



UNIVERSIDADE DA BEIRA INTERIOR
Faculdade de Engenharia
Departamento de Informática

A Web-based Solution for Virtual Machine Instances Migration Across Type-2 Hypervisors

Didier Martins Ribeiro

Submitted to the University of Beira Interior in candidature for the
Degree of Master of Science in **Informatics Engineering**
(2nd cycle of study)

Supervised by Prof. Dr. Joel José Puga Coelho Rodrigues

Department of Informatics
University of Beira Interior
Covilhã, Portugal
<http://www.di.ubi.pt>

Covilhã, Outubro de 2014

Acknowledgements

This dissertation would not be possible without the collaboration of some persons and entities, which directly and indirectly helped to make all this possible.

My sincere thanks to professor Joel Rodrigues, my mentor and coordinator, for the opportunity to work in this project and also for his advice and support in all challenges and steps.

I am most grateful to the University of Beira Interior and the Next Generation Networks and Applications Group (NetGNA), Instituto de Telecomunicações, Covilhã delegation, for all the support, excellent environment and cooperation inside the group.

Thanks to all partners involved in the project, to Tiago Simões and to all other friends and family not mentioned here.

Last but not least, I am most grateful to my family, especially for my father Manuel Apolinário Ribeiro who died suddenly in July and my mother Maria Adelina Martins Reis for all their support, love and patience during the long hours of work, where nothing would be possible without them.

Abstract

Cloud computing has improved computing efficiency by reducing the cost to the users. A current datacenter consists of tens to hundreds of thousands of servers and contains hundreds of thousands of switches connected hierarchically. Sharing processing resources through services like "software as service" (SaaS), users can amortize the cost of hardware and software. To facilitate upgrades and maintenance of systems, virtual machines (VMs) are often used to provide services, and their migrations result in better use of resources.

The cloud, supported by virtualization is emerging as an important paradigm of "oriented service". The administration of systems is critical to provide availability and performance in data systems, providing automated the required real-time capacity to meet service requests. But virtualization does not reduce the complexity of a system. In fact, the execution of multiple virtual machines (VMs) on top of a physical infrastructure can increase overall system complexity and present new challenges in his administration.

Virtualization of resources is a key component of "cloud computing" to provide computing and storage services being ubiquitous in today's "data center". Support for servers by building cluster of virtual machines is universally adopted to maximize the utilization of hardware resources.

Virtualization has become a key technology implemented by a growing number of organizations related to Information Technology (IT) worldwide. Virtualization of systems has rapidly gained popularity because of its potential to reduce the costs of TI's. Allowing IT managers to increase the use of existing physical resources and even reduce the number of deployed systems. This consolidation helps reduce hardware requirements management, reducing the need for power and cooling, and thus reducing the costs of TI's in general.

Additionally, the deployment of virtualization solutions typically means increased management tools to the existing environment. Access to software and data anywhere, anytime, on any device and in any connectivity, has long been a crucial issue for researchers and architects systems. The amount of data processed increases each year, both in large-scale systems and in smaller environments. Likewise, the computation is being performed for processing the data, and the communication is made to distribute the data. This phenomenon is associated with a steady increase in computing power, storage and communication resources available, although with different characteristics.

The impact of the current growth in the use of virtualization tools became more popular the use of virtual machines. The use of virtual laboratories-testing is becoming more popular in the practice of QA testing. This approach allows the "testers" to test different applications without relying on permanent configuration of a system.

Using virtual machines, QA tests can simulate different computers with different operating systems through a single physical computer or create a full virtual laboratory with multiple virtual machines configured differently. These virtual "computers" operate independently of each other and we can launch two or more virtual platforms simultaneously on one computer, saving the cost of having to buy more hardware just to run quality control tests. Applications running in a virtual machine behave as if they were running on its own physical system. This can also be useful to test web applications, because it can simultaneously test web applications across browsers which work independently of each other in different virtual machines, again, without the cost of buying more hardware for testing needs.

Testing applications with virtual machines may have different utilities. This can be used for distributed client-server applications, functional testing, regression test, etc. But no matter what kind of QA testing we use, it will

be more effective if automated and any kind of testing in virtual laboratories can be easily automated.

A computer system is a dynamic system and configurations of operating systems continually change. Install or update software drivers and hardware happens frequently and installing different versions of an application affects the internal structure of the system and may influence the test results. While buying multiple computers to support multiple platforms is an option for some, it is often too expensive for most testing labs.

Fortunately, virtual machines are a solution to these problems and much more cost effective. Once we have created and configured a virtual machine or a lab full of virtual machines, we can have the configuration of a stable system, which is very important when testing applications.

However, we may need a more powerful computer to run multiple virtual machines on a single computer at the same time, but that's often cheaper than buying three physical computers.

With a virtual laboratory in a computer, we can perform tests with distributed client-server applications without the need for multiple computers.

In this dissertation, based on the characteristics presented above, is presented VirtualMigra. The VirtualMigra Platform is a tool that allows the migration of virtual machines regardless of their manufacturer among different users in a LAN. The use of the Oracle Virtualbox and VMware Workstation APIs allows a comfortable and intuitive level of abstraction for the users.

Were conducted exhaustive experiments to test the platform and these were successfully performed in a real environment, thus being ready for real exploration platform.

Resumo

A computação em nuvem tem melhorado a eficiência de computação reduzindo o custo para os utilizadores. Um centro de dados atual é constituído por dezenas a centenas de milhares de servidores e contém centenas de milhares de “switches” hierarquicamente conectados. Partilhando os recursos de processamento através de serviços como “Software as Service” (SaaS), os utilizadores podem amortizar o custo do hardware e do software. Para facilitar atualizações e manutenções de sistemas, as máquinas virtuais (VMs) são frequentemente utilizadas para prestação de serviços, e suas migrações resultam numa melhor utilização dos recursos.

A computação em nuvem, apoiada pela virtualização está emergindo como um importante paradigma de serviço orientado. A administração de sistemas é fundamental para oferecer disponibilidade e desempenho em sistemas de dados, fornecendo de forma automatizada a capacidade necessária em tempo real para atender pedidos de serviços. Mas a virtualização não reduz a complexidade de um sistema. Na verdade, a execução de várias máquinas virtuais (VMs) no topo de uma infra-estrutura física pode aumentar a complexidade geral do sistema e colocar novos desafios na sua administração.

A virtualização de recursos é a componente chave da computação em nuvem para fornecer serviços de computação e armazenamento estando omnipresente nos centros de dados atuais. O apoio à servidores através da construção de cluster de máquinas virtuais é universalmente adotada para maximizar a utilização dos recursos do hardware.

A virtualização tornou-se uma tecnologia chave implementada por um número crescente de organizações relacionadas com as Tecnologias da

Informação (TI) em todo o mundo. A virtualização de sistemas rapidamente ganhou popularidade por causa do seu potencial em reduzir os custos das TI's. Permitindo aos gestores de TI aumentarem a utilização dos recursos físicos existentes e até mesmo reduzir o número de sistemas implementados. Esta consolidação ajuda reduzir os requisitos na gestão de hardware, reduzindo as necessidades de energia e refrigeração, e, assim, reduzindo os custos das TI's em geral.

Além disso, a implantação de soluções de virtualização normalmente significa um aumento de ferramentas de gestão para o ambiente existente. O acesso ao software e dados em qualquer lugar, a qualquer hora, em qualquer dispositivo e com qualquer conectividade, já há muito tempo é um tema crucial para investigadores e arquitectos de sistemas. A quantidade de dados processados aumenta em cada ano, tanto em sistemas de larga escala como em ambientes de menor dimensão. Da mesma forma, a computação mais está a ser executada para processar os dados, e mais comunicação é utilizada para distribuir os dados. Este fenómeno é associado com um aumento constante da capacidade de computação, armazenamento e recursos de comunicação disponíveis, embora com características diferentes.

O impacto do crescimento actual do uso de ferramentas de virtualização tornou mais popular a utilização de máquinas virtuais. O uso de laboratórios de testes virtuais está-se a tornar mais popular na prática de testes de QA. Esta abordagem permite que os "testers" testem aplicações diferentes sem se basearem na configuração permanente de um sistema.

Usando máquinas virtuais em testes de QA, podemos simular diferentes computadores com diferentes sistemas operativos através de único computador físico ou criar um laboratório virtual inteiro, com várias máquinas virtuais configuradas de forma diferente. Estes "computadores" virtuais funcionam de forma independente uns dos outros e podemos lançar duas ou mais plataformas virtuais simultaneamente no computador, economizando o custo de ter que comprar mais hardware apenas para

executar testes de controlo de qualidade. As aplicações em execução numa máquina virtual comportam-se como se estivessem em execução no seu próprio sistema físico. Também pode ser útil testar aplicações web, pois pode-se testar simultaneamente aplicações web em vários navegadores que funcionam independentemente uns dos outros em diferentes máquinas virtuais, mais uma vez, sem o custo de comprar mais hardware para necessidades de testing.

Testar aplicações com máquinas virtuais pode ter diferentes utilidades. Pode-se utilizar para testes distribuídos de aplicações cliente-servidor, testes funcionais, testes de regressão, etc. Mas, não importa que tipo de testes de QA que usamos, também será mais eficaz se for automatizado e qualquer tipo de testes em laboratórios virtuais podem ser facilmente automatizadas.

Um sistema de informático é um sistema dinâmico e as configurações dos sistemas operativos mudam continuamente. Instalação ou atualização de drivers de software e hardware acontece com frequência e instalação de diferentes versões de um aplicativo afeta a estrutura interna do sistema, podendo influenciar os resultados dos testes. Enquanto que a compra de vários computadores para suportar várias plataformas é uma opção para alguns, muitas vezes é muito caro para a maioria dos laboratórios de testes.

Felizmente, as máquinas virtuais são uma solução para estes problemas e a um custo muito mais eficaz. Depois de ter criado e configurado uma máquina virtual ou um laboratório cheio de máquinas virtuais, pode-se ter a configuração de um sistema estável, o que é muito importante ao testar aplicações.

No entanto, pode ser preciso um computador mais potente para executar várias máquinas virtuais num único computador ao mesmo tempo, mas isso é muitas vezes mais barato do que comprar três ou mais computadores físicos.

Com um laboratório virtual num computador, podemos executar testes distribuídos com aplicações cliente-servidor sem a necessidade de vários computadores.

Nesta dissertação, com base nas características apresentadas anteriormente, é apresentada VirtualMigra. A plataforma VirtualMigra é uma ferramenta que permite fazer a migração de máquinas virtuais independentemente do seu fabricante entre diferentes utilizadores de uma LAN. O uso das APIs do Oracle Virtualbox e do VMware Workstation permitem um confortável e intuitivo nível de abstracção para os utilizadores. Realizaram-se experiências exaustivas para testar a plataforma e estas foram realizadas com sucesso num ambiente real, estando assim a plataforma pronta para exploração real.

Keywords

Cloud Technology Virtualization Systems, Hypervisors, Virtual Machines, Cold Migration, SOAP Web Services, Oracle VirtualBox, VMware Workstation, Open Virtualization Format.

Contents

Acknowledgements.....	i
Abstract	iii
Resumo	vii
Keywords.....	xiii
Contents	xv
List of Figures	xix
Acronyms.....	xxiii
1. Introduction	1
1.1. Focus.....	1
1.2. Objectives.....	3
1.3. Main Contributions	3
1.4. Dissertation Organization	4
2. Related Work	6
2.1. Introduction	6
2.2. Cloud Computing and Hypervisors	8
2.2.1. Cloud Computing Deployment models	10
2.2.2. Cloud Computing services models	12
2.2.3. Type of hypervisors	14
2.3. Virtualization and Hypervisors type 2.....	17

2.3.1. Oracle VirtualBox	21
2.3.2. VMware Workstation	23
2.3.3. VMware Workstation Server.....	24
2.4. Virtual Machines migration	27
2.5. The OVF (Open Virtualization Format) Files	27
2.5.1. OVF Package Structure	29
3. Requirements Analysis	35
3.1. Oracle Virtualbox	36
3.1.1. Use Case Diagram.	36
3.1.2. APIs/Services and SDKs	37
3.2. VMware Workstation	39
3.2.1. Use Case Diagram.....	39
3.2.2. APIs/Services and SDKs	39
3.3. VirtualMigra.....	41
3.3.1. Use case diagram	41
3.4. The Relational Structured Database	42
4. VirtualMigra Technologies	44
4.1. Resources	44
4.1.1. Windows Server 2008 R2.....	44
4.1.2. Internet Information Services 7	44

4.1.3. Adobe Dreamweaver 8.....	45
4.1.4. Relational Database Management System - MySQL	45
4.1.5. VMware OVF Tool	46
4.1.6. Oracle VboxManage Tool.	46
4.1.7. PHP 5	46
4.1.8. PERL 5	47
4.1.9. SOAP Web Services	48
4.2. Server and Client Architectures.....	49
5. System Demonstration and Validation.....	55
5.1. VirtualMigra Platform Usage	55
5.1.1. VirtualMigra Administrator Interface Environment	56
5.1.2 VirtualMigra User Interface Environment.....	61
6. Conclusions and Future Work	65
References.....	69

List of Figures

FIGURA 1-CLOUD COMPUTING CHARACTERISTICS.	9
FIGURA 2 CLOUD COMPUTING DEPLOYMENT MODELS.	11
FIGURA 3 CLOUD COMPUTING SERVICES MODELS.	14
FIGURA 4 HYPERVISOR TYPE 1.	18
FIGURA 5- HYPERVISOR TYPE 2.	19
FIGURA 6 -OVF FILE DESCRIPTORS EXAMPLE.	33
FIGURA 7- ORACLE VIRTUALBOX API USE CASE.	36
FIGURA 8- VIX USE CASE DIAGRAM.	39
FIGURA 9- VIRTUALMIGRA USE CASE DIAGRAM.	41
FIGURA 10- SERVER AND CLIENT ARCHITECTURE.	50
FIGURA 11- VIRTUALBOX TO VIRTUALBOX MIGRATION.	51
FIGURA 12- WORSTATION TO VIRTUALBOX MIGRATION AND VICE VERSA.	52
FIGURA 13- VIRTUALMIGRA OVF FILES PROCESSING	53
FIGURA 14- ADMINISTRATOR LOGIN PAGE.	56
FIGURA 15- ADMINISTRATOR MANAGEMENT PAGE.	57
FIGURA 16- USER CREATION PAGE.	57
FIGURA 17- USER CREATION SUCCESS PAGE.	58
FIGURA 18- E-MAIL USER PASSWORD.	58
FIGURA 19- USERS LIST PAGE.	59
FIGURA 20- USER DELETE PAGE.	59
FIGURA 21- USER DELETE SUCCESS PAGE.	60
FIGURA 22- ADMINISTRATOR START MIGRATION PAGE.	60
FIGURA 23- USER LOGIN PAGE.	61
FIGURA 24- PASSWORD RECOVERY PAGE.	62
FIGURA 25- LOGOUT PAGE.	62
FIGURA 26- USER MIGRATION MANAGEMENT PAGE.	63

FIGURA 27- USER START MIGRATION PAGE.63

Acronyms

WINDBG	Debugging Tools for Windows
KD	Kernel Debugger
CPU	Central Processing Unit
ER	Entity-Relationship model
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
IO	Input/Output
IT	Information Technology
OS	Operating System
RAM	Random-Access memory
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
UBI	University of Beira Interior
API	Application Programming Interface
IAAS	Infrastructure-as-a- Service
LAN	Local Area Network
PAAS	Platform-as-a-Service
QA	Quality Assurance
SAAS	Software-as-a- Service
SLA	Service Level Agreement
VM	Virtual Machine
VMM	Virtual Machine Monitor

1. Introduction

This chapter details the focus, objectives, contributions, and an overview about the structure of this dissertation. The following sections provide an introduction to the virtualization topic, platforms, technologies, open issues, and also an introduction to a new virtualization platform, called VirtualMigra.

