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. I approve publishing this thesis or any part of it with referencing author of original text.

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

Abstract

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 gives ability to shift virtual machines between hypervisors. Economical benefit is quite obvious, since it is not necessary to buy single physical server for every service and electricity saving are also appreciable.

Another important 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 flexible enough 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 provide any storage to any physical of virtual computer.

Service virtualization, memory virtualization, I/O virtualization or database vir-

tualization are another types of virtualization. It is not necessary to mention all the types of virtualization because 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 allows 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-virtualizable 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 current 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*.

Summary on types of virtualization

There were presented some virtualization general virtualization types and their pros and cons. There is not any universal virtualization type suitable for all use

cases, thus is is always possible to decide on planned usage. It also depends whether it is required to run different kernel on single physical host or it is sufficient to share one kernel for all containers. Differences are compared in table 2.1.1.

We can divide types into two groups:

- One group provides guests with full virtual machine, every VM uses it's own isolated kernel and VMs are fulll or almost fully decoupled from hardware. Full, hardware assisted and hybrid virtualization belongs to this group.
- Members of second group are containers and paravirtualization. This group is specific by lightweight containers and host kernel shared by all running containers.

Virtualization is massively used even by czech IT companies. First group is used for example by *Wedos* for their virtual server hosting and related services. Second group is uses by *Seznam.cz* and they use LXC for web serves as well as for Hadoop cluster.

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

2.1.2 Advantages of virtualization

Most important advantages is decoupling software from physical hardware, at least in my opinion. It is possible to migrate virtual machines with running services between physical hosts without significant impact on service behavior. This brings amazing opportunity to adapt service environment on demand and scale the service.

It is possible to perform any hardware and software upgrades, because all running services may be temporarily migrated to other physical host. Virtual machines are much more easier to deploy than physical ones. It takes only a few seconds to create and run VM compared to at least hours to deploy physical machine. Deploy of virtual machine do not have to be performed by persons, because it is possible to employ an orchestration and scale up the service (add virtual machines) automatically. Reset of virtual machine is actually just software instruction in hypervisor, so it may by done remotely with ease.

Geographical backups or failover is much more easier to accomplish with virtualization approach. You can rent virtual machine from provider in foreign country

and start your services in a few moments. It is huge simplification compared with running physical machine at foreign data center.

Virtualization brings also some economical and environmental advantages. Economical advantages are quite obvious, because it is no longer necessary to buy physical servers. Non-virtualizational approach requires one physical machine for every running server, but it is not longer necessary with virtualization. It is possible to run many virtual servers or containers on single physical machine. It is also possible to move even to the higher level of CAPEX cutting and rent virtual machine from provider and absolutely eliminate need for running any server machine. Renting virtual server increases OPEX, but they are more flexible and easier to control. Electrical consumption should also be taken into account, because single physical machine, even under higher load, will definitely consume less power compared with two or more similar machines.

I asked Petr Hodač, technical manager at SiliconHill and he stated, that they managed to reduce electricity consumption of whole server room by 19% iter alia due to deployment of virtualization. It produced also additional saving, because they need less UPS batteries and less cooling capacity, but savings on cooling are not included in mentioned savings.

However there are some drawback too. Failure of physical host causes failure of all virtual machines or containers running on this host. It is kind of single point of failure, but we can fight it with duplication of service nodes between different hosts, datacenters, providers or continents. Another disadvantage is hidden in additional virtualization level, since it is necessary to take care of hypervisors. I is not a real disadvantage, since traditional non-virtualized approach needs to take a care of many virtual machines.

Deployment of virtualization should always be well planned, because it can bring many amazing advantages, but it is also able to cause a disaster in case of poor system design or lame administration.

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) or con-

tainers 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 network. 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 them 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.

Cloud computing is kind of hype these days, so it is often used just for marketing purposes and thus it is recommended to perform service analysis and do no absolutely trust every buzzword used in specification.

