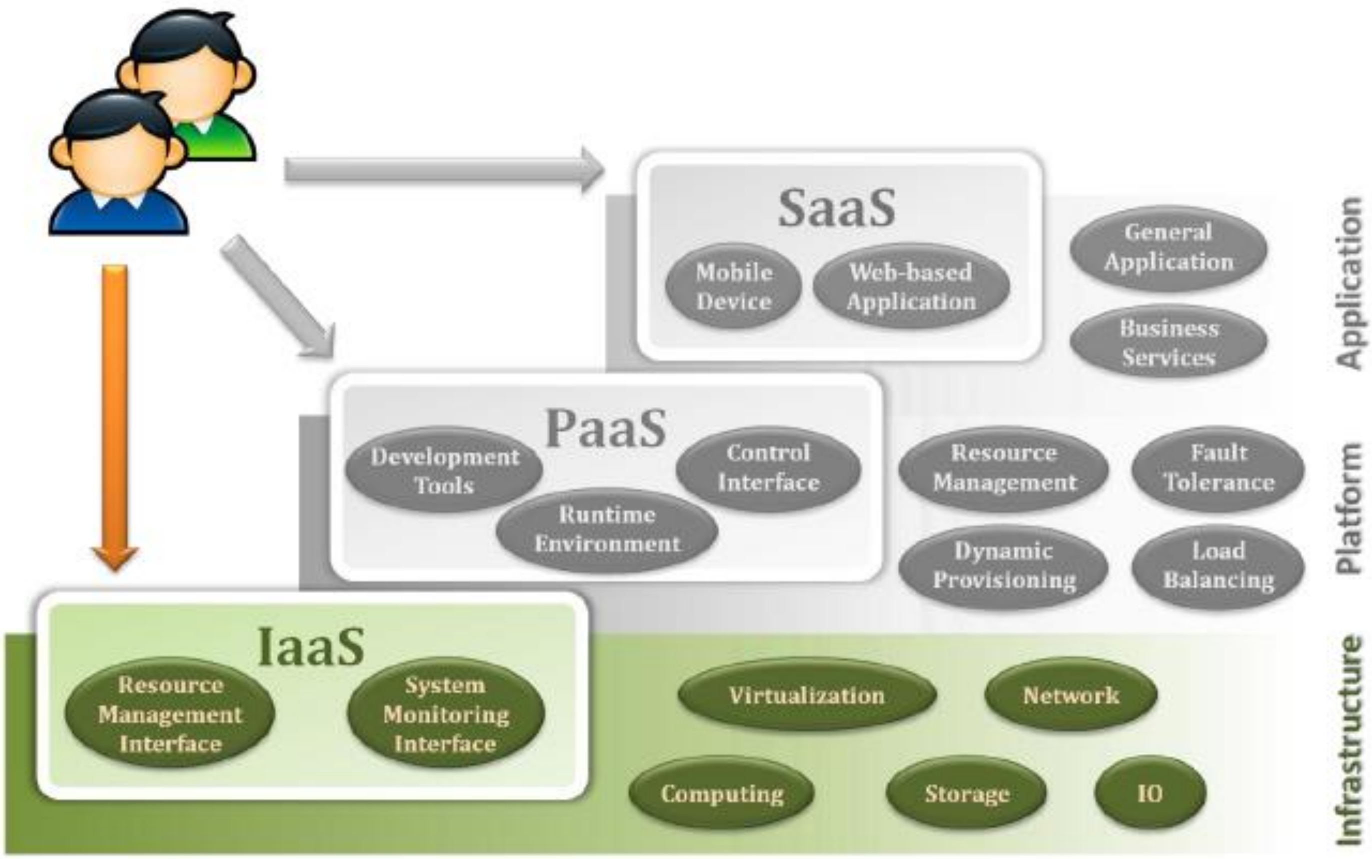
Cloud Computing
-IaaS-

Prof. Lenuța Alboaie
lalboaie@gmail.com

Contents

- IaaS
  - Necessity
  - Definition
  - Example: Eucalyptus
- Virtualization
  - General Aspects
  - Concepts
  - Taxonomy
  - Example
- Docker
- IaaS - characteristics

Cloud Computing



- IaaS – necessity?

Cloud Computing

Issues before the existence of Cloud:

- The IT companies had to invest more in increasing computation
  - Reduce the initial investment
    - Reduce capital expenditure
- Lack of IT infrastructure
- Extra (high) Costs in case of hardware errors
- …

For the final user :

- Reduce the need of local computation
- Reduce the need of local storage
- A rising number of *Thin clients*
- *…*

Cloud Computing

Solutions?

- Outsourcing solutions
  - “Someone” will manage my request of computation or storage
  - “Someone” provides me with these resources anytime
  - “Someone” will be dealing with the level of hardware
  - “Someone” will be dealing with its performance
  - “Someone”…

Cloud Computing

Solution: IaaS – Infrastructure as a Service

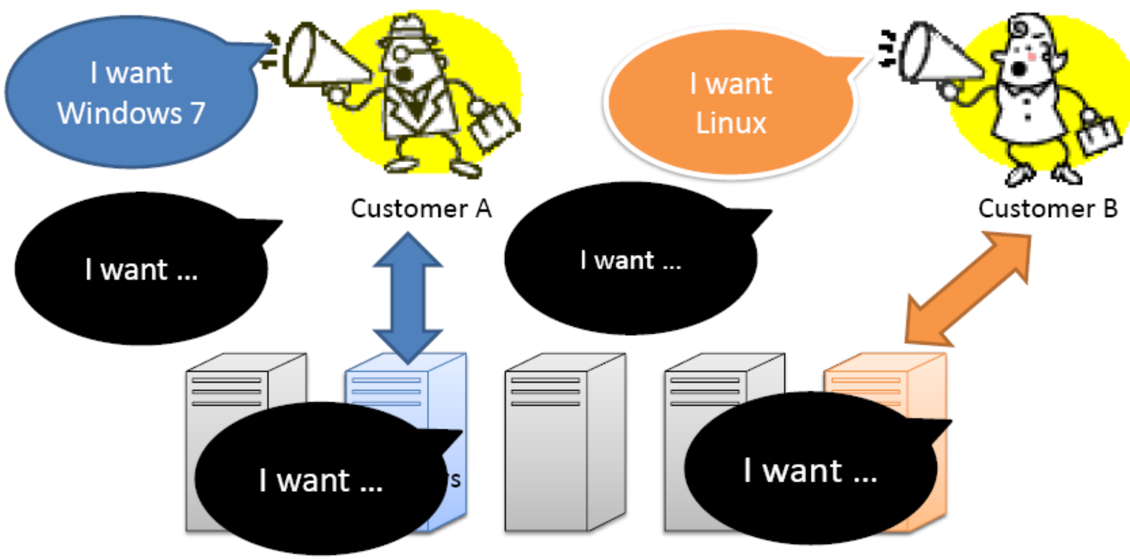
- IaaS Provider
  - Deals with all aspects related to the complexity of its IT infrastructure
  - Provides the functionalities of infrastructure
  - Guaranteese for the provided infrastructure services
  - Calculates a price taking into account all the necessary resources
  - …

How?

Cloud Computing

Let`s imagine: We play the role of an IaaS Provider– Infrastructure as a Service

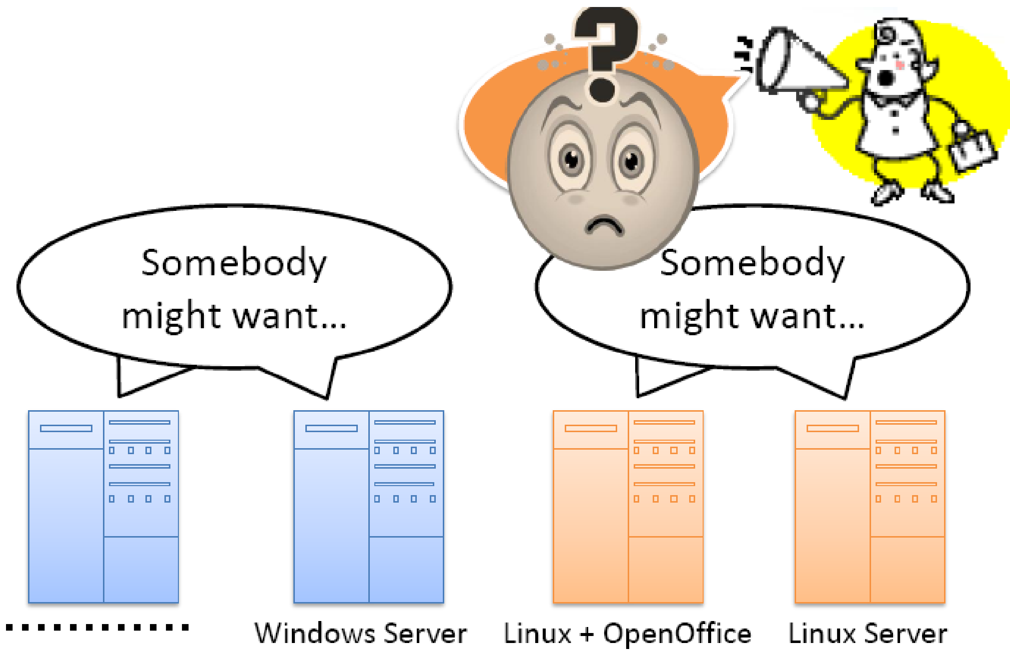
- Clients:
  - Ask for different operating systems
  - Ask for different storage systems
  - Ask for different bandwidth
  - Their requests change once in a while
- *Strategy 1*: Allocate a different physical machinery for each client