1.1. Focus

The National Institute of Standards and Technology (NIST) [1] defines cloud computing as any computing model with a shared pool of configurable computing resources such as Networks, servers, storage, applications, and services that can be accessed over the network in a convenient manner, and provisioned and released on demand with minimal management or service provider effort. Cloud Datacenters are the facilities that house the computing resources. Resources could be spread across multiple datacenter facilities for reasons such as power constraints and closeness to customers. Different datacenters networks are connected to each other over the Internet to form one large interconnected cloud network [2].

There are numerous advantages to cloud computing for both providers and customers, two main examples are:

- Cloud computing relieves customers of IT responsibilities such as upgrades and maintenance of IT infrastructure. Non-IT companies can focus on their core business and rent their computing requirements from the cloud provider. Providers can have a solid business model on just providing computing and keeping up to date with advancements of such resources.

Cloud computing provides flexibility and economy of scale. Providers can increase productivity by improving on resources as needed and provisioning only what is requested. Customers can request for more or less computing

resources whenever needed and only be charged for what they use. This prevents the customer from investing in more infrastructures if more computing resources are needed [3].

Examples of some challenges and limitations to cloud computing are defining a suitable pay structure for customers, synchronization of data both within and across datacenters, live virtual machine migration, and resource management in the datacenters [3].

Three main problem scenarios are considered for our problem motivation:

Hardware maintenance, scheduled power outage, and server relocation. Assuming a situation arises in a cloud datacenter whereby one of the host servers needs to undergo maintenance, power needs to be shut down, or that a resource need to be physically moved to a different site. In any such a case, the problem that arises is how to handle clients who are currently running virtual machines (VMs) on the subject host server. In a cloud environment these clients are paying for their services, hold service level agreements (SLA) with the cloud provider, and may have very sensitive processes currently running on the subject host. Because all the three scenarios that we focus on are known ahead of time, a solution could be to also inform the client ahead of time to make alternate arrangements.

A better solution is to move the clients' virtual machines running on the subject host to a different host whether permanently or temporarily and transparent to the client. This move is defined as migration. The virtualization tool used in this solution are Oracle Virtual Box and VMware Workstation. This was chosen because research has already started using Type 1 hypervisor in [4]. A type 1 hypervisor sits directly on to the physical hardware and manages all operating systems on the machine, whereas a Type 2 hypervisor sits on to of an operating system.

1.2. Objectives

The main objective of this dissertation is the analysis of a number of application programming interfaces (APIs) in order to provide hypervisors functionalities and simplify the development of web platforms for virtual machine migration.

In order to achieve this main objective, the following partial objectives were identified:

- Review of the related literature including cloud computing;
- Study of related work about virtualization and tools;
- The solution should be designed using type 2 hypervisors;
- Detailed analysis of the type 2 hypervisors ;
- Requirement analysis;
- Proposal and design of the platform;
- Construction and deployment of the web platform VirtualMigra;

The main objective has been widely overcome, and a new set of objectives were proposed.

1.3. Main Contributions

This dissertation focuses on researching new concepts and methods for virtualization system. These concepts represent the foundations of the joined forces of a personal virtualization environment and virtual machines migrations management system, integrated into the VirtualMigra Platform.

The main contribution of this work is the deployment of the VirtualMigra platform into an innovative system model for virtual migrations across different type 2 hypervisors, only possible given a set of related APIs, SDKs, and Web Services available on Oracle Virtualbox and VMware Workstation side.

1.4. Dissertation Organization

This dissertation is organized in six chapters and the chapters are organized as follows.

This chapter, the first, presents the focus of the dissertation including a perspective of the topic, issues, objectives, contributions, and the organization of the document. Chapter 2 describes the related work about cloud computing, hypervisors, virtualization, and virtual machine migration. Each topic is defined and also a set of related associated technologies are identified. Chapter 3 outlines the requirements analysis that was carried out prior to the improvement of the platform, focusing on the user cases, APIs and SDKs. Chapter 4 outlines a detailed description of the VirtualMigra platform, used technologies, architectures, and development scenarios. Chapter 5 details the system demonstration and validation, providing a set of screenshots to demonstrate the platform services aggregation examples. Finally, in chapter 6 is drawn the conclusion and suggested directions for further works.

2. Related Work

This chapter details the related work in cloud computing, hypervisors, virtualization, and virtual machine migration. This chapter focuses mainly on finding issues surrounding the virtualization systems topic.

2.1. Introduction

In current information technology (IT), virtualization refers to the ability to partition the hardware resources of a system such that each partition appears to be a complete hardware platform that can execute an operating system and applications independently from the operating system and applications executing concurrently within other such partitions. Virtualization is becoming an increasingly popular feature implemented in the IT infrastructure of industrial, commercial, and academic enterprises because of a number of benefits including cost reduction, application security, and flexibility. The potential for cost reduction is the main reason driving the popularity of virtualization today.

Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

By consolidating servers dedicated to several different applications onto a single physical computer supporting multiple virtual servers, enterprises can reduce the number of physical computers that must be acquired with concomitant savings in power supply, cooling, space, and maintenance costs. The energy savings also translates into goodwill that an enterprise can generate by appearing to be environmentally friendly.

Virtual hardware is the hardware (including the CPU, controllers, Ethernet devices, and disks) that is seen by the guest software. Virtual machine is the complete environment that supports the execution of guest software

A virtual machine is a full encapsulation of the virtual hardware, virtual disks, and the metadata associated with it. Virtual machines allow multiplexing of the underlying physical machine through a software layer called a hypervisor.

A virtual machine is a software computer that, like a physical machine, runs an operating system and applications. A virtual machine uses the physical resources of the physical machine on which it runs, which is called the host system. Virtual machines have virtual devices that provide the same functionality as physical hardware, but with the additional benefits of portability, manageability, and security.

A virtual machine has an operating system and virtual resources that we manage in much the same way that we manage a physical computer. For example, we install an operating system in a virtual machine in the same way that we install an operating system on a physical computer. We must have a CD-ROM, DVD, or ISO image that contains the installation files from an operating system vendor.

All virtual machines have a hardware version. The hardware version indicates which virtual hardware features that the virtual machine supports, such as BIOS or EFI, number of virtual slots, maximum number of CPUs, maximum memory configuration, and other hardware characteristics. The virtual machine hardware compatibility setting determines the hardware features of the virtual machine.

A virtual machine collection is a service comprised of a set of virtual machines. The service can be a simple set of one or more virtual machines, or it can be a complex service built out of a combination of virtual machines and other virtual machine collections. Because virtual machine collections can be composed, it enables complex nested components.

A virtual appliance is a service delivered as a complete software stack installed on one or more virtual machines.

A virtual appliance is typically expected to be delivered in an OVF package.

2.2. Cloud Computing and Hypervisors

Cloud computing is a model that has five characteristics, namely: rapid elasticity, broad network access, on-demand self-service, resource pooling and measured service. In addition to these characteristics, cloud computing is characterized by four deployment models: public, private, hybrid and community and three main service models which are: SaaS, PaaS and IaaS. In the next section, we try to give an overview of each.

A) Cloud computing characteristics

1) Rapid elasticity

Elasticity is a characteristic of the cloud computing that offers to the customer the ability to increase or decrease used resources at his will very quickly and as needed. The customer will have then the opportunity to increase the resources when he notice a considerable over charge, and to decrease them in normal times, which gives him a real possibility of evolution and resources scaling as needed. As a result, resources for consumers often seem unlimited.

2) Broad network access

Access to resources delivered by the service provider is universal, pervasive, and independent of the network structure and it is done through standard mechanisms promoting the use of heterogeneous platforms: mobile phones, laptops or PDA [5].

3) Resource pooling

Before virtualization, IT managers dedicated a physical server for each application. With virtualization and hypervisors, a single physical machine

can now host multiple virtual machines. In other words, in the Datacenter, the service provider generally adopts a multi-tenant model where we can find two applications in two separate virtual machines in single physical server

4) On-demand self service

One of the most promising characteristic of cloud computing is the self service where the customer can benefit from the service in a short time, without human intervention or any need to contact the responsible. While the principle of payment is equivalent to the consumption of electricity, the invoice will be given to the customer with only the used resources.

5) Measured service

To ensure adequate service control, the last characteristic of cloud computing offers a control and optimization of resources with the monitoring and reporting provided to the consumer and the supplier for a transparency of the used services.

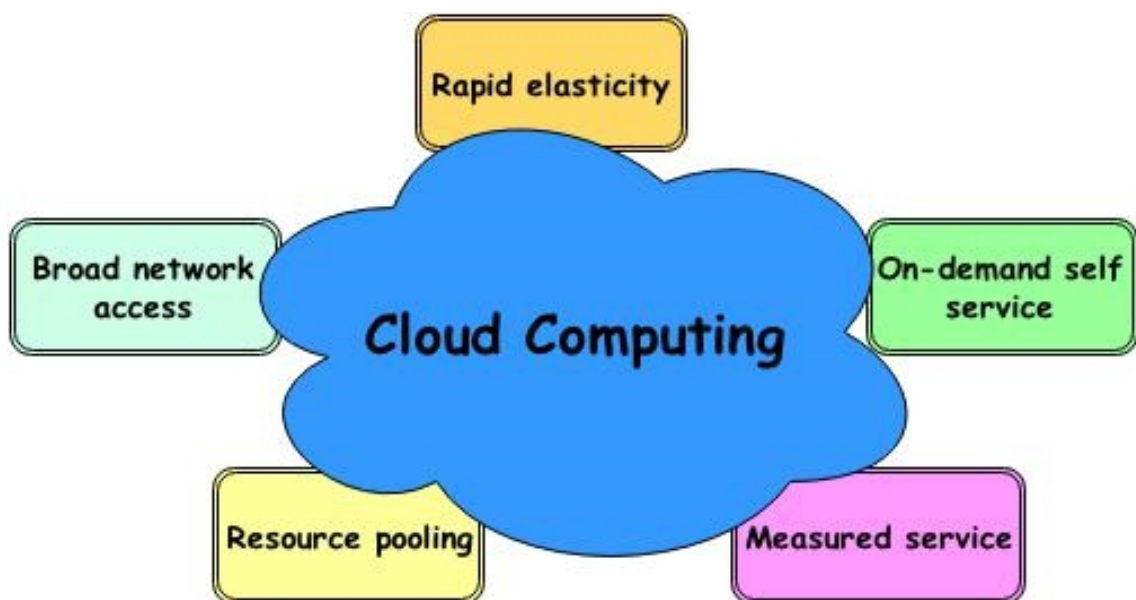


Figura 1-Cloud computing characteristics.

2.2.1. Cloud Computing Deployment models

Based on the general architecture of cloud computing, there are four deployment models: private, public, hybrid and community which are detailed in the following:

1) Private Cloud

Private cloud is defined as the establishment of a dedicated infrastructure to the client and accessed via secure networks for a limited number of users (learners, teachers, tutors ...).

Therefore; the infrastructure is directly administered by the customer or by a trusted provider. In the private cloud, there are two visions:

In the client side, implementing a private cloud is the transforming the internal infrastructure with virtualization and automation technologies to deliver more simply, more quickly and on demand resources and services.

In the provider side, a private cloud is reflected in the fact of allocation to a client a set of resources accessible via secure networks, e.g.: VPN [6].

2) Public Cloud

The public cloud is an external infrastructure to the organization, accessible via internet, managed by an owner of infrastructure with shared resources across multiple organizations. The environment can be sold to any class of buyer and thus shared with a virtually unlimited number of users on the same platform. The problem we run into with a public cloud is that the location of resources is unknown, but the advantage is being able to "rent" the resource and pay only what the client uses.

3) Hybrid Cloud

In the hybrid cloud, the company uses both a private cloud and another public. This type is often used in cases where there would be a scalability issues. In this case, it would be important to link the private cloud or internal cloud with the public one in order to communicate the two infrastructures.

4) Community Cloud

The cloud community model aims to broaden the classic scope of cloud computing since it allows the pooling of resources and collaboration beyond the boundaries of a single organization. Thus a solution to a specific professional community, an association or a government cloud that have common needs and interests (the same type of applications, security requirements ...). It can also host a generic business application and highly specialized, but common to this community, which decided to build a cloud to host and manage.