2.2.1 Deployment models

There are three scenarios possible for deploying cloud solutions. Models differs by ownership and subject responsible by administration of the system. Right solution depends on expected load, available budget as well as on expected classification of data. Public model and private model are mutually contradictory and last model called hybrid is combination of first two mentioned. Model are compared in table 2.2.2.

Private

Private model defines cloud environment build exclusively for single subject. Typical scenario is to build private cloud in datacenter owner by the subject, but it is not strictly required. There is common misunderstanding of term private, because it means private usage of cloud resources and not private ownership of cloud infrastructure. Private cloud may be leased from third-party provider and it also can be running on third-party hardware.

Running private cloud gives an advantage in elimination of any inter-tenant isolation problems and it is possible to adapt configuration to fit owner needs. There is a law, which forces sensitive data to be stored internally and with limited access, so it is necessary to build private or hybrid cloud for this kind of usage. It is not clear how to handle clouds and especially storages with law, because it this topic is wide and it is not possible to define rigid rules. There is currently running case with US judge ordering Microsort to provide data stored in Ireland. [5]

Drawback of private cloud is higher initial cost and probably also higher operational costs, but it depends on expected usage.

Public

Public cloud deployment model is based on resource sharing between the tenants. There is usually one subject called cloud provider and many customers (tenants) and these tenants buy resources and rights to use them. Resources are usually charged according to it's usage.

Billing per resource usage is called pay-per-use and it is interesting method of shifting costs between initial and operational. There are usually plans with various CPU, memory and storage options and final cost depends on real usage of resource. Pricing plan based on pay-per-use is favourable to services with low load with occasional peaks. Service under constant high load is not well suited for this payment model, because it does not bring any benefits. It is also to run scalable application, since unscalable will not be able to

Infrastructure is not dedicated and it is shared between tenants. Resources are shared, but must be strictly isolated, because it is unacceptable to allow any interference between tenants, unless they make an explicit request to allow it.

It is common to provide services with flexible parameters, for example Amazon calls it EC2 - Elastic Compute Cloud. Elasticity of provided services allows tenant to use more resources when needed and fall back to usual amount.

Hybrid

Hybrid cloud is model utilizing both, public and private, previous mentioned models. The goal is to combine advantages of both model and eliminate drawbacks. Public cloud is usually more cost effective, but may not be able to meet the security requirements and on the other hand private cloud can be designed to comply with users requests, but it is expensive. Hybrid designed solution can use private cloud part for confidential data and public cloud for less sensitive ones.

It is also possible to utilize cloud bursting in which system runs in private cloud and delegates part of load into public cloud. Lets describe it with application for collecting votes - sensitive part responsible for counting votes and generating results report will run in private cloud and public report will be saved to and served from infrastructure of public cloud. High level of security of counting votes is guaranteed and application is also able to deliver results to many subscribers as it can scale up into public cloud.

Table 2.2.2: Comparison of deployment models

model	**private**	**public**	**hybrid**
initial cost	higher	lower	medium
operational cost[1]	higher	lower	medium
security	higher	lower	medium
elasticity	lower	high	medium

2.2.2 Service models

Purpose of cloud computing is to deliver the service and provide customers with tools to manage this service. Service models differs by level of control provided to customers and thus with areas of responsibility. I am going to call border between responsibility of customer and responsibility of provider as responsibility border. Responsibility borders according to service models are depicted at figure 2.2.1.

Some of service models leave almost all control of service and responsibility at provider side and other supplies customer with more control. It is necessary to select right service model according to expected service usage and required control level.

Infrastructure as a Service

IaaS is model with the most of configuration tasks left at customer's side. Customer is responsible for virtual machines and it's services, so it gives much more flexibility than other models and it is well-suited for services with extraordinary requirements.

Customer manages virtual machines as well as running services, so provider is responsible only for virtualization and underneath layers. It is even possible to run custom operating system, but provider usually offers prepared images with different operating systems. Prepared OS images are tested and modified to run well in cloud environment. There can be installed hybrid virtualization drivers, kernel tweaked to run virtualized and it is also good idea to remove useless drivers and software.