Cloud Computing

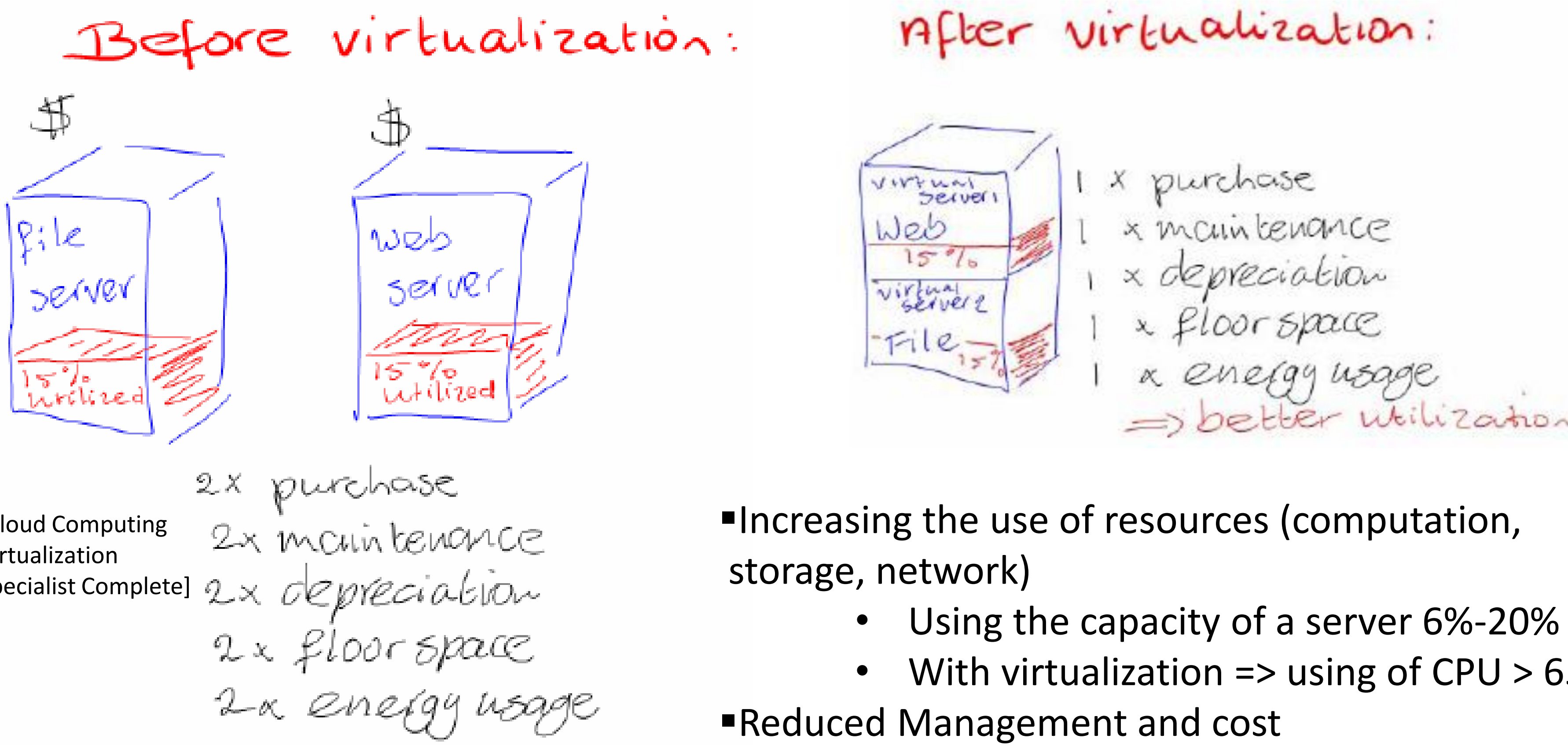
Let`s imagine: We play the role of an IaaS provider – Infrastructure as a Service

– Strategy 2: Preparing a "pool of machines" pre-installed for potentially different requests



– ....Another strategy?

Virtualization



[Cloud Computing Virtualization Specialist Complete]

- Increasing the use of resources (computation, storage, network)
  - Using the capacity of a server 6%-20%
  - With virtualization => using of CPU > 65%
- Reduced Management and cost
- Improving business flexibility
- Improving security and short period of malfunctioning
- *Greening initiatives*

"A cloud's servers need to be virtualized "

Cloud Computing

IaaS provides infrastructure for the cloud as a service, enabling a virtualized environment.

Virtualization stands for the techniques employed for creating an abstract level represented by logical resources, while based on physical resources.

The virtualization techniques that plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions:

- Computing resources
  - *Virtual Machine techniques*
- Storage resources
  - *Virtual Storage techniques*
- Communication resources
  - *Virtual Network techniques*

… *Application Virtualization, Desktop Virtualization, Data Virtualization, GPU Virtualization, ….*

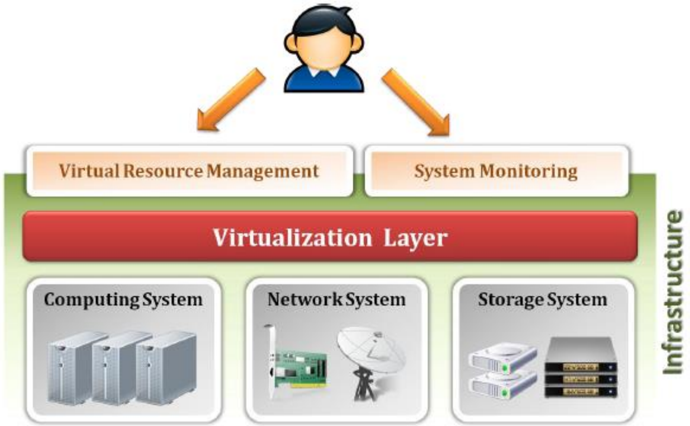
Virtualization | General Overview

What is virtualization?

•In cloud computing, virtualization refers to the abstraction and isolation of hardware resources into a form that allows them to be accessed and utilized efficiently by users or applications, regardless of the physical details of the underlying infrastructure

•The physical traits of the physical resources are hidden to the user, these perceiving only an abstract resource.

•Ex: The process of creating a virtual version of an operating system, server, storage or communication mechanisms.



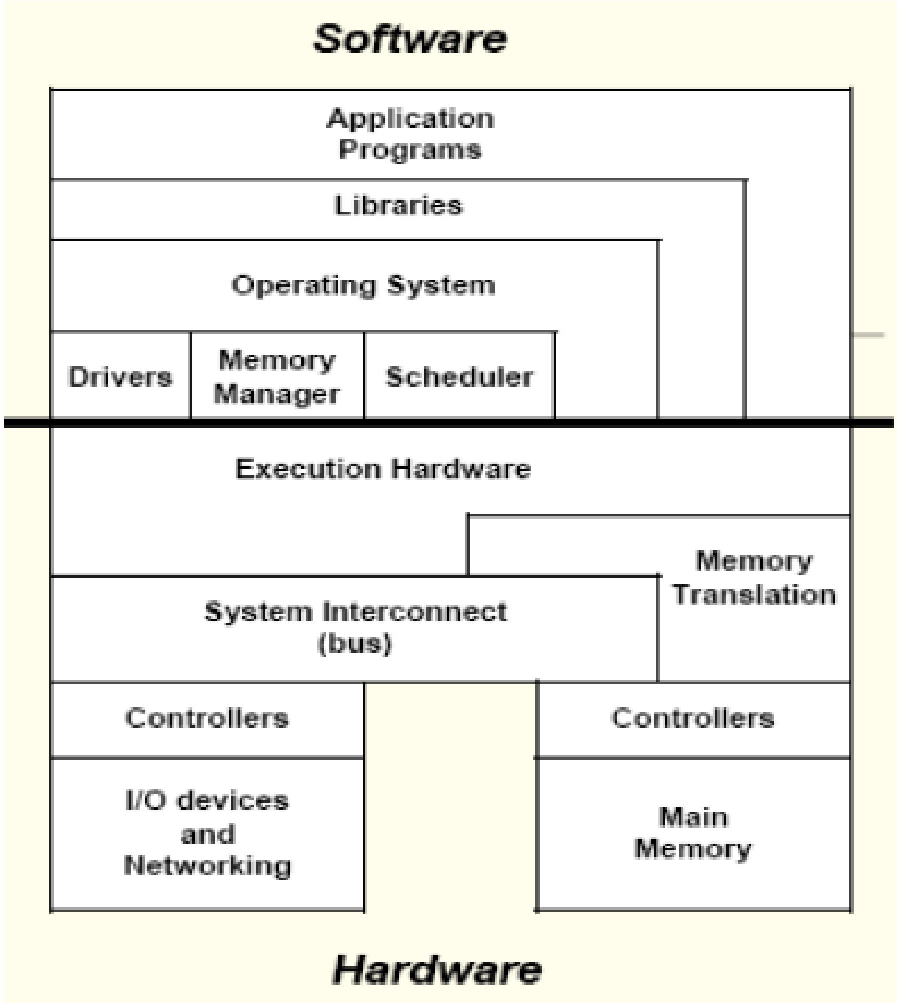
Virtualization | General Overview

- The virtualization is no novelty, and it was implemented on several levels.
- The concept of *virtual memory (1962*)
  - Although the physical location of the data is scattered on RAM and HDD, the virtual memory process creates the feeling of continuous and orderly storage.
  - Removes the programmers' concerns about the physical memory limitations
- In the sixties, the IBM *mainframes* were using the *time-sharing virtualization*