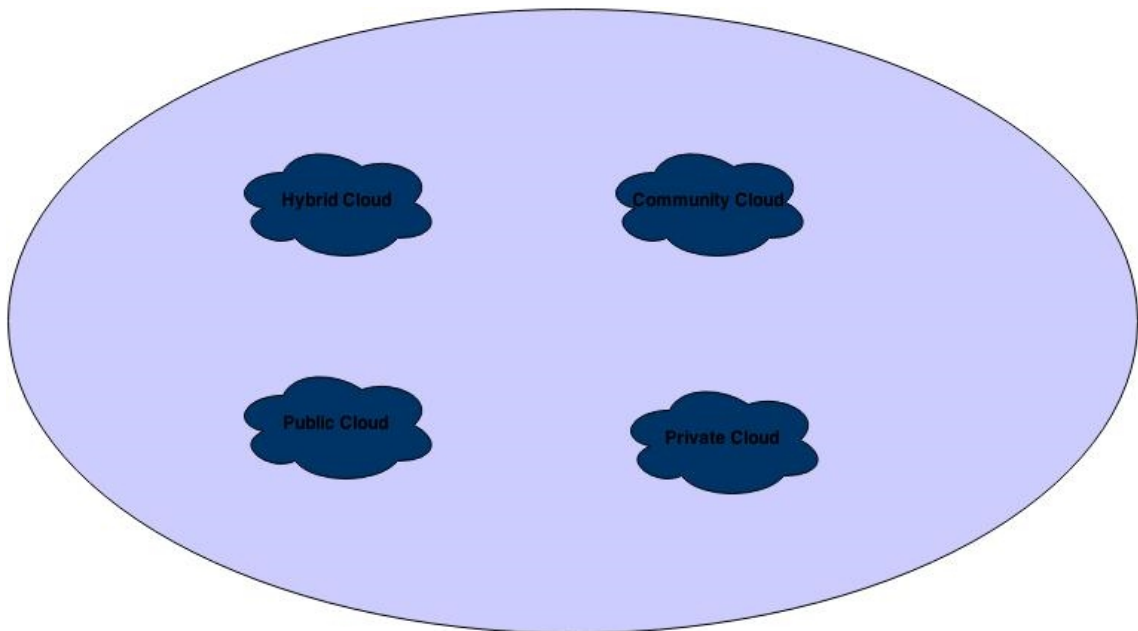


Figura 2 Cloud Computing Deployment models.

2.2.2. Cloud Computing services models

We can summarize the services offered by cloud computing in a pyramid consisting of three main layers as shown in Figure 3.

Depending on customer needs, service providers can offer three different packages, either infrastructure IaaS, platform PaaS or application SaaS, with the latter being the most widespread service model in the cloud world.

1) IaaS

The IaaS means Infrastructure as a service. This is the infrastructure part of the cloud, where the customer rents the CPU, memory, storage ... and the cost is directly related to occupancy rates. IaaS has several advantages:

- The client has full control which allows him to install any type of business software.
- Dynamic Scalability according to its requirements.
- Rapid implementation.
- Payment according to the use.

2) PaaS

The PaaS provides a flexible platform, distributed and virtualized managed by the external provider in order to develop and deploy applications and services. These are actually development tools (such as databases and development studios which are provided as a Web solution), and the necessary infrastructure to host the developed services.

The Paas is based on the flexible architecture provided by the Iaas which makes it easier for the design, development, testing, deployment and hosting of applications. We find among the benefits of Paas:

- The company will focus on development without worrying about infrastructure or development tools which will be provided the cloud supplier.
- After developing the application, the deployment is automated.
- No additional software to buy or install.

3) SaaS

The acronym SaaS is the most known service model in the cloud computing world, its meaning is Software as a service. It is an application deployment model where a provider leases an application to its customers (whether a company or an individual) as a service instead of charge licenses. This type of model transforms the software budget in variable expenditures instead of fixed. The SaaS offers several advantages:

- No need to handle the installation or update, the supplier will do so.
- Flexibility and ease of implementation.
- Payment for use.
- The customer has the opportunity to test new software with ease.
- Devices with minimal hardware requirements (mobile phones, for example) could be successfully used as cloud clients [7].

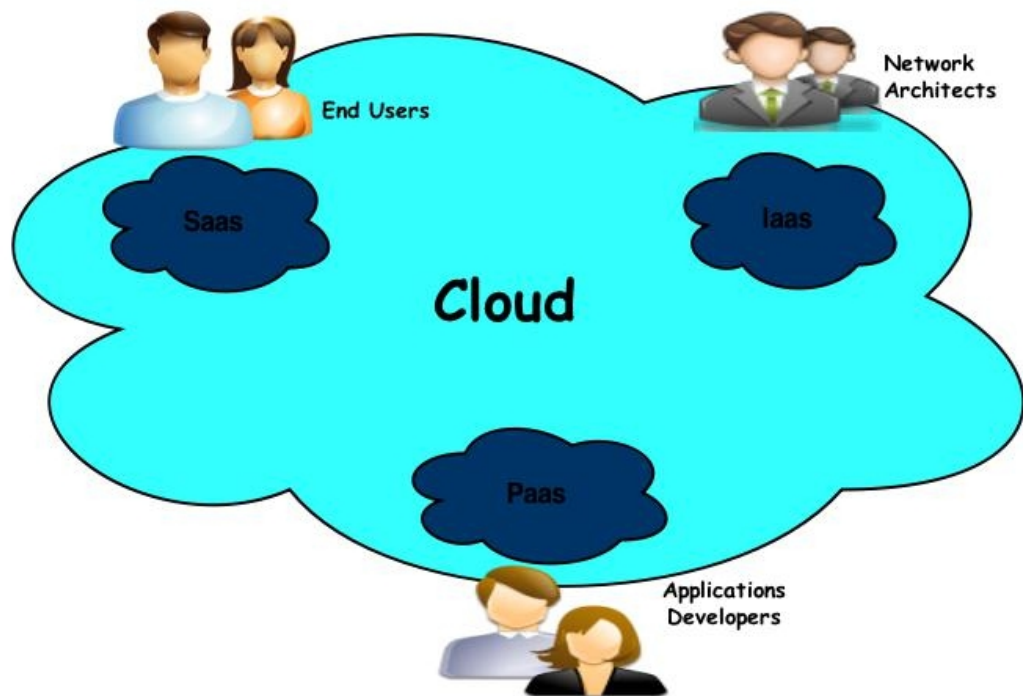


Figura 3 Cloud Computing services models.

2.2.3. Type of hypervisors

The purpose of cloud computing is to commoditize systems and hardware so the focus can be shifted to content and service delivery. However, this concept is more reliant on sound infrastructure than ever. One of the key components in rendering cloud services is a specialized server called a hypervisor. This server hosts the virtualized operating systems and applications which power the cloud services. These hypervisors are capable of hosting multiple virtual machines at the same time. They are a crucial component in the cloud architecture.

Hypervisors either hold virtual machines on their internal file system or have access to remote virtual machine storage. They also have access to data storage and to the otherwise-isolated management network. The ability to copy, destroy, or modify large quantities of output is extremely appealing. By design, hypervisors sit atop significant computing hardware.

On another scale, the stability of cloud operations can be compromised if a corrupted hypervisor reports conflicting updates of system state. By altering the metadata it is possible to tie up resources on other hypervisors, force the migration of virtual machines, and confuse load balancing processes in the system manager. These risks have a scary implication: if one hypervisor is not trusted the entire cloud becomes irrelevant.

Organizations have been quick to integrating cloud computing into their information technology strategies. Clouds provide an additional layer of abstraction between computing hardware and the services which end users consume. Clouds may be designed for private or public access. They may be hosted within the organization or contracted via service level agreements. Some firms lease the infrastructure, some use the platform, and yet others pay for software as a service. Despite the many nuances, all clouds are based on the same concept: they are a platform for providing information services to geographically-dispersed individuals. A number of factors have spurred the growth of this infrastructure. For instance, clouds reduce the cost of IT infrastructure. Data centers require an ongoing investment in computer hardware. Clouds provide access to the same resources without the upfront cost. Further, they provide scalability without plateaus or lumpy investments. More storage can be added on demand. Finally, clouds provide consistent global access. Employees, customers, and partners from all over the world can access an organization's information resources.

Virtual machines run on hosts with specialized programs called hypervisors. There are two types of hypervisors. The first type resembles applications which can be installed on top of a computer's existing operating system. They are best suited for use in the early development and configuration stages of virtual machines. The second category of hypervisors called "bare-metal" is self-contained operating systems. With less overhead, they are more efficient and easier to customize [9]. For this reason, data centers most commonly equip their hosts with bare-metal hypervisors.

Because datacenters often employ hundreds or even thousands of hypervisors, specialized servers are used to manage the hypervisors. Depending on the range of their administrative capabilities, these servers are often called virtualization or cloud managers. They aggregate the hypervisor's computing resources and distribute the virtual machines workload.

Because of their critical role in provisioning information services, hypervisors are often targeted for attack. Their slim operating systems have minimal security features, making them relatively easy to enter. This can be accomplished by brute force, social engineering, or by escalation of privileges. Once access is obtained, attackers may target the virtual machines or data stores, or they may opt to exploit the hypervisor's operating system as a base for running illicit programs [10]. Further, the attack may focus on causing havoc in the cloud, using the hypervisor as a tool to inject malicious metadata about system state [11].

Hypervisors presently control both memory allocation and memory isolation through the same technique-address translation. The hypervisor determines a set of memory pages to be allocated to the virtual machines and maintains a mapping table for each. It processed a nest page table to further disassociate memory allocations from actual hardware. This interesting approach appears to make it difficult for virtual machines to escape the confines of their allocated space, but it does not prevent attackers from accessing system memory and compromising metadata.

2.3. Virtualization and Hypervisors type 2

The motivations for introducing virtualization are many:

- Improved system utilization - multiprogramming.
- Simplified system migration.
- Reduced downtime at system upgrade.
- Running heterogeneous systems (multiple OSes) on a single machine.
- Running legacy applications on new hardware.
- Providing secure execution environments or secure monitoring

Among these, simplified system migration and reduced downtime do not really fit into the context of most embedded systems. Furthermore, as pointed out in [12], with the current trend towards usage of multicore chips, heterogeneous operating system support becomes a less viable argument for virtualization as each core or a dedicated set of cores in such a system can be configured to run a particular OS (as long as the memory can be physically partitioned). The system utilization, hardware abstraction and not least the security arguments for hypervisors are still viable even on multicore chips.

Hypervisors are useful because they can encapsulate a running system, including kernel and hardware state, within the abstraction of a virtual machine, they mediate all interactions between hardware and guest virtual machines, and they provide correctness and performance isolation among virtual machines [13].

A Hypervisor, also known as a Virtual Machine Monitor [14], coordinates operations between a VM and the physical hardware through virtual hardware. The hypervisor can multiplex access to the physical hardware, giving the perception of multiple copies of a single hardware. This allows multiple virtual machines running simultaneously.

There are two major architectures of virtualization software available: hosted or type 2 and bare-metal or type 1 as show in Figure 4. While both

types of software can be applied in engineering and specifically test applications, there are important differences between the two architectures that should be understood before beginning a design. Specifically, the lack of broad I/O access and support for real-time operating systems in hosted virtualization software may make bare-metal virtualization a better solution for many test applications.

In the type 2 virtualization architecture, hypervisor software is installed directly on a host operating system such as Windows as show in Figure 5.

While the hypervisor allows multiple operating systems to execute in parallel on the same computer, scheduling and I/O access are provided by the underlying host OS. This is an important difference from type 2 virtualization; because hosted virtualization software relies on a host OS for scheduling capabilities, most hosted solutions cannot guarantee the determinism of virtualized operating systems running on top of the hypervisor. Some examples of Hypervisors Type I are VMware ESX, Hyper-V, and Xen [15] and for Hypervisors Type II Virtualbox and Workstation.

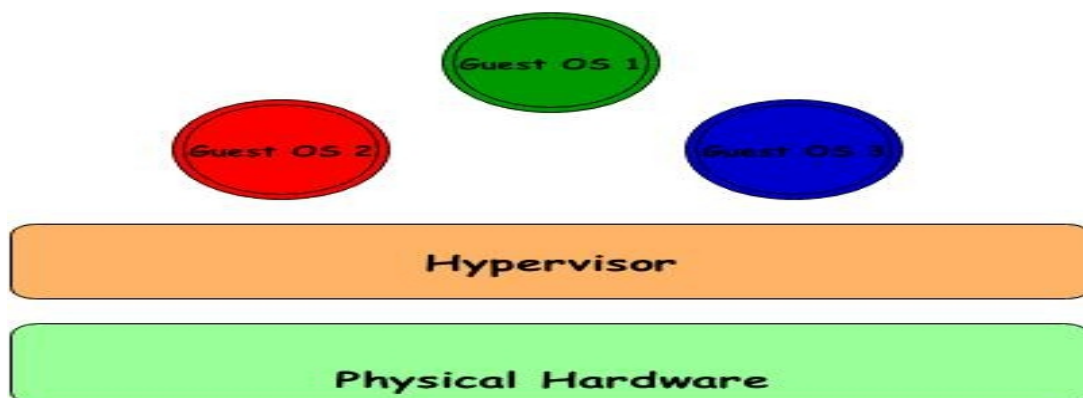


Figura 4 Hypervisor type 1.