This model is good choice if special configuration is needed, but service deployment is more difficult because some expertise is required. IaaS can be used if customer require additional level of security, because virtual machine can use crypted volume and make data unreadable for provider. Unfortunately it is still possible for provider to acquire confidential from other sources, for example from memory, but it is much complicated to perform this.

Typical example of IaaS is Amazon Web Services and Active24's "Virtuální Privátní servery".

Platform as a Service

Border of responsibility of PaaS is located two layers higher compared to IaaS. Service provider is responsible for platform and all underlaying layers, thus provider takes care of same layers as in IaaS plus operating system and platform. Leaving operating system maintenance on provider's side may be beneficial, because provider can adjust operating system for virtualization and takes care about software updates.

Provider usually manages a lot of operating systems for many customers, so this updating and maintenance tasks may be automatized or executed in batch. Sharing operating system layer between customers with preserving adequate level of isolation can save many resource and make operating system administration even easier.

Customer using service according to this model runs his own software and does not take care of any lower layers. It is not necessary to do any administration tasks and more effort can be given to application development.

This model is well-suited for running applications without any special requirements. It makes service deployment faster and easier, but is more limited by used platform. Typical example is project Evia and Microsoft Azure.

Software as a Service

SaaS is model with none of administration tasks left at customer's responsibility.

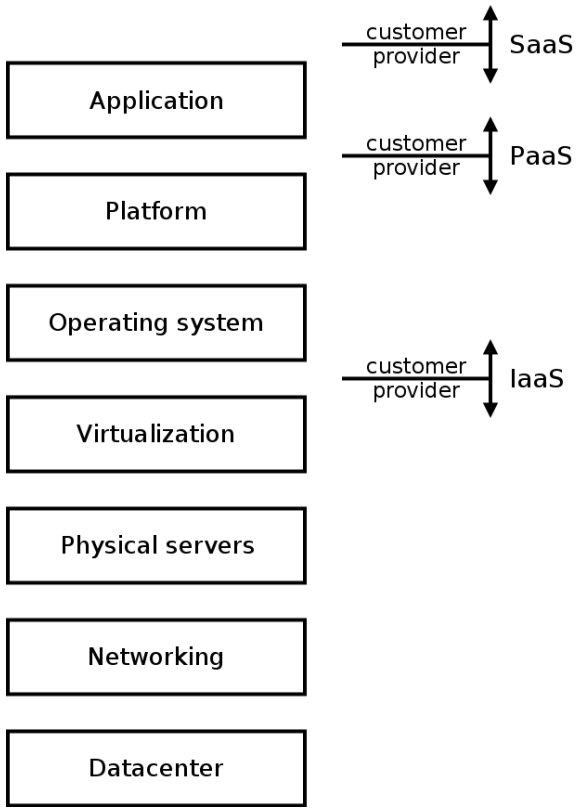

Figure 2.2.1: Service model responsibility

2.2.3 Networking

2.2.4 Storage

2.2.5 Orchestration software

OpenNebula

2.3 Migration of virtual machines

2.4 Distributed data center

Practical part

Methodology overview
Framework
Results

List of Abbreviations

CAPEX	Capital Expenditures.
CPU	Central Processing Unit.
IaaS	Infrastructure as a Service.
IOPS	Input/Output Operations Per Second.
IT	Information Technology.
KVM	Kernel-based Virtual Machine.
LXC	LinuX Containers.
NIST	National Institure of Standards and Technology.
OPEX	Operating Expenditures.
OS	Operating System.
PaaS	Platform as a Service.
SaaS	Software as a Service.
SDN	Software Defined Networking.
UPS	Uninterruptible Power Supply.
US	United States.
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].

[5] Ellen Nakashima. Judge orders microsoft to turn over data held overseas. http://www.washingtonpost.com/world/national-security/judge-orders-microsoft-to-turn-over-data-held-overseas/2014/07/31/b07c4952-18d4-11e4-9e3b-7f2f110c6265_story.html. [Online; retrieved 2014-09-12].