Virtualization | General Overview

- The abstraction of a computing system:
  - Each level is abstracted and hides the details of lower levels
  - Uses the lower level functions and provides an abstraction to the upper level

  Example: the files are an abstraction of a disk

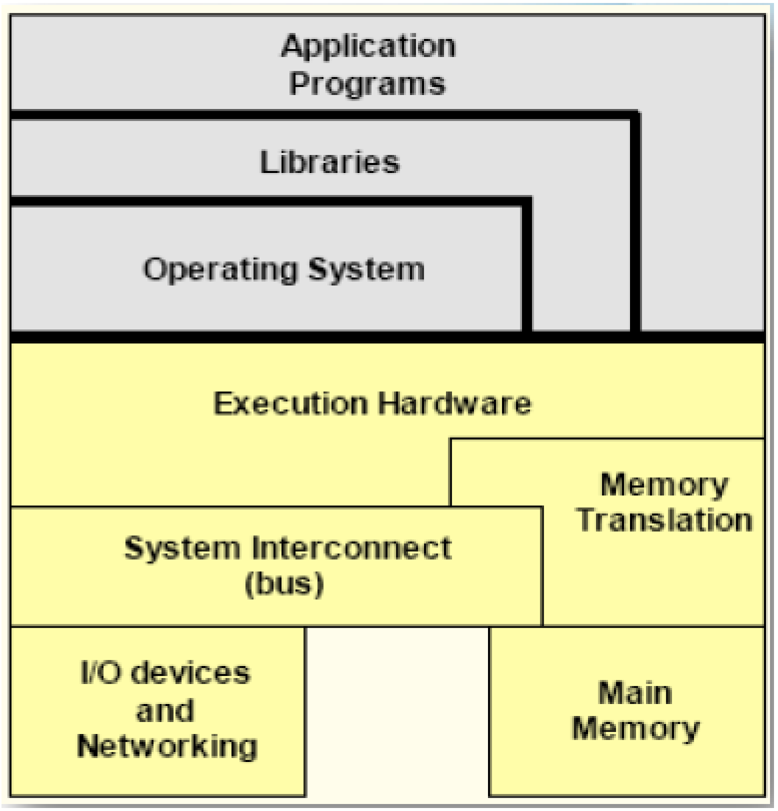


Virtualization | General Overview

- Abstraction on machine level
  - From the developers' perspective, a machine is defined by ISA (Instruction Set Architecture)
- The division between hardware and software

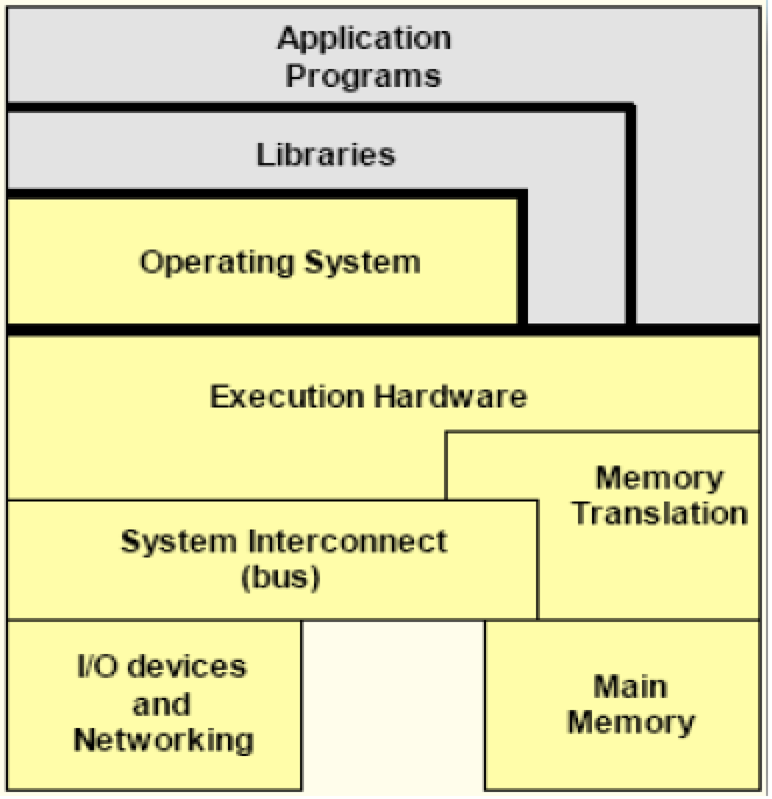
Examples:

- X86 (based on Intel 8086 CPU)
- MIPS (Microprocessor without Interlocked Pipeline Stages)
- ARM (on Mobile devices or Embedded)



Virtualization | General Overview

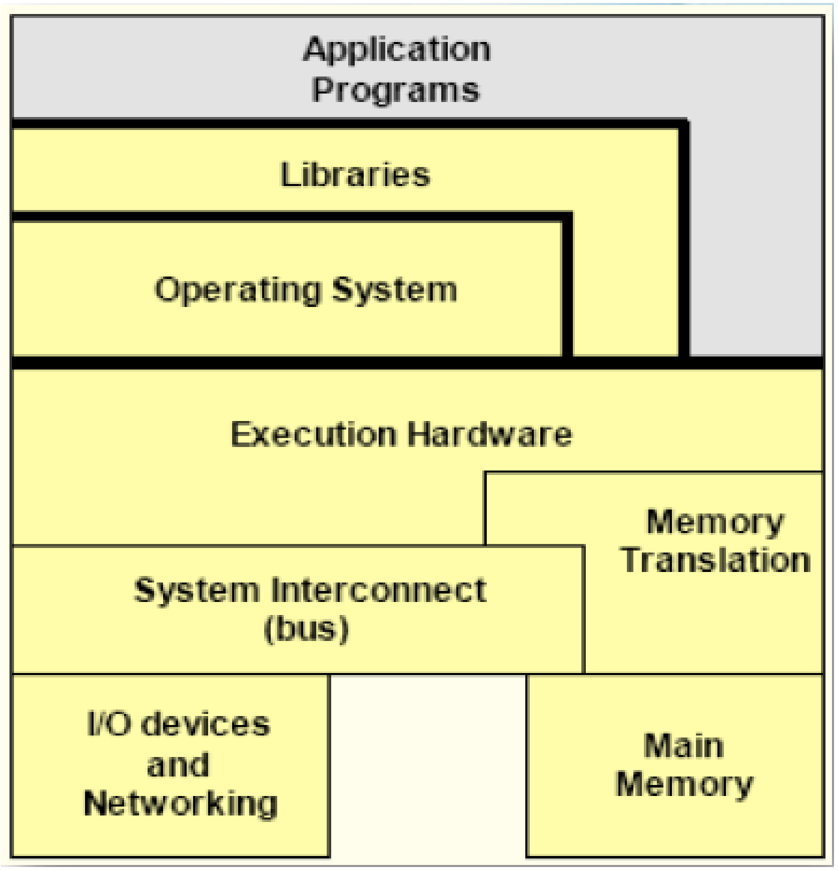
- Abstraction on the operating system level (OS)
  - For libraries developers, a machine is represented by ABI (*Application Binary Interface*) – an OS interface usable on the upper level
  - Examples:
    - ISA users
    - OS system calls



Virtualization | General Overview

Abstractization on library level

- For application developers, a machine is defined by an API (Application Programming Interface)