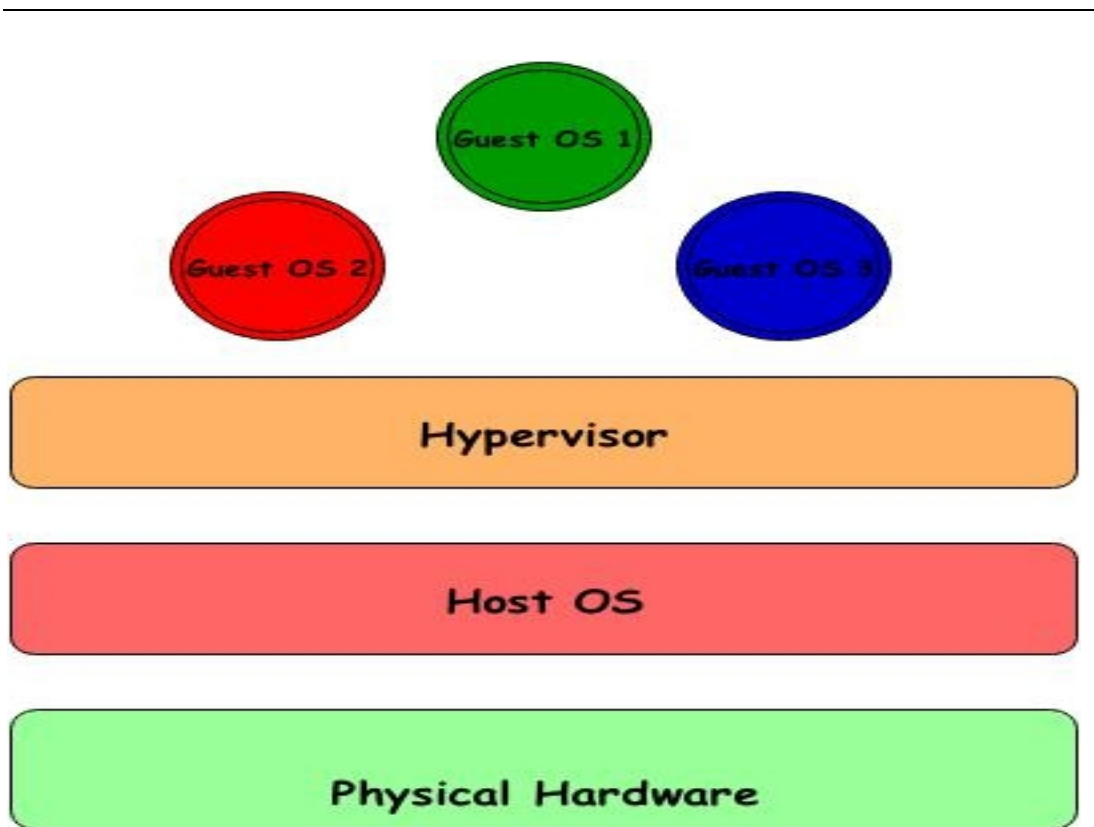


Figura 5- Hypervisor type 2.

From the perspective of the physical processor, the hypervisor needs to perform the following two main functions save and restore the state of the virtual processor - this includes the user-visible general purpose registers as well as the system registers (e.g., mode control registers, memory management unit's registers and the interrupt controller's registers). Essentially, whenever a guest operating system accesses these registers or executes any instruction that updates the system's state or reports on important system state, the processor, instead of executing the instruction, invokes the hypervisor. The hypervisor emulates the execution of the instruction, storing the results in appropriate general purpose registers, if required (i.e., any changes to the general purpose registers expected from the normal execution of the instruction are reported back to the guest environment in the guest's general purpose registers). The hypervisor then resumes execution of the guest at the point after the instruction that

caused the hypervisor to be invoked the operating system continues to control all aspects of the computer such as memory management (including paging and virtual memory), I/O, and interrupt handling. Note that the operating system is unaware of the existence of the hypervisor and the hypervisor is invoked asynchronously by the processor in response to certain events. Therefore, the hypervisor's code and data is stored in locked (i.e., immovable, always resident) areas of memory [16].

The control fields are used by the hypervisor to define the events that cause the physical processor to stop the execution of the virtual machine and return control to the hypervisor.

These fields also provide information to the hypervisor as to the reason why the hypervisor is being invoked (e.g., which instruction caused the virtual machine to exit, the instruction's length, and which registers are affected). This information can be used by the hypervisor to emulate the execution of the instruction/event and resume the execution of the virtual machine.

The guest state fields hold the contents of the control, descriptor, and segment registers that are loaded into the virtual processor when the guest code begins or resumes execution. The hypervisor initializes these fields to setup a new virtual machine and the physical processor stores the contents of the virtual processor's control registers back into these fields when the virtual machine execution stops and control is returned to the hypervisor. The hypervisor can then modify these registers as needed in order to emulate the execution of the instruction or event that cause the hypervisor to be invoked before resuming the execution of the virtual machine [16].

2.3.1. Oracle VirtualBox

VirtualBox is a cross-platform virtualization application who can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

The techniques and features that VirtualBox provides are useful for several scenarios:

Running multiple operating systems simultaneously. VirtualBox allows to run more than one operating system at a time. This way, we can run software written for one operating system on another (for example, Windows software on Linux or a Mac) without having to reboot to use it. Since we can configure what kinds of "virtual" hardware should be presented to each such operating system, we can install an old operating system such as DOS or OS/2 even if the real computer's hardware is no longer supported by that operating system [17].

Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With VirtualBox, such a complex setup (then often called an "appliance") can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into VirtualBox.

When dealing with virtualization it helps to acquaint oneself with a bit of crucial terminology, especially the following terms:

The Host operating system (host OS) is the operating system of the physical computer on which VirtualBox was installed. There are versions of VirtualBox for Windows, Mac OS X, Linux and Solaris hosts;

The Guest operating system (guest OS) is the operating system that is running inside the virtual machine. Theoretically, VirtualBox can run any x86 operating system (Windows, FreeBSD, OpenBSD), but to achieve near-

native performance of the guest code on the machine, we had to go through a lot of optimizations that are specific to certain operating systems. So while our favorite operating system may run as a guest.

The VM is the special environment that VirtualBox creates for the guest operating system while it is running. In other words, we can run a guest operating system "in" a VM. Normally, a VM will be shown as a window on the computer's desktop.

Internally, VirtualBox thinks of a VM as a set of parameters that determine its behavior. They include hardware settings (how much memory the VM should have, what hard disks VirtualBox should virtualize through which container files, what CDs are mounted etc.) as well as state information (whether the VM is currently running, saved, its snapshots etc.).

VirtualBox is functionally identical on all of the host platforms, and the same file and image formats are used. This allows to run virtual machines created on one host on another host with a different host operating system; for example, it's possible to create a virtual machine on Windows and then run it under Linux.

In addition, virtual machines can easily be imported and exported using the Open Virtualization Format (OVF), an industry standard created for this purpose.

VirtualBox can present up to 32 virtual CPUs to each virtual machine, irrespective of how many CPU cores are physically present on the host [18].

2.3.2. VMware Workstation

VMware Workstation is a desktop software that allows us to run multiple x86-compatible desktop and server operating systems simultaneously on a single PC, in a fully networked, portable virtual machine with no rebooting or hard drive partitioning required.

Workstation is used in the software development, quality assurance, training, sales, and IT fields.

Some Workstation streamlines software development and testing:

- Develop and test multiple operating systems and applications on a single PC.
- Connect virtual machines to simulate and test multitier configurations.
- Use multiple snapshots and debugging support to facilitate testing.
- Archive test environments on file servers where they can be easily restored or shared [19].

Workstation enhances productivity of IT professionals:

- Configure and test desktops and servers as virtual machines before deploying them to production.
- Test new multitier applications, application updates, and operating system patches on a single PC.
- Host legacy applications within virtual machines, facilitating operating system migrations and eliminating the need to port legacy applications [19].
- Create a virtual library of end-user configurations on a shared drive.

Workstation facilitates computer-based training and software demos:

- Package and deploy classroom material in virtual machines.
- Allow students to experiment with multiple operating systems, applications, and tools in secure, isolated virtual machines.

- Configure virtual machines to undo all changes at shutdown.
- Demo complex or multitier configurations on a single laptop.

We can also use the standalone OVF Tool to convert an OVF virtual machine to VMware runtime format. Application developers can use APIs, SDKs, and IDEs to write and debug applications that run in virtual machines [19].

Virtual hard disk files include information such as the operating system, program files, and data files.

Virtual disk files have a .vmdk extension.

With virtual symmetric multiprocessing (SMP), we can assign processors and cores per processor to a virtual machine on any host system that has at least two logical processors.

Workstation considers multiprocessor hosts that have two or more physical CPUs, single-processor hosts that have a multicore CPU, and single-processor hosts that have hyperthreading enabled, to have two logical processors [20].

2.3.3. VMware Workstation Server

VMware Workstation Server is a service that runs on the Workstation host system. Remote Workstation users connect to VMware Workstation Server when they run shared virtual machines on the host system.

We can enable or disable virtual machine sharing and remote access, change the HTTPS port that VMware Workstation Server uses on the host system, and change the shared virtual machines directory.

When we enable virtual machine sharing and remote access, Workstation starts VMware Workstation Server and configures the service to start with the host system.

When we disable virtual machine sharing and remote access, Workstation disables virtual machine sharing and stops VMware Workstation Server. We

cannot create shared virtual machines and remote users cannot connect to the host system.

We can use Workstation to connect to a remote server that is running Workstation, ESX, ESXi, or vCenter Server [21].

When we connect to a remote server for the first time, Workstation asks we whether to save the login information. We can configure Workstation to never ask we to save login information for a remote server.

After we connect to a remote server, the remote host and remote virtual machines appear in the library. If the remote server is running vCenter Server, datacenters and folders appear in the library [22].

To interact with a remote host, we select it in the library. The tasks that we can perform on a remote host appear on the tab for the remote host. For example, we might be able to restart, shut down, or suspend the remote host and create virtual machines.

To interact with a remote virtual machine, we select it in the library. We interact with remote virtual machines in the same way that we interact with local virtual machines, but some features and devices are not supported.

The permissions determine the actions that we can perform on remote hosts and remote virtual machines. When a feature is not supported, or when we do not have permission to use it, the associated menu item is unavailable.

When we upload a virtual machine to a remote server, Workstation copies the virtual machine to the remote host and datastore that we select. The original virtual machine remains on the host system.

A role is a predefined set of privileges. Privileges define individual rights that a user requires to perform actions and read properties. Workstation

includes a default set of system roles. We can also create their own roles [23].

A single user might have different roles for different objects. For example, if we have two shared virtual machines, virtual machine A and virtual machine B, we might assign a particular user the Administrator role on virtual machine A and the Read Only permission on virtual machine B.

We can control which users can access remote hosts and shared virtual machines by creating permissions. To create a permission, we pair a user or group with a role and associate that pairing with an object. The role defines the actions that a user or group can perform, the user or group indicates who can perform the actions, and the object is the target of the actions.

Users can inherit permissions through group membership and through the object hierarchy. When we assign permissions to a group, all of the users in the group inherit those permissions. If we define multiple group permissions on the same object and a user belongs to two or more of those groups, the user inherits all of the privileges assigned to the groups. If we define a permission for the user on the object, that permission takes precedence over all group permissions [24].

2.4. Virtual Machines migration

VM migration, from a source cloud to a destination one over a Wide Area Network (WAN) implies to transfer three elements: memory, status, and storage (the disk-image containing file system and user data). This may represent a bottleneck as it implies to transfer GB and GB of data over the network, with a consequent waste of resources, both in the source and the destination Clouds [25].

VM migration consists of transferring a VM from a source to a destination physical host. Basically there are two kinds of VM migration: “hot” (or live) and “cold” migration. In “hot” migration the VM does not lose its status and the user does not perceive any change. In “cold” migrations the VM loses its status and the user notices a service interruption. The downtime is referred to the time elapsing from the instant when the VM is turned off in the source host and the moment in which the same VM is turned on in the destination host. If the downtime is negligible and the VM status is maintained, the migration is defined “hot”, otherwise “cold” [26].

2.5. The OVF (Open Virtualization Format) Files

OVF is a platform-independent, efficient, extensible, and open packaging and distribution format for virtual machines. OVF format provides a complete specification of the virtual machine, including the full list of required virtual disks and the required virtual hardware configuration, including CPU, memory, networking, and storage. An administrator can quickly provision an OVF-formatted virtual machine with little or no intervention [27].

The Open Virtualization Format (OVF) Specification describes an open, secure, portable, efficient and extensible format for the packaging and

distribution of software to be run in virtual machines. The key properties of the format are as follows:

- Optimized for distribution

OVF supports content verification and integrity checking based on industry-standard public key infrastructure, and it provides a basic scheme for management of software licensing.

- Optimized for a simple, automated user experience OVF supports validation of the entire package and each virtual machine or metadata component of the OVF during the installation phases of the virtual machine (VM) lifecycle management process [28]. It also packages with the package relevant user-readable descriptive information that a virtualization platform can use to streamline the installation experience.

- Supports both single VM and multiple-VM configurations

OVF supports both standard single VM packages and packages containing complex, multi-tier services consisting of multiple interdependent VMs.

- Portable VM packaging

OVF is virtualization platform neutral, while also enabling platform-specific enhancements to be captured. It supports the full range of virtual hard disk formats used for hypervisors today, and it is extensible, which allow it to accommodate formats that may arise in the future. Virtual machine properties are captured concisely and accurately.

- Vendor and platform independent

OVF does not rely on the use of a specific host platform, virtualization platform, or guest operating system.

- Extensible

OVF is immediately useful – and extensible. It is designed to be extended as the industry moves forward with virtual appliance technology. It also

supports and permits the encoding of vendor-specific metadata to support specific vertical markets [28].

- Localizable

OVF supports user-visible descriptions in multiple locales, and it supports localization of the interactive processes during installation of an appliance. This capability allows a single packaged appliance to serve multiple market opportunities.

- Open standard

OVF has arisen from the collaboration of key vendors in the industry, and it is developed in an accepted industry forum as a future standard for portable virtual machines. It is not an explicit goal for OVF to be an efficient execution format. A hypervisor is allowed but not required to run software in virtual machines directly out of the Open Virtualization Format.

2.5.1. OVF Package Structure

An OVF package shall consist of the following files:

- one OVF descriptor with extension `.ovf`
- zero or one OVF manifest with extension `.mf`
- zero or one OVF certificate with extension `.cert`
- zero or more disk image files
- zero or more additional resource files, such as ISO images

The file extensions `.ovf`, `.mf` and `.cert` shall be used.

