Proclamation

I declare that this thesis was made by myself with assistance of my supervisor. All parts taken over word by word from literature or other publications are referenced and identified. I approve publishing this thesis or any part of it with referencing author of original text.

In Prague at 2014-11-11 ....

Abstract

This master thesis deals with testing data networks in cloud environment. Techniques of inter and intra-cloud networks are described in theoretical part as well as virtual machine migrations. Practical part brings methodology and framework for testing virtual machine migration. Measurements are performed at OpenNebula cloud environment with KVM virtual machines.

Contents

Introduction

Theoretical part

2.1 Virtualization

Virtualization is, in my opinion, the most important technology in data centers, because it caused significant progress in this field. It is not technology itself, so it should rather be called model than technology.

Definition of virtualization as stated in [1] says that "virtualization is a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications or end users interact with those resources. The concept of virtualization is very broad and can be applied to devices, servers, operating systems, applications and even networks." This definition gives description of the virtualization and can be applied to any type of virtualization.

The most common approach is virtualization of computers, because it is the oldest one and most widely used there days. It started in 1960s with mainframes as an attempt to employ resource sharing and this idea is still alive in current time. Virtual computer is logical representation of computer in software. [1] Virtual computers are usually called virtual machines (VM) and physical machine hosting VMs is called hypervisor. Rigorous term for physical hosting machine is host and hypervisor is software performing the virtualization, but word hypervisor is widely used in technical text for machine as well. It is possible and very advantageous to host many virtual machines on single physical computer, because it brings technical and economical benefits. Decoupling computer and it's software from hardware is important advantage, because it brings additional level of abstraction and allows you to shift virtual machines between hypervisors. Economical benefit is quite obvious, since it is not necessary to buy single physical server for single service and electricity saving are also appreciable.

Another type of virtualization is virtualization of networks. It is usually used together with computer virtualization, since it gives an occasion to separate network devices from network itself. Physical machines are not as flexible as VMs are, so plugging them into virtual network is not as beneficial as VMs, because there are still physical network cables, that can be hardly virtualized. There is a hot topic called Software Defined Networking (SDN) having potential to provide virtualization info physical network infrastructure, thus it may be good idea to integrate physical machines into virtual network as well.

Storage virtualization should also be taken into account, because it provides abstraction of the storage. Typical unvirtualized storage uses some physical device for storing data and metadata, but this approach is not enough flexible since it is usually limited to just one physical machine or group of machines connected to shared storage. It is necessary to find any method of storage virtualization, which would be able to connect any storage to any physical of virtual computer.

Service virtualization, memory virtualization, I/O virtualization or database virtualization are another types of virtualization. It is not necessary to mention all the types of virtualization since it is possible to virtualize almost everything and emerging of new types is quite probable.

Term virtualization is going to be used in further text as computer virtualization, another types of virtualization will always be denoted.

2.1.1 Types of virtualization

There are three different virtualization types and they vary by method used t add virtualization layer between guests and physical hardware. It it not possible to easily choose better or worse virtualization types, because it depends on intended usage, character of computing tasks and required operating system.

Architectures of computers, especially x86, are designed to run on physical devices, thus is in not easy to virtualize them. Access to hardware is controlled by priority levels called rings. Lowest priority is used by userspace applications and highest priority (ring 0) is reserved for operating system. It is necessary to insert virtualization layer between operating system and hardware, but there is not any ring with higher priority than operating system uses. This problem needs to be solved and it is not only one challenge. There are sensitive instructions incompatible with virtualization, because they use different semantics when they are not run in ring 0, as mentioned in [2].

Paravirtualization

Paravirtualization is type of virtualization with necessity of modifications in guest kernel. Modifications of kernel are necessary, because operating system uses non-virtualizable instructions that are trying to gain direct access to the hardware. These instruction need to be replaced with hypercalls that communicate directly with virtualization layer of hypervisor. [2] It is obvious, that guest operating system knows it is running virtualized.

Biggest advantage of paravirtualization is lower overhead compared to other types, because it is not necessary to translate instructions before running. However this advantages becomes less significant during time since there are already available processors optimized to run hardware assisted virtualization with less overhead. Main drawback of this type of virtualization is need for modifications done at an operating system, which is not always possible or allowed. Running modified OS also brings additional administration and thus additional cost.

It is possible to take a different look at paravirtualization and do not try to create entire virtual machine, but use operating system-level virtualization, where kernel allows to run multiple userspaces. These userspaces are called containers and therefore this approach is sometimes called container virtualization. It does not provide entire isolated virtual machine, but allow you to run software packed in container. It is advantageous because there is almost none overhead in running software from container while maintaining sufficient level of container isolation. Container virtualization is applicable for situation, where whole virtual machine is not needed and then brings huge performance improvements since operating system layer is shared. Some says, that containers are going to bring next revolution info virtualization. For example Dustin Kirkland, Cloud Solutions Product Manager at Canonical wrote: "Linux containers, repositories of popular base images, snapshots using modern copy-on-write filesystem features. Brilliant, yet so simple. Docker.io for the win!" [3]. I think, that container virtualization may brings compelling advantages and I also like using it, but it is not suitable for every situation. It is still technically kind of paravirtualization and thus it is limited to provide only additional layer on host's operating system.

Full virtualization

Virtualization type capable or running unmodified operating system is called full virtualization. It utilizes runtime translation, which captures non-virtualizable commands and emulates them using hypervisor virtualization layer. Virtualizable instructions are executed directly on the hardware. Modification of "problematic" calls is carried by the hypervisor and it is the main difference compared with paravirtualization.