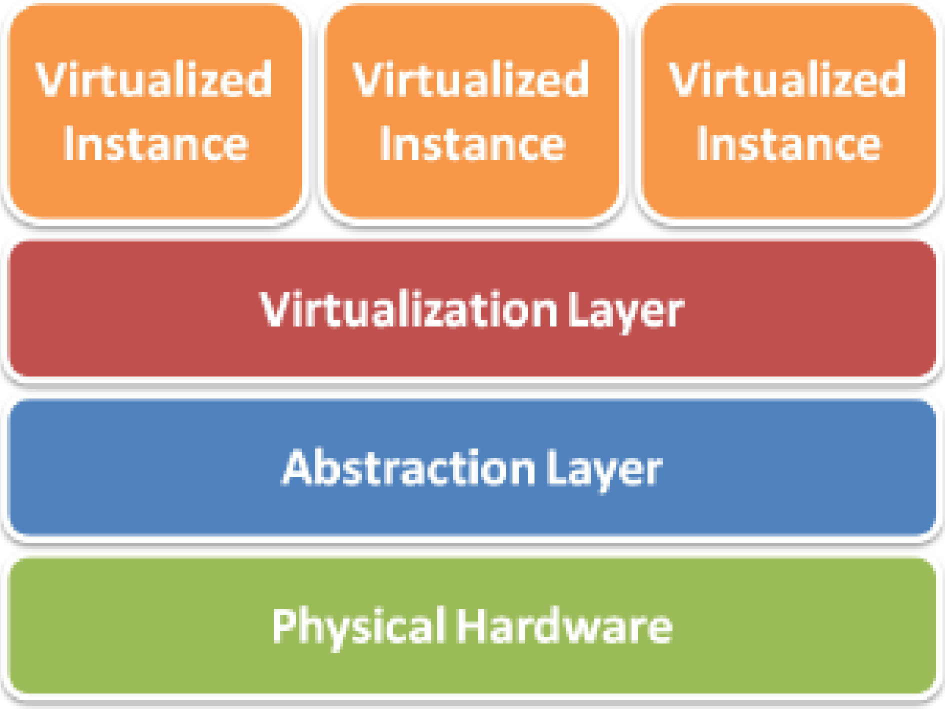


Examples:

- ISA use
- Graphic libraries
- C standard libraries

Virtualization | General Overview

- The virtualization concept is ubiquitous
- In IaaS the virtualization is present on all physical devices
  - Server, Storage, Network



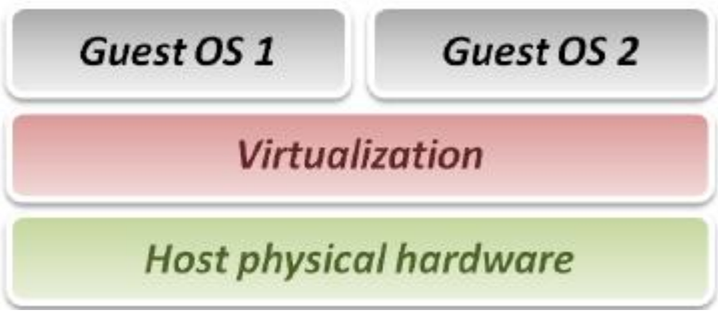
Virtualization

- Terminology
  - **VM (Virtual Machine)** – the software that implements a machine's functionalities and enables program execution
  - **Host** (Target) – the primary environment that is to be the virtualization target
  - **Guest** (Source) – the virtualized environment that is to be the virtualization source
  - **Emulation**
    - The simulation of an independent environment where *guest ISA* and *host ISA* are different
    - Example: An x86 architecture emulation on an ARM platform
  - **Virtualization**
    - The simulation of an independent environment where *guest ISA* and *host ISA* are the same
    - Example: the virtualization on a x86 architecture of multiple instances

Virtualization

- Emulation
  - Provides a method that enables sub(systems) to have the same interface and characteristics as others

  - The virtualization may be considered a special case of emulation (multiple virtualization techniques have been inherited based on emulation techniques)



Virtualization

- Terminology

Hypervisor: This is the layer of software that creates and runs virtual machines. There are two types of hypervisors: Type 1 (or bare-metal) and Type 2 (or hosted).

Containers: These are a lightweight form of virtualization at the operating system level, allowing you to run multiple isolated user-space instances.

Resource Pooling: This concept refers to the provider's ability to serve multiple consumers (multiple tenants) with provisional and scalable services.

Snapshot: A saved state of a VM at a specific point in time. It can be used to restore a VM to that point.

Live Migration: This is the process of moving a running VM or application between different physical machines without disconnecting the client or application.

Orchestration: Automated management, coordination, and arrangement of computer systems, middleware, and services.

Software-Defined Networking (SDN): An approach to networking that uses software-based controllers or application programming interfaces (APIs) to direct traffic on the network and communicate with the underlying hardware infrastructure.

Software-Defined Storage (SDS): This concept abstracts storage resources and enables policy-based management of data storage independent of the underlying hardware.

Virtualization

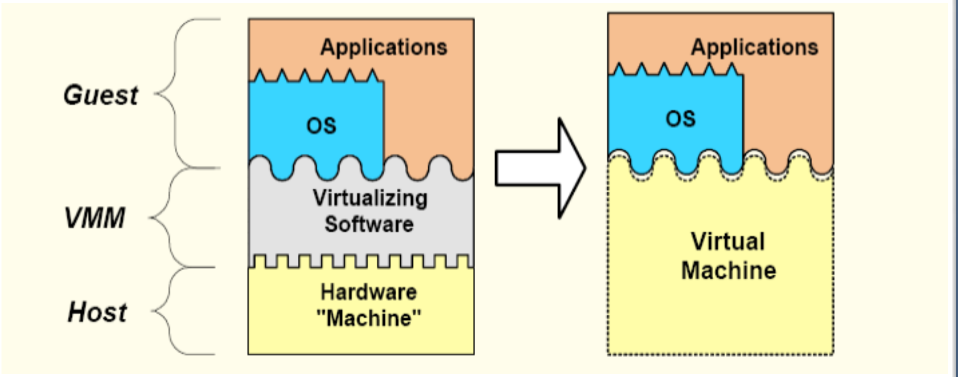
- Examples

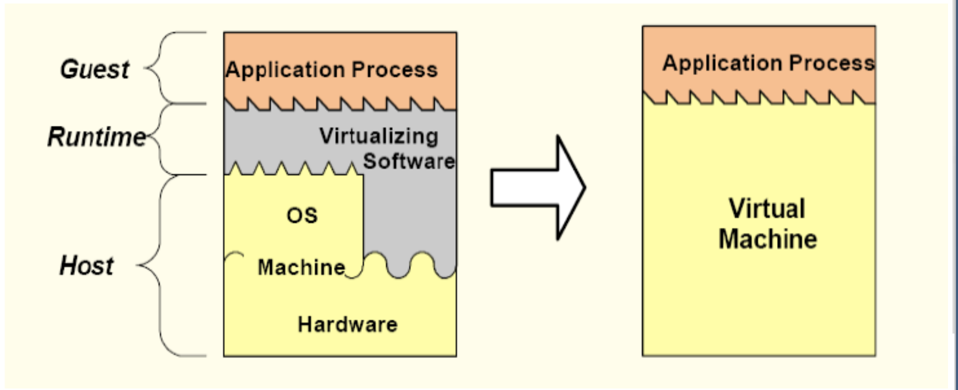
- *System Virtual Machine(SVM)*
  - Provides the whole operation system on a different or identical host ISA
  - Built on ISA level

- *Process Virtual Machine*
  - It usually executes *guest* applications with an ISA different by the *host* ISA
  - Coupling at ABI (Application Binary Interface) level via a *runtime system*

	System Virtual Machine	Process Virtual Machine
Emulation	*Transmeta Crusoe* (*Emulate x86 on VLIW cpu*)	*Multi-processing system*
Virtualization	*XEN, KVM, VMWare* (*x86 virtualization software*)	*JVM, Microsoft CLI* (*High level language virtualization*)

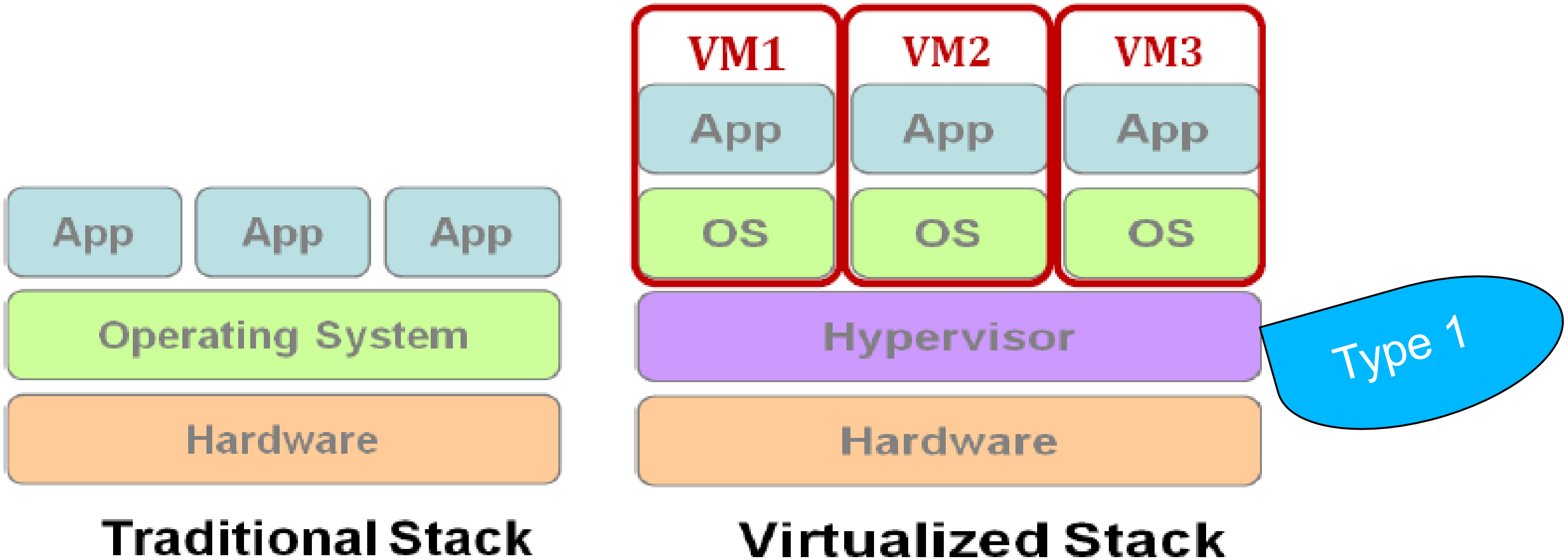
Techniques utilized in IaaS





Virtualization

- VMM (Virtual Machine Monitor) or Hypervisor is the software level that provides the virtualization