OVF does not require any specific disk format to be used, but to comply with this specification the disk format shall be given by a URI which

identifies an unencumbered specification on how to interpret the disk format. The specification need not be machine readable, but it shall be static and unique so that the URI may be used as a key by software reading an OVF package to uniquely determine the format of the disk [29]. The specification shall provide sufficient information so that a skilled person can properly interpret the disk format for both reading and writing of disk data. It is recommended that these URIs are resolvable.

The Envelope element describes all metadata for the virtual machines (including virtual hardware), as well as the structure of the OVF package itself as show in Figure 6.

The outermost level of the envelope consists of the following parts:

- A version indication, defined by the XML namespace URIs.
- A list of file references to all external files that are part of the OVF package, defined by the References element and its File child elements. These are typically virtual disk files, ISO images, and internationalization resources.
- A metadata part, defined by section elements.
- A description of the content, either a single virtual machine (VirtualSystem element) or a collection of multiple virtual machines (VirtualSystemCollection element).
- A specification of message resource bundles for zero or more locales, defined by a Strings element for each locale.

The file reference part defined by the References element allows a tool to easily determine the integrity of an OVF package without having to parse or interpret the entire structure of the descriptor. Tools can safely manipulate (for example, copy or archive) OVF packages with no risk of losing files.

Virtual machine configurations in an OVF package are represented by a

“VirtualSystem” or “VirtualSystemCollection” element. These elements shall be given an identifier using the “ovf:id” attribute. Direct child elements of a “VirtualSystemCollection” shall have unique identifiers.

The “VirtualSystem” element describes a single virtual machine and is simply a container of section elements.

This specification defines three conformance levels for OVF descriptors, with “1” being the highest level of conformance [29]:

- OVF descriptor uses only sections and elements and attributes that are defined in this specification.

Conformance Level: 1.

- OVF descriptor uses custom sections or elements or attributes that are not defined in this specification.

Conformance Level: 2.

- OVF descriptor uses custom sections or elements that are not defined in this specification and at least one such extension is required. The definition of all required extensions shall be publicly available in an open and unencumbered XML Schema. The complete specification may be inclusive in the XML schema or available as a separate document.

Conformance Level: 3.

The use of conformance level 3 limits portability and should be avoided if at all possible.

The conformance level is not specified directly in the OVF descriptor but shall be determined by the above rules.

This specification supports abstract or incomplete hardware descriptions in which only the major devices are described. The hypervisor is allowed to create additional virtual hardware controllers and devices, as long as the required devices listed in the descriptor are realized.

The consumer of the OVF package should select the most appropriate virtual hardware description for the particular virtualization platform.

Virtual disks and their metadata are described outside the virtual hardware to facilitate sharing between virtual machines within an OVF package.

The OVF environment defines how the guest software and the deployment platform interact. This environment allows the guest software to access information about the deployment platform, such as the user-specified values for the properties defined in the OVF descriptor.

The environment specification is split into a protocol part and a transport part. The protocol part defines the format and semantics of an XML document that can be made accessible to the guest software. The transport part defines how the information is communicated between the deployment platform and the guest software.

The environment document is an extensible XML document that is provided to the guest software about the environment in which it is being executed [30].

```

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData"
  xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
  xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns="http://schemas.dmtf.org/ovf/envelope/1"
  xml:lang="en-US">
<References>
  <File ovf:id="de-DE-resources.xml" ovf:size="15240"
    ovf:href="http://mywebsite/virtualappliances/de-DE-resources.xml"/>
  <File ovf:id="file1" ovf:href="vmdisk1.vmdk" ovf:size="180114671"/>
  <File ovf:id="file2" ovf:href="vmdisk2.vmdk" ovf:size="4882023564"
ovf:chunkSize="2147483648"/>
  <File ovf:id="file3" ovf:href="resource.iso" ovf:size="212148764"
ovf:compression="gzip"/>
  <File ovf:id="icon" ovf:href="icon.png" ovf:size="1360"/>
</References>
<!-- Describes meta-information about all virtual disks in the package -->
<DiskSection>
  <Info>Describes the set of virtual disks</Info>
  <!-- Additional section content -->
</DiskSection>
<!-- Describes all networks used in the package -->
<NetworkSection>
  <Info>List of logical networks used in the package</Info>
  <!-- Additional section content -->
</NetworkSection>
<SomeSection ovf:required="false">
  <Info>A plain-text description of the content</Info>
  <!-- Additional section content -->

```

Figura 6 -OVF File descriptors example.

3. Requirements Analysis

This chapter describes the requirements analysis, also called requirements engineering consisting in the process of determining user expectations for a new or modified product. This process of requirements analysis was started before the development of this platform in the pre-development of the same.

After a strict analysis, we had to choose a programming language to build the platform, two different type 2 hypervisors with an API compatible with the chosen language, a Data Base Management System and a Web Server. Regarding hypervisors were chosen “Oracle VirtualBox” and “VMware Workstation”, take account “Oracle Virtualbox” has an API for PHP language and Workstation an API for Perl language, two server side languages for web development.

The VirtualMigra platform was programmed entirely with PHP 5 using the Oracle VirtualBox (PHP) and VMware Workstation (PERL) APIs. A brief description of this programming languages and other used technologies are presented in Chapter 4.

In the following sections, we presented the study how each APIs/Services and SDKs and the effective operational structure design of the VirtualMigra Platform.

This operational structure has been changed and adapted several times during the different development stages.

3.1. Oracle Virtualbox

3.1.1. Use Case Diagram.

Use case diagram is a methodology used to identify, clarify and organize system requirements, composed by a set of possible interactions sequences between systems and users in a particular environment. In the Figure 7 we present a partial “Oracle VirtualBox” API use case diagram.

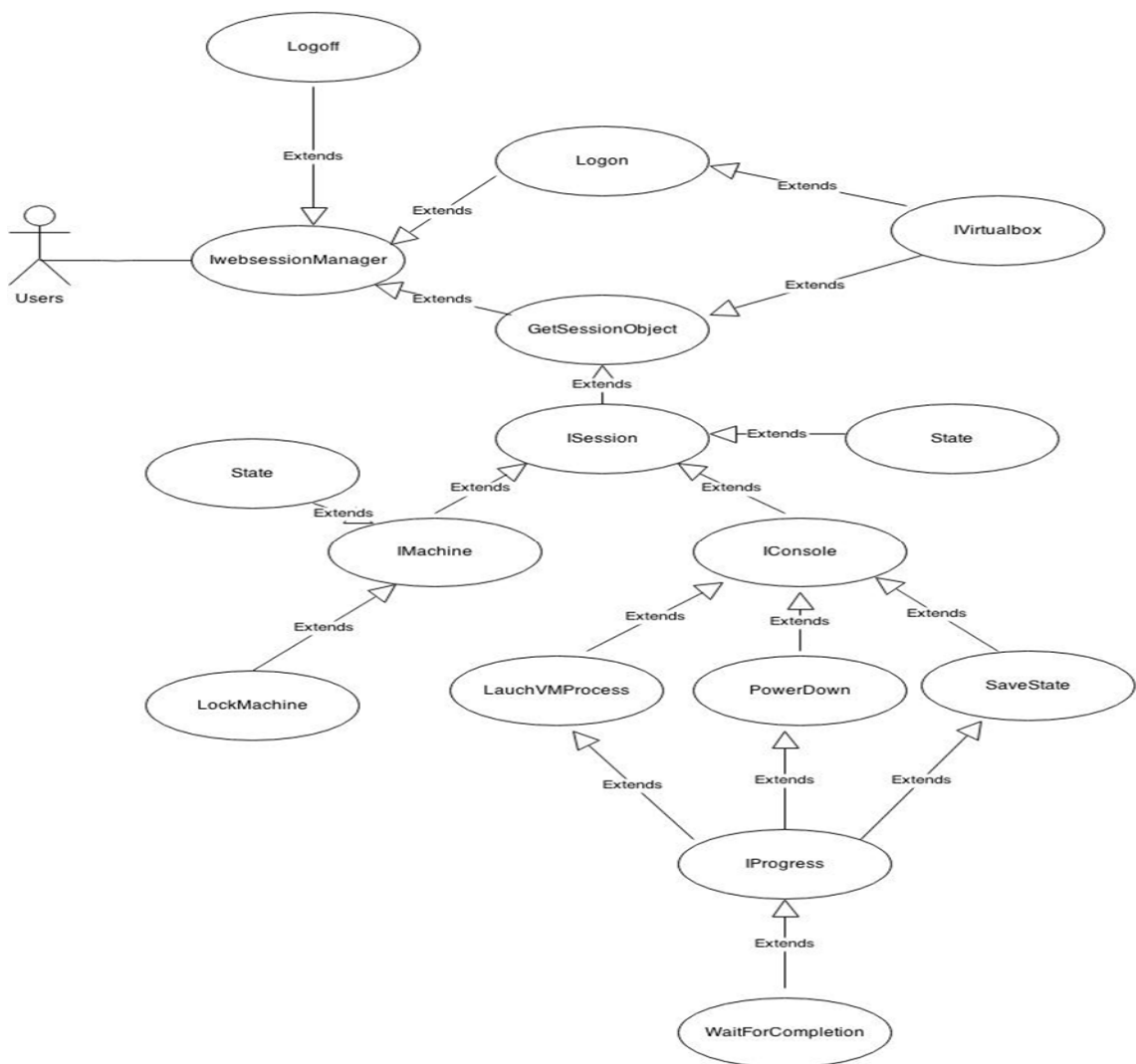


Figura 7- Oracle VirtualBox API use case.

3.1.2. APIs/Services and SDKs

Developers have at their disposal a set of methods delivered in the form of APIs, Web Services and SDKs, allowing VirtualBox comes with comprehensive support for third-party developers. The "Main API" of VirtualBox exposes the entire feature set of the virtualization engine.

The API is made available to C++ clients through COM (on Windows hosts) or XPCOM (on other hosts). Bridges also exist for SOAP, Java and Python [31].

VirtualBox comes with a web service that maps nearly the entire API. The web service ships in a stand-alone executable that, when running, acts as an HTTP server, accepts SOAP connections and processes them.

Since the entire web service API is publicly described in a web service description file (in WSDL format). These days, that includes most programming languages that are available: Java, C++, .NET, PHP, Python, Perl [32].

Web services are a particular type of programming interface. Whereas, with "normal" programming, a program calls an application programming interface (API) defined by another program or the operating system and both sides of the interface have to agree on the calling convention and, in most cases, use the same programming language, web services use Internet standards such as HTTP and XML to communicate.

In order to successfully use a web service, a number of things are required - primarily, a web service accepting connections, a service descriptions and then a client that connects to that web service. The connections are governed by the SOAP standard, which describes how messages are to be exchanged between a service and its clients. The service descriptions are governed by WSDL.

These previous technologies relied on specific binary protocols and thus proved to be difficult to use between diverging platforms, web services enfold these incompatibilities by using text-only standards like HTTP and XML.

The relevant standards here are called SOAP and WSDL, where SOAP describes the format of the messages that are exchanged (an XML document wrapped in an HTTP header), and WSDL is an XML format that describes a complete API provided by a web service. WSDL in turn uses XML Schema to describe types.

Whenever a client makes a call to a web service, this involves a complicated procedure internally. These calls are remote procedure calls. Each such procedure call typically consists of two “message” being passed, where each message is a plain-text HTTP request with a standard HTTP header and a special XML document following. This XML document encodes the name of the procedure to call and the argument names and values passed to it [33]

3.2. VMware Workstation

3.2.1. Use Case Diagram

Use case diagram is a methodology used to identify, clarify and organize system requirements, composed by a set of possible interactions sequences between systems and users in a particular environment. In the Figure 8 we present a partial VMware Workstation API use case diagram.

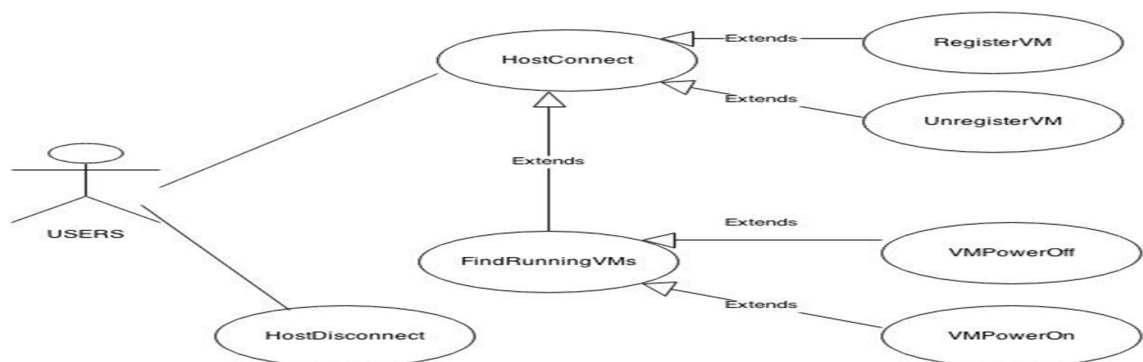


Figura 8- VIX use case diagram.

3.2.2. APIs/Services and SDKs

The VIX API allows to automate virtual machine operations on most current VMware platform products.

The VIX API is a library for writing scripts and programs to manipulate virtual machines. It is high-level, easy to use, and practical for both script developers and application programmers.

The Vix API is object-based. Most API functions either create objects or operate on the properties of existing objects.

Vix objects works with handles. Handles are opaque identifiers (actually integers) that can be passed as parameters to functions. Handles are run-time only and are unique only within a client's address space.

We use the VIX API to write programs that automate virtual machine operations. Functions enable we to power virtual machines on and off, register them, and run programs to manipulate files in the guest operating systems. Additional language bindings are available for Perl, COM, and shell scripts [34].

3.3. VirtualMigra

3.3.1. Use case diagram

Use case diagram is a methodology used to identify, clarify and organize system requirements, composed by a set of possible interactions sequences between systems and users in a particular environment. In Figure 9 we present the main use case diagram of the VirtualMigra Platform.

