definitive Guide to the Xen Hypervisor", Prentice Hall , 2007.

[8] "Red Hat Cluster Suite Overview", http://www.centos.org/docs/4/4.5/SAC_Cluster_Suite_Overview/index.html, 2007.

[9] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka and E. Zeidner," Internet Small Computer Systems Interface (iSCSI)", The Internet Society, http://www.ietf.org/rfc/rfc3720.txt, , 2003.

[10] R. Dworschak, S. Huber, A. Leitner and J. Pöttinger, "Load balancing of virtual machine", http://lbvm.sourceforge.net/docu/, 2008.

[11] T. Halinka,, "The openQRM user´s guide", http://www.openqrm.com/, 2008.

AUTHORS

A. González (armidegonzalez@gmail.com) is with the Computing Engineering School of the University of Las Palmas de Gran Canaria. Edificio de Informática y Matemáticas, Campus de Tafira, Las Palmas, CP 35017 Spain.

C.R. García (rgarcia@dis.ulpgc.es) is with the Computer Science Department of the University of Las Palmas de Gran Canaria, Edificio de Informática y Matemáticas, Campus de Tafira, Las Palmas, CP 35017 Spain.

S. Candela (scandela@dis.ulpgc.es) is with the Computer Science Department of the University of Las Palmas de Gran Canaria. Edificio de Informática y Matemáticas, Campus de Tafira, Las Palmas, CP 35017 Spain.

This article is an extended version of a paper presented at the IEEE EDUCON20211 Conference, held from April 4th-6th, 2011, in Amman, Jordan. Received April 29th, 2011. Published as resubmitted by the authors May 2nd, 2011.