- Properties (according to Popek and Goldberg), VMM must comply with three properties:
  - Equivalence (~ as the real machine)
  - Resource control (~ total control)
  - Efficiency (~native execution)

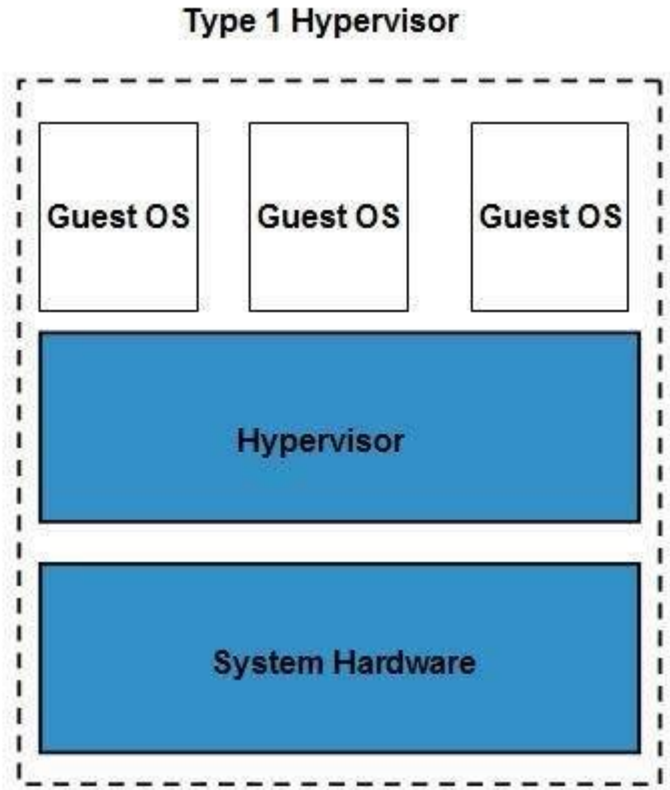
Virtualization

Classification based on the **implementation approach** and the **level of hardware abstraction** they provide:

Hypervisor-Based Virtualization:

– ***Bare Metal Virtualization (type-1 hypervisor)***

- VMM (**Virtual Machine Monitor)** directly runs on the *host* hardware and plays the role of hardware control and monitor of *guest operating systems*
- **Hardware-assisted virtualization** refers to specific processor extensions that improve the efficiency and performance of the virtualization
- Intel's VT-x and AMD's AMD-V technologies are hardware features built into the processors that enable a type-1 hypervisor to more efficiently run virtual machines by allowing some of the virtualization functions to be handled directly by the physical CPU
    - => some tasks that the hypervisor would usually have to emulate can be done natively by the CPU hardware -> close to native performance
- bare metal hypervisors don't necessarily require hardware-assisted virtualization to operate, BUT this technology significantly improves their performance and capability
- Examples: LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Server, VirtualLogic VLX



[https://www.tutorialspoint.com/cloud_computing/cloud_computing_virtualization.htm]

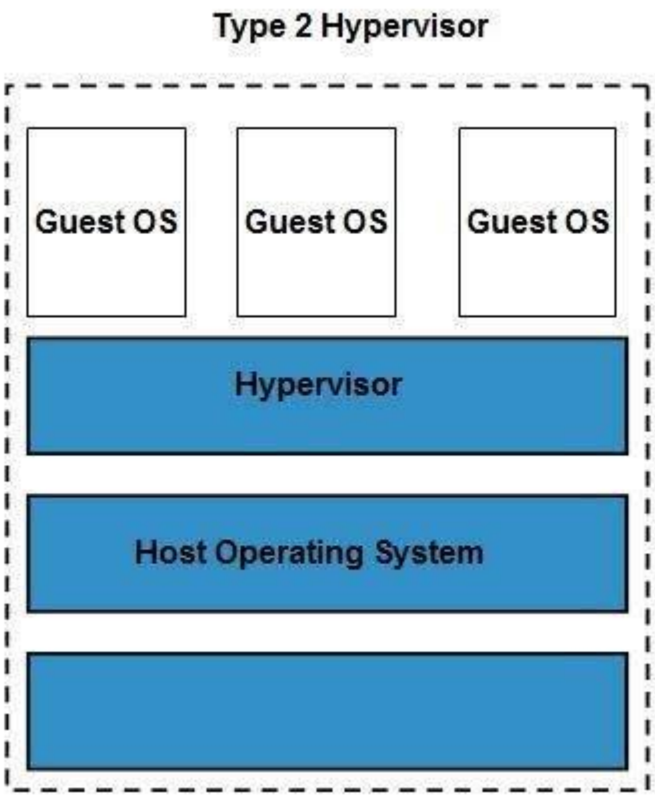
Virtualization

Classification based on the **implementation approach** and the **level of hardware abstraction** they provide:

Hypervisor-Based Virtualization:

– ***Hosted Virtualization (type -2 hypervisor)***

- involves a layer of software—the Virtual Machine Monitor (VMM)—that operates atop a existing operating system
- host OS provides standard drivers and system services to the VMM, which in turn creates, runs, and manages virtual machines (VMs).
- Each VM is a separate software container that encapsulates an operating system and applications, allowing them to run as if they were on their own separate physical machine
- Examples: Microsoft Hyper V, VMWare Fusion, Virtual Server 2005 R2, VMWare workstation 6.0,…



[https://www.tutorialspoint.com/cloud_computing/cloud_computing_virtualization.htm]

Virtualization

Classification based on the **implementation approach** and the **level of hardware abstraction** they provide:

Operating System-Based Virtualization:

– ***OS-Level Virtualization***

– doesn’t involve hypervisors at all

– it works by providing isolated user-space instances, which are called containers

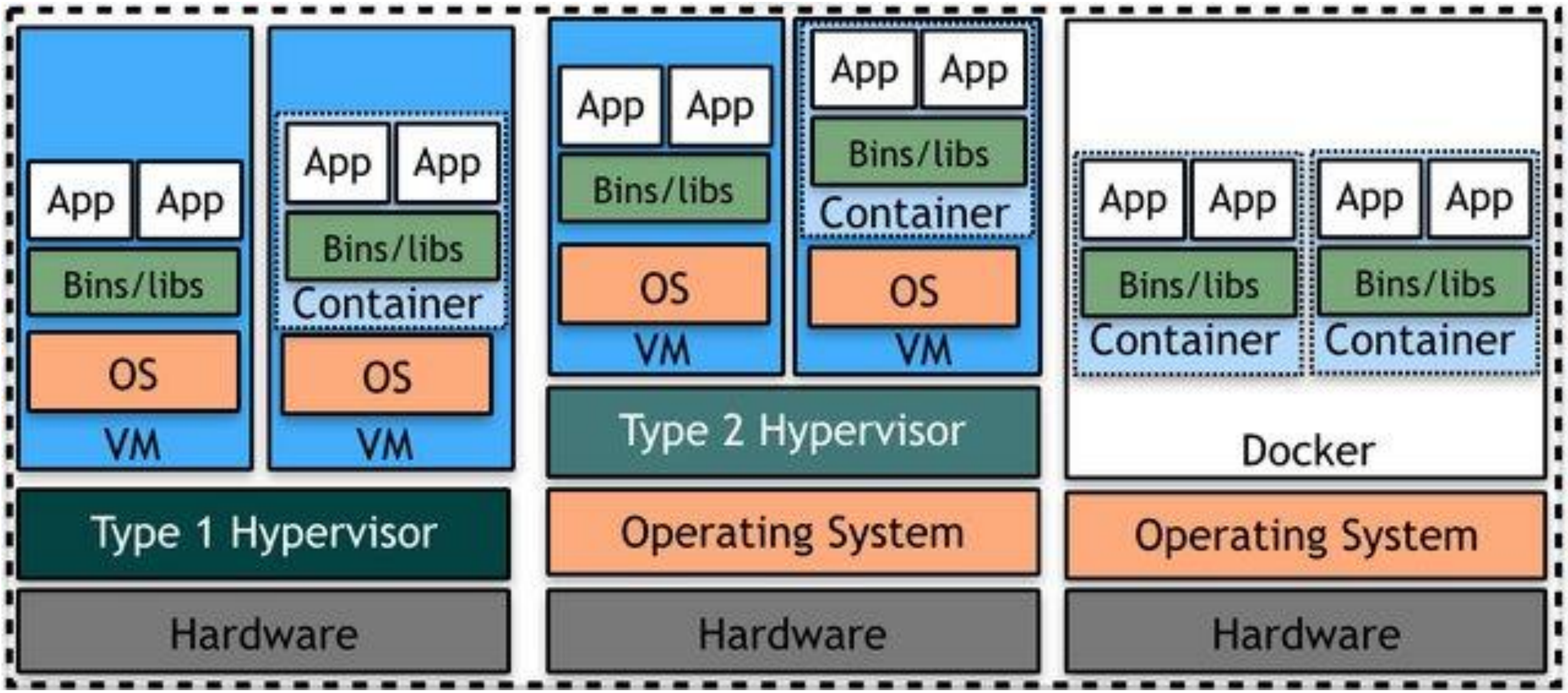
– these containers share the same kernel of the host's operating system but have separate user spaces.