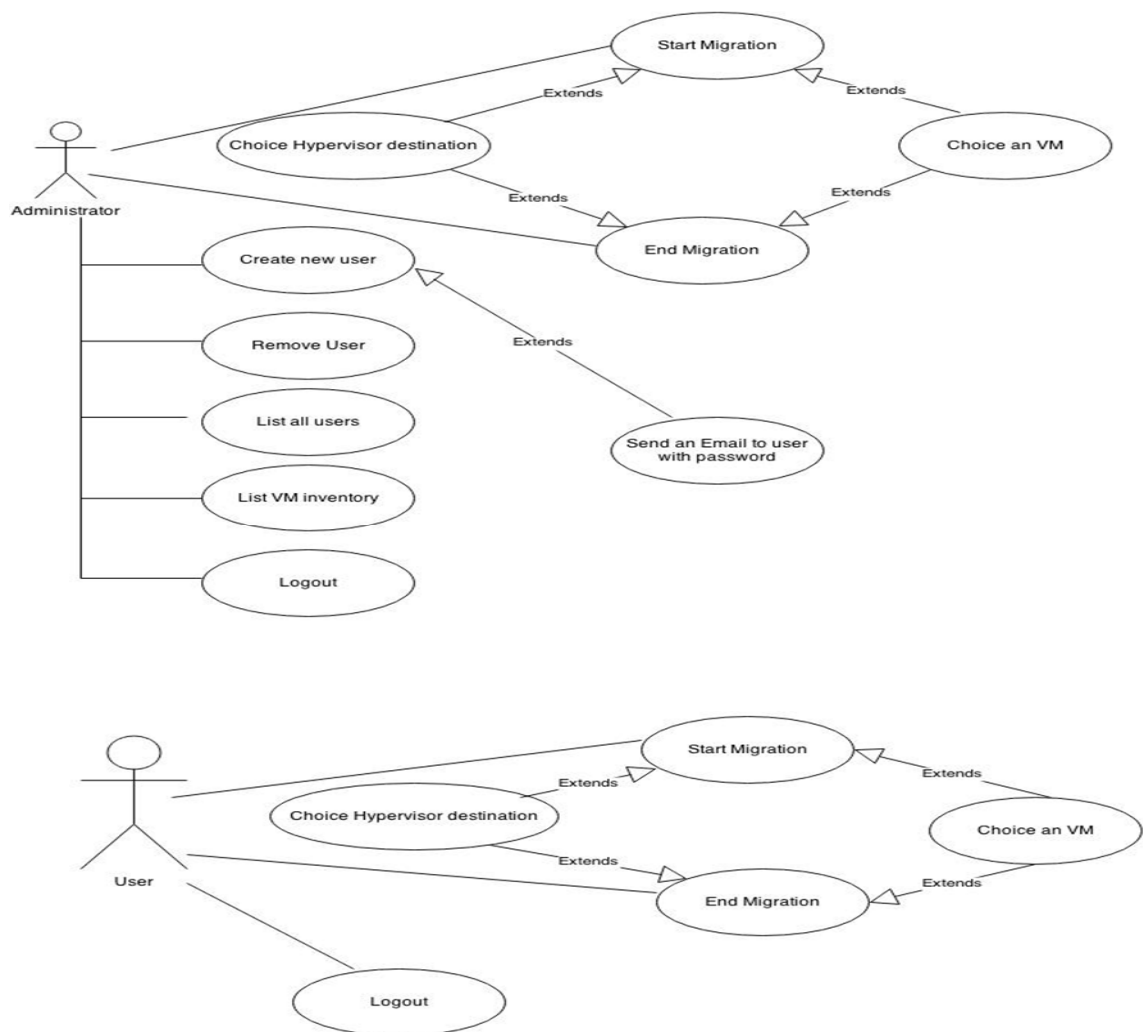


Figura 9- VirtualMigra Use case Diagram.

3.4. The Relational Structured Database

A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.

3.4.1. Entity Relationship Model

In software engineering an entity relationship model (ER) is a representation of abstract and conceptual data. In other words the entity relationship model is a high-level data model composed by entities and entities-relationships that is useful in development of a conceptual design for a database. When a relational database needs to be designed it is necessary to construct and draw an ER before the beginning of application development process. Thus this model provides an abstraction of operation principles of application.

Our model is presented in Appendix 1.

4. VirtualMigra Technologies

This chapter outlines a description of the VirtualMigra platform, resources, architectures and detailed description of the different tools and development technologies.

4.1. Resources

A number of resources were used to develop and support the platform. In the following sub-sections we present these resources.

4.1.1. Windows Server 2008 R2

Windows Server 2008 R2 is a server operating system manufactured by Microsoft. It was released on July 22, 2009 and launched on October 22, 2009. This operating system was built on top of Windows NT 6.1, the same core operating system used with the client-oriented Windows 7.

This version includes new function for active directory management (this active directory is a structure used on Microsoft Windows computers and servers to store information about, networks users and domains), virtualization tools, and native support for several processors.

Additionally, it also includes a new Internet informational services (IIS) version, version 7 [35].

4.1.2. Internet Information Services 7

The Internet information services is a software included in Windows Server 2008 R2 and other Windows product families, wich provides security-enhanced, easy-to -manage web servers for developing and reliably hosting web applications, and web services. This enables organizations to deliver rich, web-based experiences. Internet information services supports HTTP, HTTPS, FTP, FTPS, SMTP and NNTP [36].

4.1.3. Adobe Dreamweaver 8

Adobe Dreamweaver is a sophisticated Web design app that best serves professionals incorporating multimedia elements and database-driven content into their sites.

Dreamweaver offers its users a comprehensive and dynamic interface for developing sites. Dreamweaver has support for a wide variety of programming language (PHP, HTML, CSS, JAVASCRIPT, ASP) [37].

4.1.4. Relational Database Management System - MySQL

MySQL is a relational database management system (RDBMS) based on structured query language (SQL) released in January 1998 that allows the creation of relational database structures on a Web server, in order to store data or automate procedures. MySQL runs in all platforms, including Linux, Unix, Mac OS, and Windows. It is fully multi-threaded using kernel threads and provides application program interfaces (APIs) for many programming languages, including C#, C, Eiffel, Java, Perl, PHP, Python and TCL.

MySQL is used in a wide range of applications, including data warehouse, e-commerce, e-learning, web databases, login applications and distributed applications. It is also increasingly embedded in third-party software and other technologies.

The software XAMPP can be used to easy deploy and install a MySQL Server, this software is an open source cross-platform Web server solution stack package, consisting of Apache HTTP server, MySQL and a set of common interpreters [38].

4.1.5. VMware OVF Tool

VMware OVF Tool is a command-line utility that enables a user to import and export OVF packages to and from a wide variety of VMware products.

We can use OVF Tool to distribute and import virtual machines and vApps. For example, we can create a virtual machine within VMware vSphere™, and use OVF Tool to export it into an OVF package for installation, either within our organization or for distribution to other organizations. OVF facilitates the use of vApps, which consist of preconfigured virtual machines that package applications with the operating system that they require [39].

4.1.6. Oracle VBoxManage Tool.

VBoxManage is a command-line interface to VirtualBox. With it, we can completely control VirtualBox from the command line of the host operating system. VBoxManage supports all the features that the graphical user interface gives us access to, but it supports a lot more than that. It exposes really all the features of the virtualization engine, even those that cannot be accessed from the GUI [40].

4.1.7. PHP 5

PHP [41], invented by Rasmus Lerdorf in 1994, is a dynamic object-oriented language focused on server-side application development. It is now one of the most popular languages, as of October 2013 ranking 5th on the TIOBE programming community index [42], used by 78.8 percent of all websites whose server-side language can be determined [43], and ranking as the 4th most popular language on GitHub by repositories created in 2013 (as of October 27) [44]. This popularity has led to the creation of a number of large, widely-used open source applications and application frameworks, including WordPress [45], Joomla [46], Drupal [47], MediaWiki [48], Symfony [49], Magento [50], CodeIgniter [51], and has made it a popular choice for developers creating new web applications and frameworks.

The availability of such large, open-source systems provides an ideal ecosystem for empirical software engineering research. PHP is also an important target for program analysis research. Most PHP applications are web-based, giving an urgency to analyses focused on detecting potential security errors. At the same time, the dynamic nature of the language (e.g., duck typing, reflection, dynamic inclusion of source files, runtime construction and evaluation of code), as well as its use on larger and larger systems, increases the importance of analyses targeted at program understanding, automated refactoring, and programmer tool support, all areas where PHP currently lags behind languages such as Java.

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, IIS, and many others. And this includes any web server that can utilize the FastCGI PHP binary, like lighttpd and nginx. PHP works as either a module, or as a CGI processor [52].

4.1.8. PERL 5

Short for Practical Extraction and Report Language, Perl is a programming language developed by Larry Wall, especially designed for processing text. Because of its strong text processing abilities, Perl has become one of the most popular languages for writing CGI scripts. Perl is an interpretive language, which makes it easy to build and test simple programs [53].

Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system

administration, web development, network programming, GUI development, and more.

The language is intended to be practical rather than beautiful. Its major features are that it's easy to use, supports both procedural and object-oriented (OO) programming, has powerful built-in support for text processing, and has one of the world's most impressive collections of third-party modules [54].

4.1.9. SOAP Web Services

As a new distributed computing technology, Web services are more and more often used for interoperation among heterogeneous platforms [55]. Web Services provide a conceptual foundation and technology infrastructure for service-oriented computing. It allows program written in different languages on different platforms to communicate with each other in a standard-based way. They are considered as reusable software components over the Internet, not limited to current environments, instead they can be deployed into every software system that is Web-service-aware [56].

The world wide web consortium (W3C) defines Web service as a software system designed to support interoperable machine -to-machine interaction over a network. It has an interface description in a machine processable format and those others systems interact with the Web service in a manner prescribed by its description using SOAP- based Web services or REST-based Web services [57]. Simple object access protocol (SOAP) is a protocol specification for exchanging data within a network. This type of Web service have little relation to the web, where applications access and invoke the functional components located on remote machines through the APIs over the network, where they also convert data from platform-specific formats into XML form.

4.2. Server and Client Architectures

In terms of hardware architecture, we have asset important parts that enables a better management of different environments.

Considering this architecture, the VirtualMigra Platform implement the “cold” migration technique and client devices can migrate their virtual machines across different Type 2 hypervisors. The Figure 10 shows the architecture used. This architecture was conceived and studied as the most efficient in terms of resources, services and performance for all machines according to the characteristics of the same avoiding a sharp overload and possible bottleneck in the network. This architecture was built for all processing and storage is done on a server, facilitating the migration and the storage of large files like the files of Virtual Machines.

Given that the VMware Workstation API can only be used remotely being in “Server Mode” and taking account of systems pedagogy , it makes no sense each remote user have VMware Workstation in Server Mode without being able to enjoy the full potential of the software due to the performance of a normal PC with this type of Technologies. Furthermore, if a Workstation user want to change the password in its computer, VirtualMigra which did not allow the connection because it needs that to connect.

Some potential of VMware Workstation being in “Server Mode” are:

- Datastore creation for VMs and the handling of files with large size,
- Remote access the VMware Workstation Server by any client with remote VMware Workstation host installed.
- Roles and permission for users who have created an account on the machine where VMware Workstation Server is installed.

With this architectures, all users account can be manage by the Administrator Server who can be also the Administrator of the VirtualMigra

Platform. Only users Server can be Administrator of the VirtualMigra Platform.

We decided to install VMware Server Workstation on the Server allowing all remote VMware users will be able to connect to the VMware Workstation Server taking advantage of the capabilities of the Server and VMware Workstation Server.

Furthermore, the VMware Workstation API does not allow manipulate some features of the VMware Workstation (in Server Mode) like roles and permissions users and other features.

All the “shared processing” is done on the Server with a bigger storage and memory capacity. Through the VirtualMigra Platform, the users Server can then migrate Virtual Machines of VirtualBox users through the VirtualMigra Platform and also manage the Virtual Machines of VMware Users with roles e permissions and other features.

On the other hand, accessing to the VirtualMigra Platform remote VirtualBox users can send and receive Virtualbox machines between them and VMware Workstation users as show in Figure 10, 11 and 12.

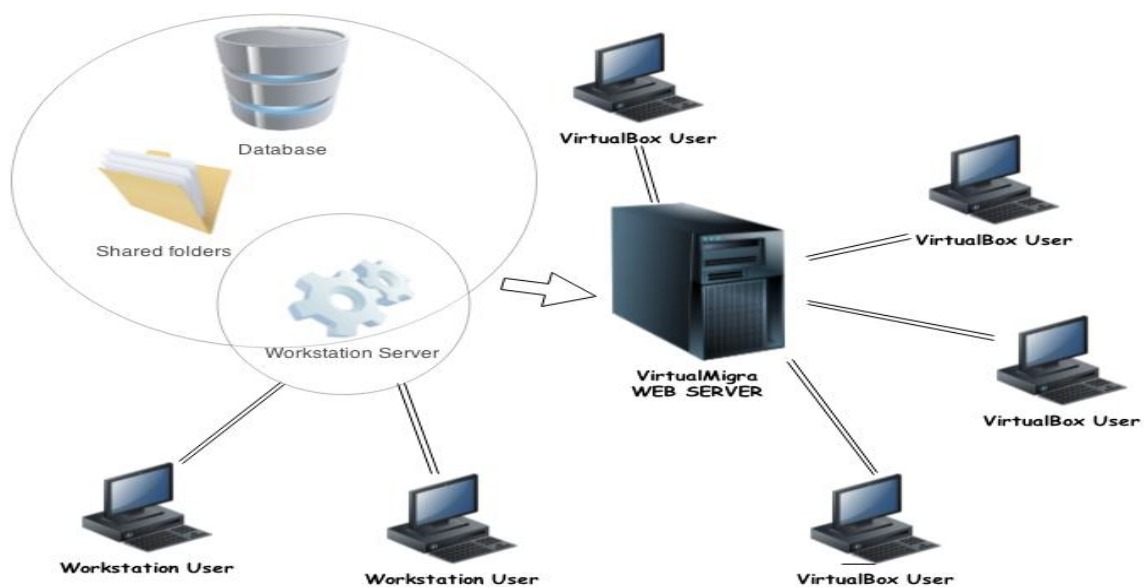


Figura 10- Server and client Architecture.