Most important benefit of full virtualization is it's ability to run guest operating system without any changes, so guest OS is not aware of being virtualized. This makes guest operating system fully abstracted from underlaying hardware, it is possible to multiple different operating system on single host and provides simple migration from physical to virtual machine. Drawback of this type is overhead caused by catching and translating non-virtualized calls.

Hardware assisted virtualization

Full virtualization has significant overhead caused by binary translation, so CPU vendors introduced technologies capable of inserting virtualization layer between ring 0 and physical hardware. It speeds-up trap of privileged and sensitive calls to the hypervisor and it is not necessary to perform binary translation of to modificate kernel of guest operating system.

Benefit of this type is quite obvious, because it lowers virtualization overhead and thus provides better performance compared with full virtualization together with elimination of need for guest kernel modifications compared with paravirtualization. It is necessary to have a support in host's CPU is primary drawback of this type, but there is support in almost every processor in the marker.

Running unmodified guest operating system leaves all necessary translations of instructions on hypervisor layer, so I would be good to to introduce small changes to guest's operating system, which will reduce work left for the hypervisor but also do not need any significant changes in guest's kernel. This approach is called hybrid virtualization and it is subset of hardware assisted virtualization. Installation of additional drivers is required, but it is not necessary to apply any changes on whole kernel. These drivers are aware of virtualization and use virtualization layer directly without any translations made by the hypervisor. This method increases driver's IOPS and therefore it is usually used for virtualized network cards and storages. Driver able to deliver hybrid virtualization is *virtio* for KVM, Xen call it *paravirtualized device drivers* and VMWare *Guest Tools*.

Table 2.1.1: Comparison of virtualization types

Type	**method**	**guest modif.**	**usage**
Paravirtualization	hypercalls by guest kernel	yes	same workloads and same OS
Full	translation of instructions	no	when full abstraction is needed
Hardware assisted	translation with help of hardware	no	same as full, but with compatible CPU
Hybrid	translations and driver changes	driver only	when possible to install additional drives

Summary on types of virtualization

2.1.2 Levels of virtualization

Cloud based systems depends on virtualization, as it was already mentioned before, and it is necessary to think about different usages of virtualization. There are tree approaches how to categorize virtualization: service virtualization, computer virtualization and

2.1.3 Advantages of virtualization

2.2 Cloud computing

It is possible find many services called "cloud based" and it is important to agree on accurate definition of these services. It is quite clear, that cloud based service will use principle of cloud computing. Definition of cloud computing by NIST says, that "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) than can be rapidly provisioned and released with minimal management effort or service provide interaction." [4]. This definition clarifies what cloud computing is, but says nothing about parameters and used technologies.

I think, that it would be more convenient to start definition from lower levels, which provides elementary parts, and get to the cloud service afterwards. This definition gives different look at cloud computing than NISTs, but it uses same conditions and therefore results are basically same. It focuses on currently used principles, which may change during time, so it may not be valid after some time, but it provides more technical overview on operation of cloud services.

Cloud computing services are nowadays heavily dependent on virtualization, because it allows to replace physical machines with virtual machines (VMs) and brings a lot more flexibility than physical machine can ever provide.

Basic part of cloud computing system is virtual machine. Physical machine can also be part of the cloud system, but it is not able to deliver required rapid provisioning and it is not possible to deploy physical machine without service provider interaction. Virtual machine is elemental resource and also use some additional resources. These resources can be for example networking, which is used for interconnection between VMs as well as for reaching customers, storage used for system internal or customer data. It is important do employ some configuration management and orchestration, because it is able to deliver rapid provisioning of virtual machines and minimizes effort required for administration.

Virtual machines together provides the service, which is exposed to users via any kind of networking. It doesn't matter whether customers access the service directly at virtual machines or via a proxy, but hiding worker VMs brings additional flexibility for migration and scalability.

Difference between cloud computing and bare virtualization is intelligence included in cloud, because it may be controlled automatically according to events or monitoring observed at cloud system. It is common to supply customers with configuration interface, which allows to tune service parameters and provides user friendly interface for administration. Bare virtualization does not offer any intelligence, even if it is equipped with shiny user interfaces with opportunity to scale virtual machines up or down, because all change performed manually.

2.2.1 Service models

2.2.2 Network in cloud

2.2.3 Storage in cloud

2.2.4 Orchestration software

OpenNebula

2.3 Migration of VMs

2.4 Distributed data center

Practical part

List of Abbreviations

CPU	Central Processing Unit.
IOPS	Input/Output Operations Per Second.
KVM	Kernel-based Virtual Machine.
NIST	National Institure of Standards and Technology.
OS	Operating System.
SDN	Software Defined Networking.
VM	Virtual Machine.

List of Figures

List of Tables

Bibliography

[1] IBM Corporation. Virtualization in education. http://www-07.ibm.com/solutions/in/education/download/Virtualization%20in%20Education.pdf, 2007. [Online; retrieved 2014-09-17].

[2] Chris Horne. Understanding full virtualization, paravirtualization, and hardware assist. http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf. [Online; retrieved 2014-08-20].

[3] Dustin Kirkland. Docker in ubuntu, ubuntu in docker. http://blog.docker.com/2014/04/docker-in-ubuntu-ubuntu-in-docker/. [Online; retrieved 2014-09-20].

[4] T. Mell, P. Grance. The NIST definition of cloud computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. [Online; retrieved 2014-08-17].