– they're more lightweight than full virtual machines because they don't need to emulate or simulate hardware and don't have the overhead of running separate kernel instances.

– Examples of OS-level virtualization: Docker, LXC (Linux Containers), or FreeBSD Jails (see Week 6)

Virtualization

Classification based on the **implementation approach** and the **level of hardware abstraction** they provide:



[Al-dhuraibi, Yahya & Fawaz, Paraiso & Djarallah, Nabil & Merle, Philippe. (2017). Elasticity in Cloud Computing: State of the Art and Research Challenges. IEEE Transactions on Services Computing. PP. 1-1. 10.1109/TSC.2017.2711009]

Virtualization

Classification based on the **implementation approach** and the **level of hardware abstraction** they provide:

– ***Full-Virtualization***

– **With Hardware Assistance**: Uses processor extensions like Intel VT-x or AMD-V to improve performance.

– **Without Hardware Assistance**: Employs software methods like binary translation for virtualization without direct hardware support.

- VMM emulates sufficient hardware to enable an unmodified *guest OS*

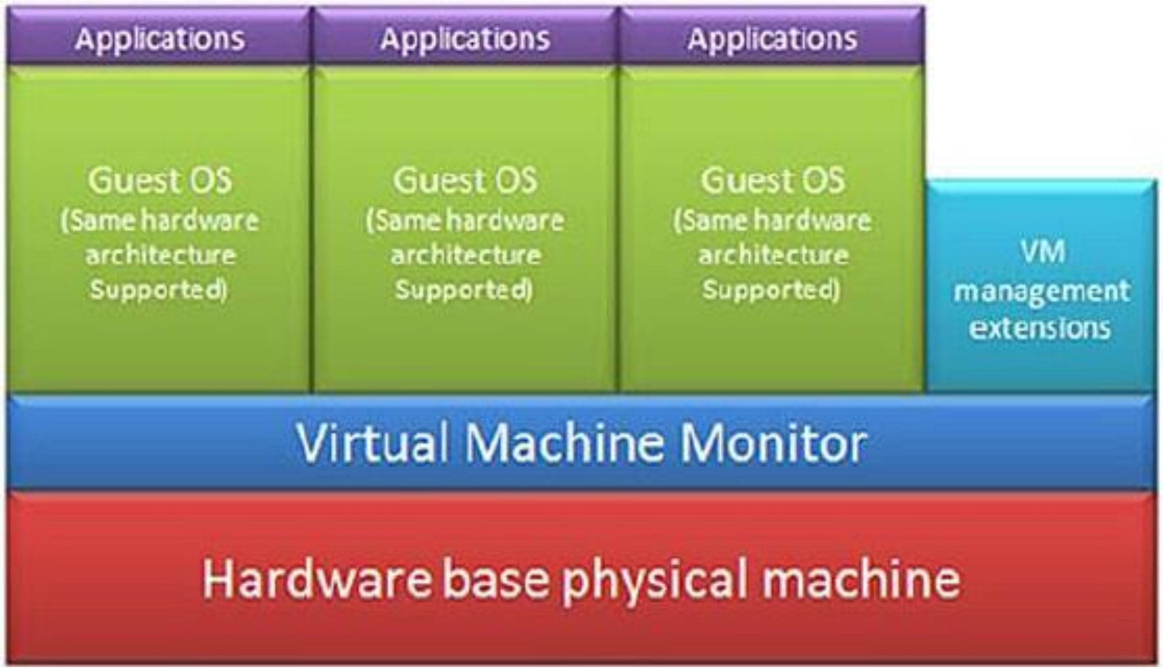
– ***Para-Virtualization***

- VMM does not emulates on hardware level, but provides a special API that may be used only as a modified *guest OS*