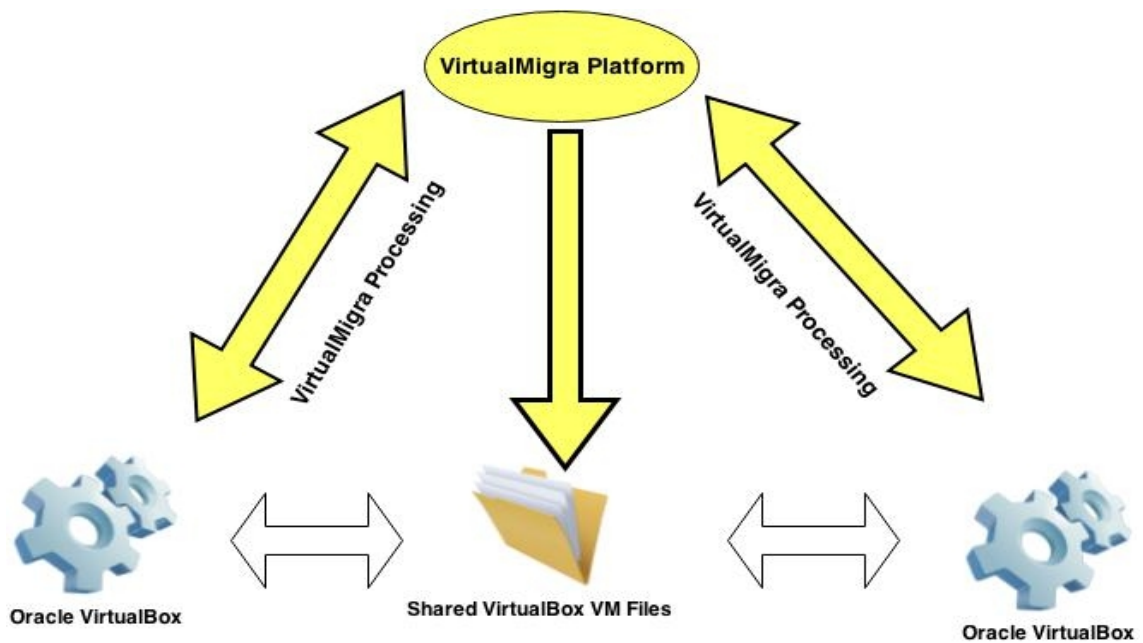


Figura 11- VirtualBox to VirtualBox Migration.

VirtualBox Virtual Machines files differs from VMware Virtual Machines files such as configuration files format, virtual disks format and other features.

The VirtualMigra Platform use “Oracle VboxManage” and “VMware OVF Tool” for create and convert OVF Files as show in Figure 12.

Using the same way, the VirtualBox OVF Files differs from a VMware OVF Files and the VirtualMigra Platform need to process and set the different parameters of the different OVF Files for migration succeed as show in Figure 13.

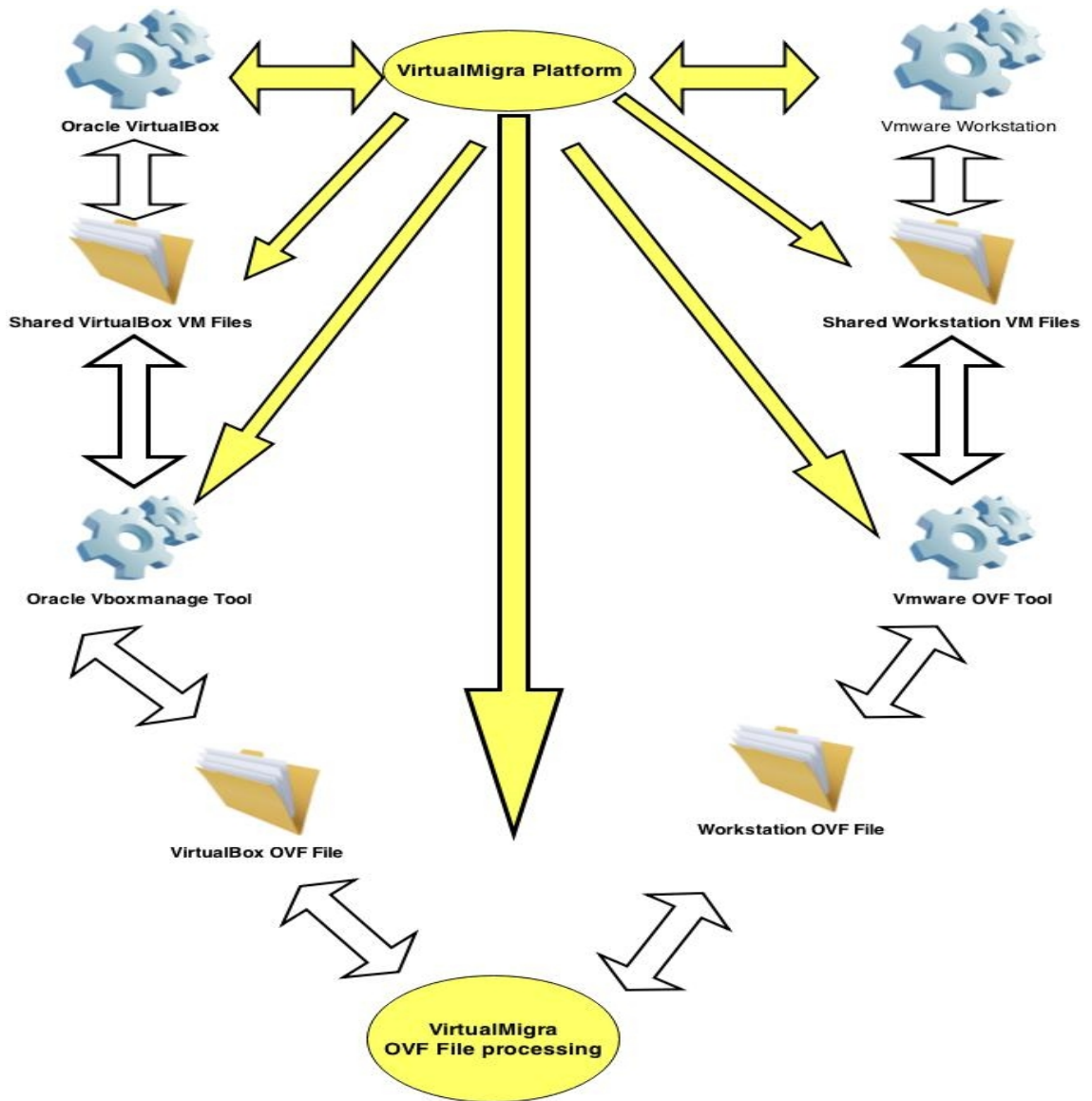


Figure 12- Workstation to VirtualBox Migration and vice versa.

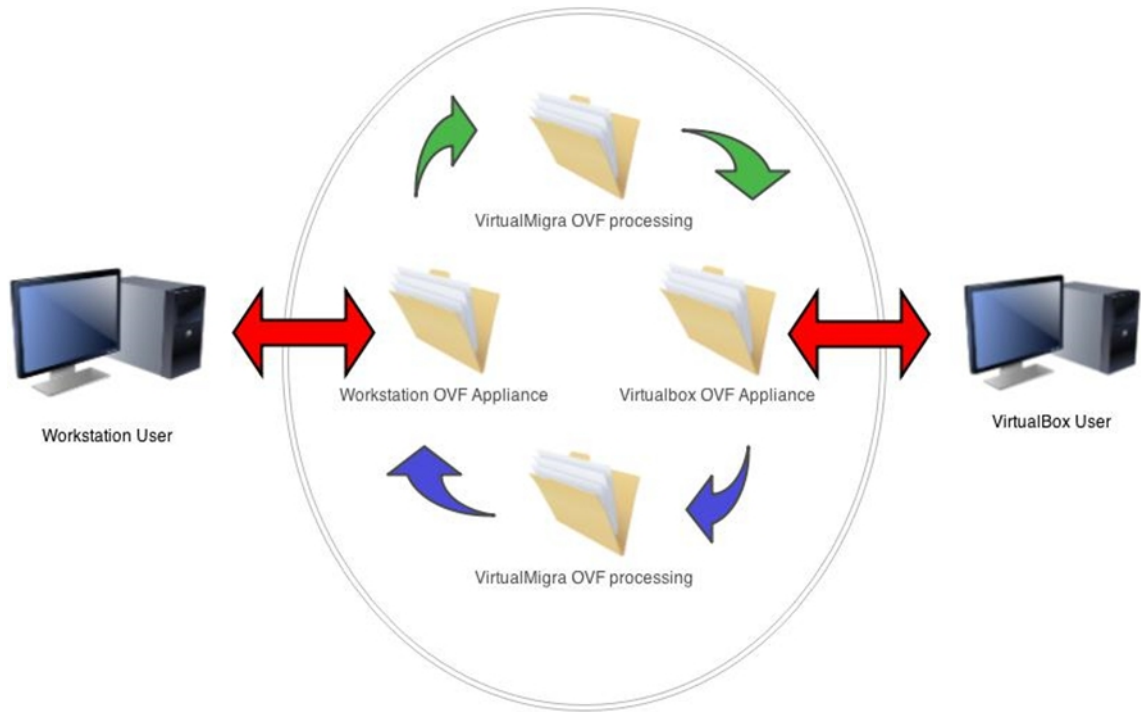


Figura 13- VirtualMigra OVF files processing

5. System Demonstration and Validation.

This chapter outlines the demonstration and validation of the platform, composed by the demonstration of the main interface and its elements.

5.1. VirtualMigra Platform Usage

In this subsection we present a set of user interfaces, wich we describe as platform usage.

These user interfaces were designed to consume current Virtualbox and Worksation APIs, services that exists.

In the following paragraphs we describe every element presented in the VirtualMigra user interface main environment.

The authentication administrator and user interface component is show in Figure 14 and 23. This element enables the possibility to authenticate users and provide an access to different working environments.

The desktop user interface element appears after the authentication has been succeed and aims to present a migration environment as is show in Figure 15 and 26.

5.1.1. VirtualMigra Administrator Interface Environment

The users who belong to the administrator group can manage users, add users, remove users, list users, list virtual machine inventory, start a migration and end a migration. The Administrator management interfaces are show in Figure 14, 15, 16, 17, 18, 19, 20, 21 and 22.



VirtualMigra

Welcome to the administrator login of VIRTUALMIGRA !

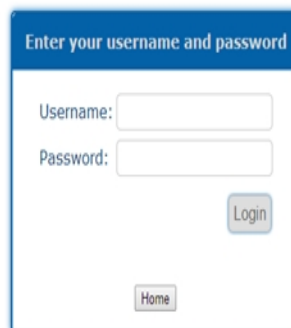


Figura 14- Administrator login page

VirtualMigra

Welcome **admin** to the administrator page of VIRTUALMIGRA !

Choice an action !

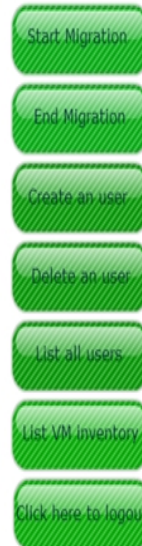


Figura 15- Administrator management page.



VirtualMigra

Welcome **admin** to the administrator page of VIRTUALMIGRA !

Create an user !

Enter the data

Name:

Surname:

Username:

User type:
(admin/user)

E-mail:

IP Address:

Info:

[Home](#) | [Click here to logout !](#)

Figura 16- User creation page



VirtualMigra

Welcome **admin** to the administrator page of VIRTUALMIGRA !

User creation is success !

Enter the data, again !

Nome:

Sobrenome:

Username:

User type:
(admin/user)

E-mail:

IP Address:

Info:

Figura 17- User creation success page

The screenshot shows a Gmail interface with a confirmation email from "World Wide Web Server" received on August 27th. The email subject is "Confirmação de cadastro - virtualmigra.com". The body of the email reads: "Prezado Didier Ribeiro, O registo na plataforma do VirtualMigra foi feita com sucesso, www.teusite.com/". It also lists the user details: "Usuario: Senha: Obrigado! Webmaster". A note at the bottom states: "Esta é uma mensagem automática, por favor não responda!". The interface includes a search bar, navigation buttons, and a sidebar with folders like "Boîte de réception (3 483)", "Messages suivis", "Important", "Messages envoyés", "Brouillons (43)", and "Cercles".

Figura 18- E-Mail User Password.



VirtualMigra

Welcome **admin** to the administrator page of VIRTUALMIGRA

Administrator List !

NAME	SURNAME	USERNAME	INFO	EMAIL	IP ADDRESS	DATE OF REGISTRATION	DATE OF LAST LOGIN
Didier	Ribeiro	admin	Administrador	didier.4mypeople@gma	0.0.0.0	2014-10-05 19:52:52	2014-10-05 20:42:37

User List !

NAME	SURNAME	USERNAME	INFO	EMAIL	IP ADDRESS	DATE OF REGISTRATION	DATE OF LAST LOGIN
Rui	Salvado	u001	Utilizador do departamento de RH	rui.meupessoal@gmail	10.0.0.6.162	2014-10-05 20:38:26	
Tomas	Costa	u003	Estagiario	tomas.meupessoal@gma	10.0.0.6.198	2014-10-05 20:35:41	
Julia	Gomes	u004	Estagiario	julia.meupessoal@gma	10.0.0.6.199	2014-10-05 20:31:29	

[Home](#)

[Click here to logout](#)

Figura 19- Users list page



VirtualMigra

>

Welcome **admin** to the administrator page of VIRTUALMIGRA !

Delete an user !

Enter the username

Username:

[Delete](#)

[Home](#)

[Click here to logout !](#)

Figura 20- User delete page.