Can be implemented by both Type 1 and Type 2 hypervisors

Virtualization

– ***Full-Virtualization***

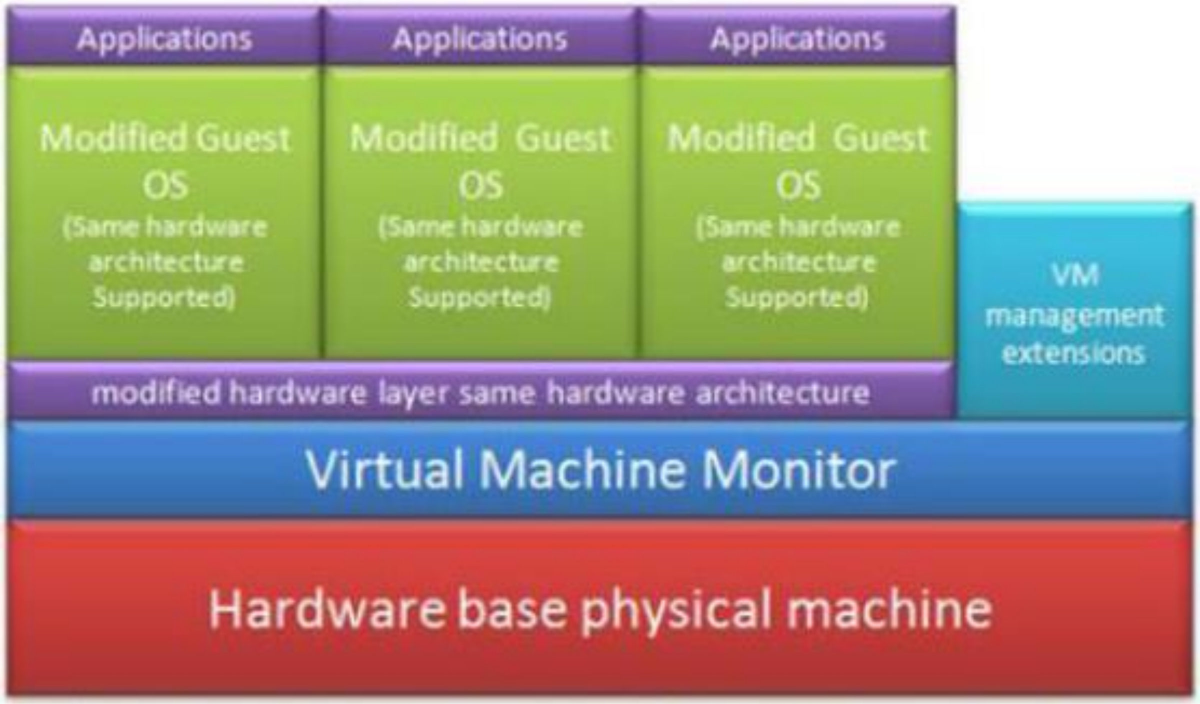


– Pros: does not require the *OS guest* modification

– Cons: lower performance

Virtualization

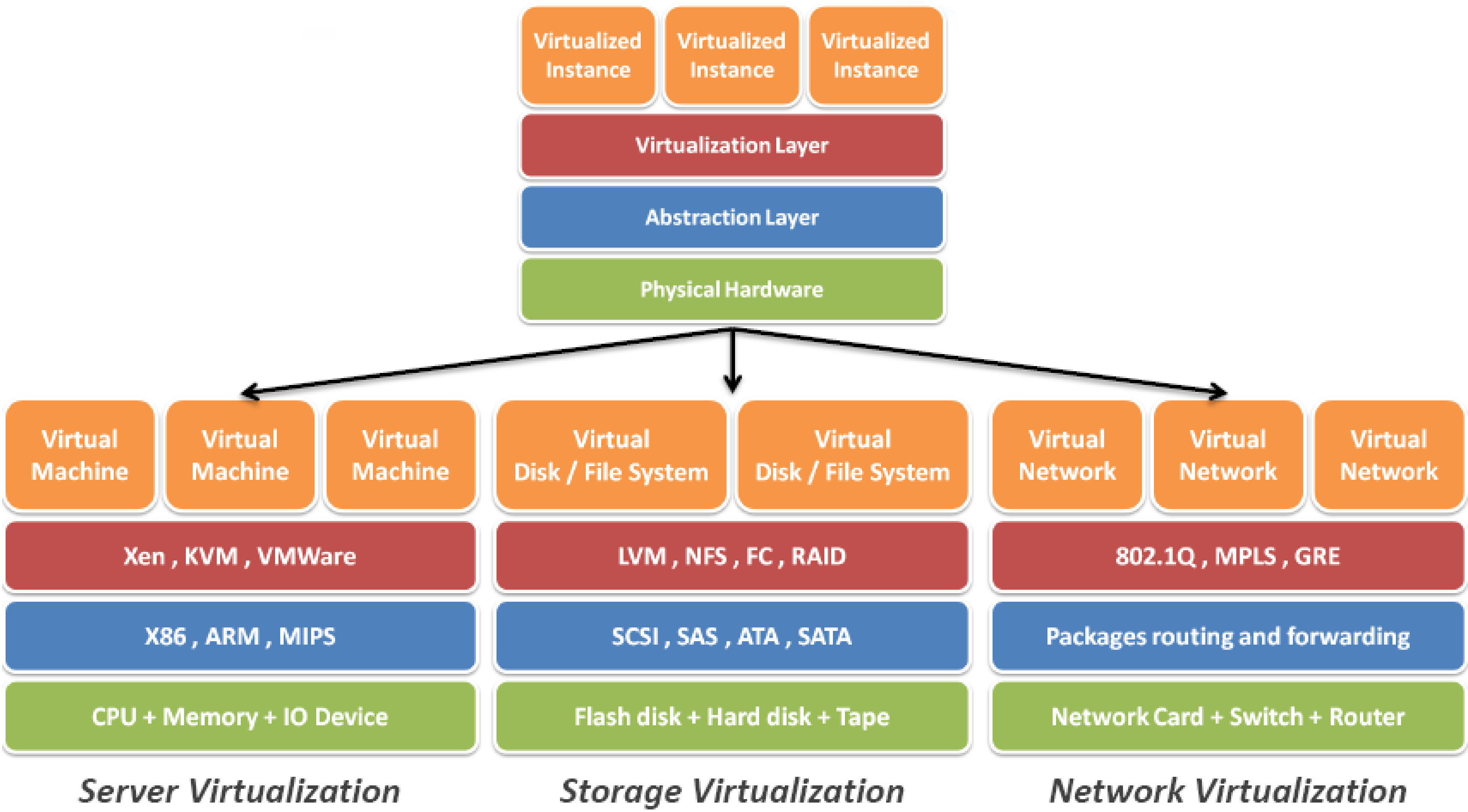
– ***Para-Virtualization***



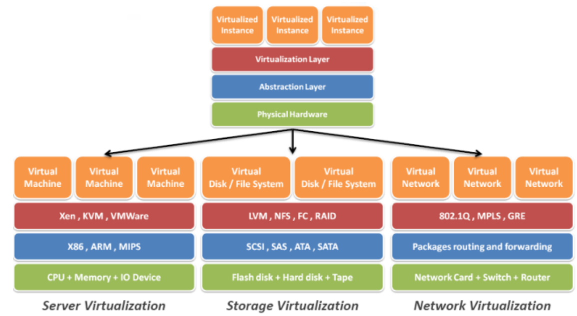
– Pros: high performance
– Cons: requires modification on *guest OS* level

Virtualization

Virtualization techniques| Examples



Virtualization



Examples:

Xen

•Initially developed by the Cambridge University in the late 1990s

•Linux Foundation took over the project in 2013

•Xen-based hypervisor is a Type 1 hypervisor, which enables IT administrators to run multiple OSes on the same hardware and has a small management layer to manage shared resources

•Since 2010, Xen – GNU General Public Licence (GPLv2)

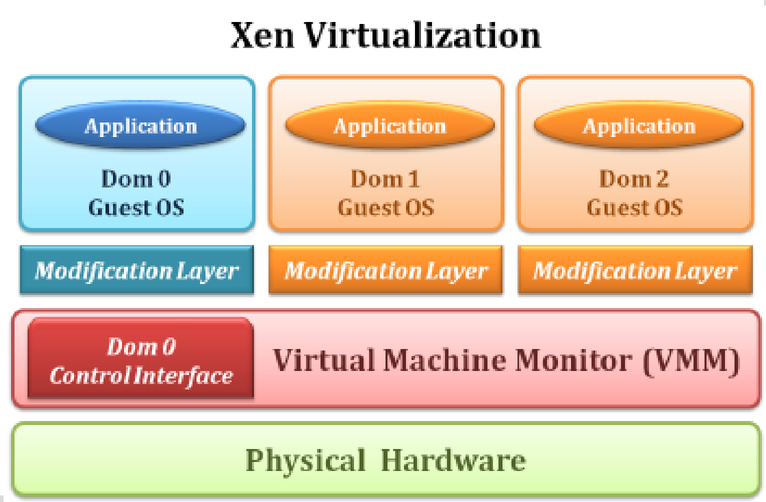
Bare metal

Para-Virtualization (PV) or

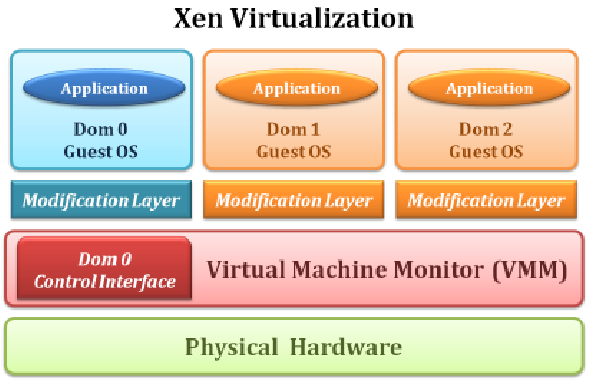
Full Virtualization

•Intel and AMD contributed to the Xen support

• Used by: Citrix, Oracle, Amazon



Virtualization

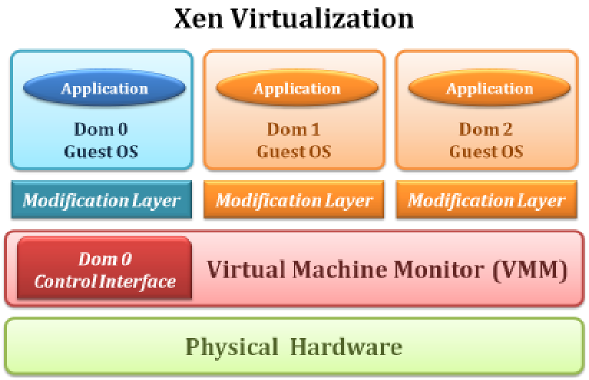


Examples:

Xen architecture

- Xen utilizes a microkernel architecture that separates the hypervisor core from the virtual machines it manages. This design enhances security and stability by minimizing the code that runs in privileged mode
- Domain-0 (Dom0): This is the first virtual machine launched by the Xen hypervisor and runs a full operating system. Dom0 has special administrative privileges and manages hardware interactions and device controllers. It is responsible for creating and managing guest virtual machines (DomU), including the installation and support of device drivers, networking, and storage.
- User Domains (DomU): These are guest virtual machines running on the same physical hardware under the control of the Xen hypervisor. DomUs can either be paravirtualized, meaning they are aware of and optimized for running in a virtualized environment, or they can use full virtualization, in which case they behave as if they are running on physical hardware.

Virtualization

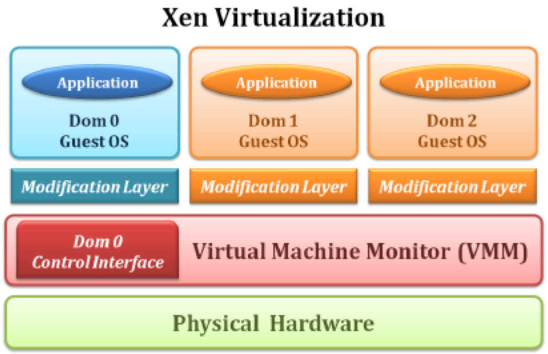


Examples:

Xen architecture

- Interaction between Dom0 and DomU: Dom0 facilitates the interaction with the hardware, including networking and storage, for DomU. It uses paravirtualized device drivers to provide superior performance by reducing virtualization overhead.
- Resource Management: Xen allows efficient allocation and isolation of physical resources between Dom0 and DomU, enabling administrators to finely control the use of CPU, memory, and other resources.
- Security and Isolation: Xen provides strong isolation between virtual machines, which enhances security. Xen secures each DomU so that even if one is compromised, the impact on other virtual machines and the host system is minimized.