VirtualMigra

Welcome **admin** to the administrator page of VIRTUALMIGRA !

User delete is success !

Enter the data

Username:

Delete

Home

Click here to logout

Figura 21- User delete success Page.



VirtualMigra

Welcome **admin** to the administrator page of VIRTUALMIGRA !

Select one Virtual Machine

Select the hypervisor destination

Select the hypervisor destination

Oracle Virtualbox

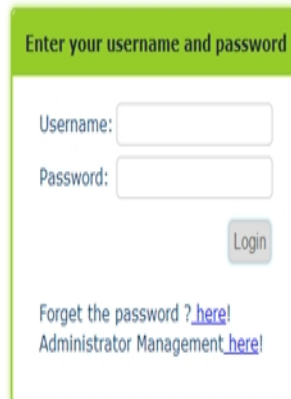
OK

home

Figura 22- Administrator start migration page.

5.1.2 VirtualMigra User Interface Environment

A normal user who belong to user group can indicate the hypervisor destination and after, choice the virtual machine who want start or conclude a migration. The User management interfaces are show in Figure 23, 24, 25, 26 and 27.



The image shows a user login page. It has a green header bar with the text "Enter your username and password". Below the header bar are two input fields: "Username:" and "Password:". To the right of the "Password:" field is a "Login" button. Below the input fields are two links: "Forget the password ? [here!](#)" and "Administrator Management [here!](#)".

Figura 23- User login page.



VirtualMigra

Welcome to the password recovery of VIRTUALMIGRA !

Enter your e-mail address

E-mail:

Create password

Home

Figura 24- Password recovery page.

You've been logged out!



VirtualMigra

Welcome to VIRTUALMIGRA !

here!' and 'Administrator Management [here!](#)'."/>

Enter your username and password

Username:

Password:

Login

Forget the password ? [here!](#)
Administrator Management [here!](#)

Figura 25- Logout page.



VirtualMigra

Welcome **u001** to the user page of VIRTUALMIGRA !

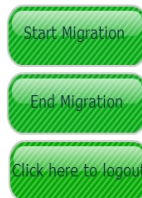


Figura 26- User Migration Management page.



VirtualMigra

Welcome **u001** to the user page of VIRTUALMIGRA !

Select one Virtual Machine Select the hypervisor destination

Select the hypervisor destination

- Oracle Virtualbox
- Vmware Workstation

Figura 27- User start migration page.

6. Conclusions and Future Work

This dissertation described the proposed VirtualMigra Platform and the creation, extension, and inclusion of a number of services into the platform. In addition, this dissertation described the work performed to create a solution to support the development of virtual environments.

This solution was proposed on the sense of improve virtual systems encountered in the related literature about the topic.

Chapter one presented the focus of the dissertation including a perspective of the cloud computing topic and related issues. The objectives and main contributions were presented, and the document organization was also included. Chapter two addressed the related work about cloud computing, hypervisors, virtualization, and virtual machine migration. This chapter introduced the origins and purpose of the virtualization and, therefore, the concept of cloud computing. The evolution of systems virtualization, its importance and related problems in the literature were discussed. A set of definition applied to a number of virtualization techniques and tools, including related concepts like Virtual Machines files and Virtual Machines migration were identified.

Additionally, a number of commercial platforms were identified in order to include a discussion about main features of virtualization solutions. Finally, the concept of the Open Virtualization Format, problems, proprietary platforms, and Web Services were included.

Chapter tree explored the requirements analysis to determinate the features and interactions of the proposed system. This chapter details the use cases for a range of proposed VirtualMigra Services, Oracle VirtualBox and VMware Workstation APIs and SDKs.

Chapter four introduced the VirtualMigra platform technologies and a set features of the VirtualMigra Platform. Additionally, the database ER

diagram and the class diagram of the core components of the system were included. Finally, used technologies, architectures and exposure details of the concept were presented.

Chapter five presented the demonstration and validation of the platform, interfaces and resources. The platform usage, namely the VirtualMigra users interfaces and management interface were demonstrated.

A virtual machine is a software computer that, like a physical machine, runs an operating system and applications. A virtual machine uses the physical resources of the physical machine on which it runs, which is called the host system. Virtual machines have virtual devices that provide the same functionality as physical hardware, but with the additional benefits of portability, manageability, and security.

All virtual machines have a hardware version. The virtual machine hardware compatibility setting determines the hardware features of the virtual machine.

For virtual machines running mostly office and Internet productivity applications, using multiple virtual processors is not beneficial. For server workloads and data-intensive computing applications, adding extra virtual processors may provide an increase in application performance.

The Memory management limitations on operating systems cause virtual machine memory to overcommit, which severely affects system performance. The total amount of memory that we can assign to all virtual machines running on a single host machine is limited only by the amount of RAM on the host machine.

We can use virtual machines to debug kernel code on one system without the need for two physical computers, a modem, or a serial cable. We can use a Debugging Tools for Windows (WinDbg) or a Kernel Debugger (KD) to debug kernel code in a virtual machine over a virtual serial port.

The migration of virtual machine between same manufacturer or not is not so straightforward how we think and furthermore the VirtualMigra Platform have the design and development to provide a best interactions with this type of technologies.

Finally, this platform helps users and developers to construct a group of must-have administration and testing tools. Experiments were driven with success and it is ready for real use.

As a suggestion for future work the current version of the VirtualMigra Platform should be improved with more controls over Virtual Machine. Additionally, the design and development of a set of migration services in order to provide interactions with other hypervisors (Microsoft Hyper-V, VMware VSphere, VMware VCloud), cloud storage platforms and work with other VMware OVF appliance is suggested. Finally, VirtualMigra can be use like a web tool for others external platform using a set of VirtualMigra services may be proposed.

References

[1] Peter Mell, Timothy Grance. "The NIST Definition of Cloud Computing Computer Security Division , National Institute of Standards and Technology" , Gaithersburg, 2 0 1 1 .

[2] Dacentec. "What is a Cloud Datacenter". 2011.

<http://www.dacentec.com/What+is+a+Cloud+Datacenter> (accessed 2014).

[3] O. Brian, T. Brunschwiler, H. Christ, B. Falsafi, M. Fischer, S. G. Grivas, C.

Giovanoli, R. E. Gisi, R. Gutmann, M. Kaiserswerth, M. Kundig, S. Leinen, W.

Muller, D. Oesch, M. Redli, D. Rey, R. Riedl, A. Schar, A. Spichiger, U. Widmer, A. Wiggins, M. Zollinger, M. Kaiserswerth. "Cloud Computing". White paper, Swiss Academy of Engineering Sciences , Seidengasse : Society of American Travel Writers SATW, 2013.

[4] R. Bradford, E. Kotsovinos, A. Feldmann, H. Scioberg. "Live Wide-Area Migration of Virtual Machines Including Local Persistent State". Deutsche Telekom Laboratories, Berlin: Virtual Execution Environments VEE'07, 2007.

[5] Bhaskar Prasad Rimal, Eunmi Choi, "A taxonomy and survey of cloud computing systems", 2009 Fifth International Joint Conference on INC, IMS and IDC, published by IEEE Computer Society.

[6] Ramgovind S, Eloff MM, Smith E, "The management of security in cloud computing", IEEE - 2010.

[7] P.Pocatilu, F.Alecu, M.Vetrici " Measuring the Efficiency of Cloud Computing for E-learning Systems".

[8] Anas BELAHCEN “C-MADAR Learning: Cloud based learning Environment”
Anas BELAHCEN Ecole Nationale supérieure d’informatique et d’analyse des
systèmes (ENSIAS), équipe de recherche LeRMA Mohammed V Souissi
University Rabat, Morocco.

[9] “Using Virtualization to Reduce the Cost of Test”. Casey Weltzin
Product Marke National Instruments Austin, TX USA 2009.

[10] L. van Doorn and R. Sailer, "Virtualization and Hardware-Based
Security", IEEE Security & Privacy, vol. 6, pp. 24-31, 2008.

[11] M. Schwartz. (2012). New Virtualization Vulnerability Allows Escape to
Hypervisor Attacks. Available:
<http://www.informationweek.com/news/security/vulnerabilities/227001055>

[12] G. Heiser, “The Role of Virtualization in Embedded Systems”, First
Workshop on Isolation and Integration in Embedded systems (IIES ´08), April
2008.

[13] “Pocket Hypervisors: Opportunities and Challenges”. Landon P. Cox
Peter M. Chen Duke University University of Michigan, 2007.

[14] “A summary of virtualization techniques”.Fernando Rodriguez-Haro,
Felix Freitag, Leandro Navarro, Efraín Hernandez-Sancheza, Nicandro
Farías-Mendoza, Juan Antonio Guerrero-Ibaneza, Apolinar Gonzalez-Potesa
.University of Colima, México, Faculty of Engineering Technical University
of Catalonia (UPC), Computer Architecture Department, Spain.2012.

[15] C. D. Graziano, “A performance analysis of xen and kvm hypervisors
for hosting the xen worlds project”, Master’s thesis, Iowa State University,
2011.

[16] “Teaching Hypervisor Design, Implementation, and Control to
Undergraduate Computer Science and Computer Engineering Students”
Yoginder S. Dandass Mississippi State University 2012.

[17] <http://www.oracle.com/us/technologies/virtualization/oraclevm/oracle-vm-virtualbox-ds-1655169.pdf>

[18] Oracle Corporation. Oracle VM Virtual Box User Manual. Oracle, 2012.

[19] http://www.VMware.com/pdf/ws10_manual.pdf

[20] <http://www.VMware.com/pdf/desktop/ws10-getting-started.pdf>

[21] <http://pubs.VMware.com/workstation10/index.jsp?topic=%2Fcom.VMware.ws.using.doc%2FGUID-CACF4B1E-4611-4CAD-9F3A-110EBB141B81.html>

[22] <http://pubs.VMware.com/workstation10/index.jsp#com.VMware.ws.using.doc/GUID-C396AB3A-5BF2-4A97-AC2C-12C76FD64C72.html>

[23] <http://pubs.VMware.com/workstation10/index.jsp#com.VMware.ws.using.doc/GUID-D140D21C-2A9C-457C-9B50-FEB941D4DBC5.html>

[24] <http://pubs.VMware.com/workstation10/index.jsp#com.VMware.ws.using.doc/GUID-12B7DF5F-E9B6-4963-858E-504584122023.html>

[25] Anja Strunk, "Costs of virtual Machine Live Migration:A Survey". IEEE Eighth World Congress on Services,2012.

[26] P. Getzi Jeba Leelipushpam ,Dr. J Sharmila, "Live Vm Migration techniques in cloud environment-A Survey",Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).

[27] "Virtualization MANagement (VMAN)". Initiative DMTF Standards for Virtualization Management,2010.

[28] "Bottom-up harmonisation of management attributes describing hypervisors and virtual machines". Vitalian A. Danciu Munich Network Management (MNM) Team Institut fur Informatik Ludwig-Maximilians-Universitat Munchen, 2011.

[29] Open Virtualization Format Specification, DMTF 2010.












[30] OVF 2.0 F.A.Q . DMTF, 2013.

- [31] <http://download.virtualbox.org/virtualbox/SDKRef.pdf>
- [32] <https://www.virtualbox.org/sdkref/>
- [33] <https://www.virtualbox.org/sdkref/annotated.html>
- [34] https://www.VMware.com/support/developer/vixapi/vix11_reference/intro.html
- [35] [http://technet.microsoft.com/en-us/library/dd349801\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/dd349801(v=ws.10).aspx)
- [36] <http://www.iis.net/learn/get-started/introduction-to-iis/introduction-to-iis-architecture>.
- [37] <https://www.adobe.com/support/documentation/en/dreamweaver/documentation.html>
- [38] <http://dev.mysql.com/doc/refman/5.7/en/>
- [39] <https://www.VMware.com/support/developer/ovf/ovf350/ovftool-350-userguide.pdf>
- [40] <https://www.virtualbox.org/manual/ch08.html>
- [41] <http://www.php.net>
- [42] <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [43] <http://w3techs.com/technologies/details/pl-php/all/all>
- [44] The query for this is based on <http://adambard.com/blog/top-github-languages-for-2013-so-far/>.
- [45] <http://wordpress.org/>
- [46] <http://www.joomla.org/>
- [47] <http://drupal.org/>
- [48] <http://www.mediawiki.org/wiki/MediaWiki>








-
- [49] <http://symfony.com/>
- [50] <http://www.magentocommerce.com/>
- [51] <http://ellislab.com/codeigniter>
- [52] <http://www.php.net>
- [53] <http://www.webopedia.com/TERM/P/Perl.html>
- [54] <http://perldoc.perl.org/perlintro.html>
- [55] X. Feng, J. Shen and Y.Fan, “REST: An alternative to RPC for Web services architecture”, in 2009 First International Conference on Future Information Networks (ICFIN), Beijing, China, October 14-17, 2009, pp.7-10.
- [56] J.P. Thomas, M. Thomas, G. Ghinea, “Modeling of Web services flow”, in IEEE International Conference on E-Commerce (CEC), Newport Beach, USA, June 24-27, 2003, pp. 391-398.
- [57] D.A. Menasce, “Composing Web Services: A QoS View”, in IEEE Internet Computing, vol. 8 (6), pp. 88-90, November 2004.





Appendix 1

Entity Relationship Data Model

users	
	userid: INT
	name: VARCHAR(20)
	surname: VARCHAR(20)
	email: VARCHAR(45)
	username: VARCHAR(20)
	senha: VARCHAR(20)
	usertype: VARCHAR(20)
	info: VARCHAR(255)
	registerdate: DATETIME
	lastlogindate: DATETIME
	ipadress: VARCHAR(45)

1:N

pathvm	
	idvm: INT
	users_userid: INT (FK)
	username: VARCHAR(20)
	path: VARCHAR(45)
	migrationdate: DATETIME
	<i>caminhovm_FKIndex1</i>
	users_userid

vminventory	
	idvminvent: INT
	username: VARCHAR(45)
	path: VARCHAR(45)
	migrationdate: DATETIME