Virtualization



Examples:

Xen in Cloud

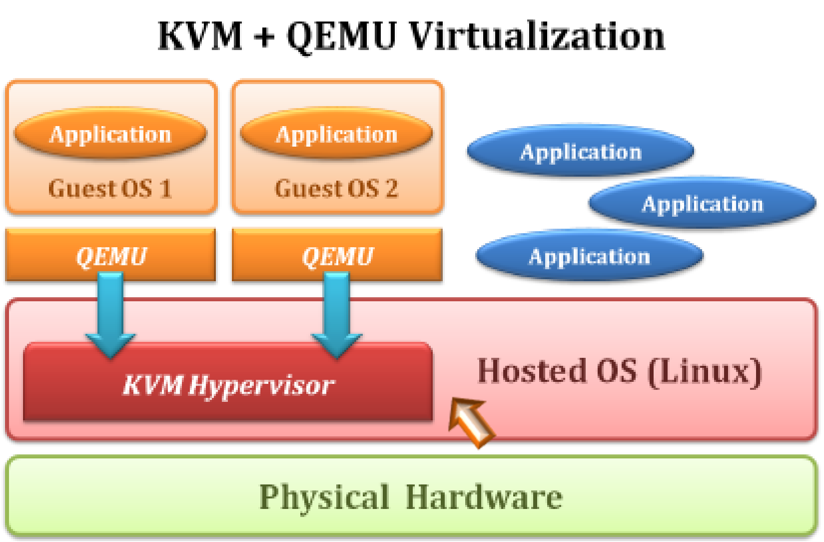
- Xen Server as a Foundation for Citrix Hypervisor:
  - Xen Server provides the underlying infrastructure for Citrix Hypervisor, widely used in enterprise cloud environments to deliver advanced features such as live migration, disaster recovery, and high availability
- Integration with Cloud Orchestration Tools:
  - Xen seamlessly integrates with popular cloud orchestration tools like Apache CloudStack and OpenStack, which facilitate the deployment, management, and configuration of cloud environments
    - With OpenStack: In OpenStack environments, Xen is supported through the XenAPI, which is part of the Nova compute module. This allows users to utilize Xen for server virtualization within the OpenStack framework, benefitting from Xen's performance and security features.
- Xen's Role in Public and Hybrid Clouds: Xen is utilized in both public and hybrid cloud setups due to its high performance, scalability, and security. Public cloud providers, such as Amazon Web Services (AWS), offer Xen-based virtual instances, demonstrating the hypervisor's capability to support massive, multi-tenant cloud infrastructures.

Virtualization

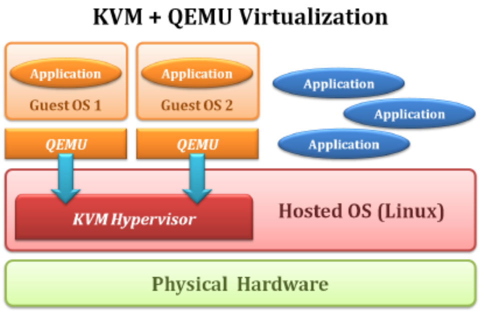
Examples:

KVM (Kernel-based Virtual Machine)

- Adopted in Linux in 2007, is a hypervisor that virtualizes OSes on x86 server hardware
- KVM it turns the Linux kernel into a Type 1 hypervisor.
    - Although it is part of the Linux kernel, once loaded, it allows the kernel to function as a bare-metal hypervisor.
    - This sets it apart from other Type 2 virtualization solutions, such as VMware Workstation or Oracle VirtualBox, which run as applications within the host operating system.
- **Ubuntu** (Server Edition) provides KVM (Kernel-based virtual machine);
    - KVM may use AMD and Intel harware virtualization techniques => efficient virtualization process
    - KVM, implemented on the OS kernel may run multiple virtual machines with great OS variety (Linux, Windows, MacOS, Solaris)
- Used by: OpenStack, oVirt, Amazon



Virtualization



KVM (Kernel-based Virtual Machine)

KVM Component:

- This is an integral part of the Linux kernel and assumes the role of a hypervisor
- It is loaded as a module into the Linux kernel and utilizes specific hardware virtualization facilities provided by modern processors Intel VT-x or with AMD-V technology

QEMU: QEMU (Quick Emulator)

- is used together with KVM for hardware emulation and to provide virtual machines access to physical devices, such as disks, networks, USB devices etc
- QEMU can run independently for complete hardware emulation, but when combined with KVM, it enhances performance due to hardware acceleration

Hosted OS (Linux): This is the host operating system that runs on the physical hardware. Since KVM is part of the Linux kernel, the host operating system directly benefits from virtualization and can concurrently run other operating systems as virtual machines.

Virtual Machines (VMs): Within KVM, each VM is represented as a Linux process, thereby benefiting from all the security and resource management of the kernel. These VMs can run various operating systems, from Linux to Windows or macOS, each with its own isolated memory space and CPU access.

Virtualization

HVM or Xen hypervisors

- Xen hypervisor uses a microkernel design that runs on bare-metal hardware and can run on systems without virtualization extensions
  - Is an issue for older hardware
  - From Xen 4.14 version - security-focused features (e.g. a lightweight VM fork to analyze malware)
- KVM is that it functions at the Linux OS kernel; this means that KVM gets bug fixes and security updates as Linux publishes new releases
  - KVM Nitro is an Amazon Elastic Compute Cloud capability that carves out isolated compute environments within the same instance
  - Security and isolation are the primary motivations to protect sensitive data at the VM level

Virtualization

HVM or Xen hypervisors

- Oracle and Citrix have a large customer base and push Xen as their primary hypervisor
- Red Hat, SUSE, support KVM as a virtualization option for their Linux versions
- For cloud:
  - Citrix and Oracle have a Xen-based offering
  - Google on KVM
  - Amazon offers both Xen and KVM
- Major cloud vendors provide both web-based and programmatic interfaces to enable flexibility for IT teams and admins
- Consider the existing virtualization software and how well it integrates with any prospective cloud provider – hybrid and on-premise cloud perspective

Virtualization

When to Choose Xen?

Stability and Maturity: Xen is one of the **oldest open-source hypervisors** and has a strong track record of stability and performance in production environments. It's a good choice if you value proven reliability in your virtualization layer

Paravirtualization: Xen shines in **environments where paravirtualization is preferred,** especially for non-Windows guest operating systems. Paravirtualization with Xen can offer better performance for certain workloads because it allows for more direct interaction between the OS and the hardware

Security: Xen has a **strong focus on security**, with advanced isolation features that can be crucial for multi-tenant environments such as public clouds. If security and isolation are paramount, Xen's architecture offers benefits

Independent Project: Being an independent project, Xen might be preferred in environments that **seek to avoid vendor lock-in** associated with solutions that are closely tied to specific commercial entities

Virtualization

When to Choose KVM?

Integration with Linux: KVM is integrated into the Linux kernel, making it **a seamless choice for Linux environments**. This integration facilitates management and potentially simplifies support for Linux-based infrastructures

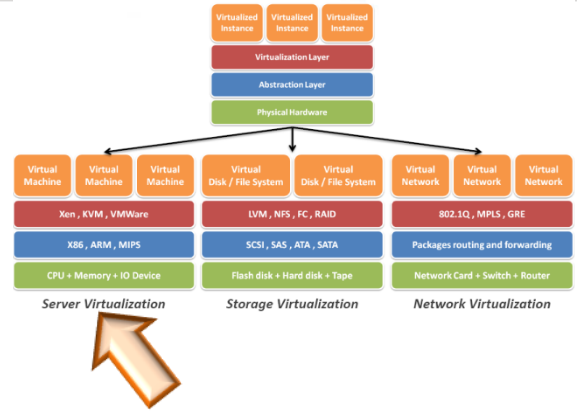
Performance: For raw performance, especially with **hardware supporting virtualization extensions (like Intel VT or AMD-V)**, KVM can be faster because it leverages the existing Linux kernel for many tasks, reducing overhead

Hardware Support: KVM typically benefits from better support for **newer hardware features** due to its inclusion in the Linux kernel. If you're using the latest hardware technologies, KVM might offer more out-of-the-box support.

Ease of Use and Management: With tools like libvirt and a strong ecosystem of management interfaces (e.g., Virt-Manager, oVirt), KVM can be **easier to manage,** especially for organizations already familiar with Linux administration.

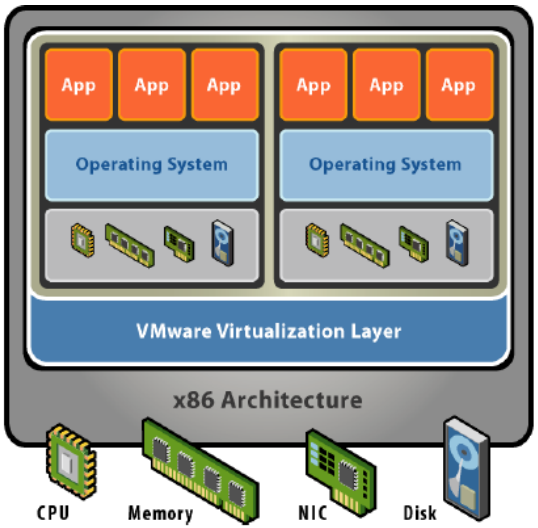
Community and Vendor Support: KVM enjoys **robust support from major Linux vendors** (e.g., Red Hat, Canonical), which can be a decisive factor for organizations that rely on enterprise support for their infrastructure.

Virtualization



VMware

- Company founded in 1998 , Palo Alto, California
- Owned by the EMC Corporation
- Implements *bare-metal* or *hosted* virtualization



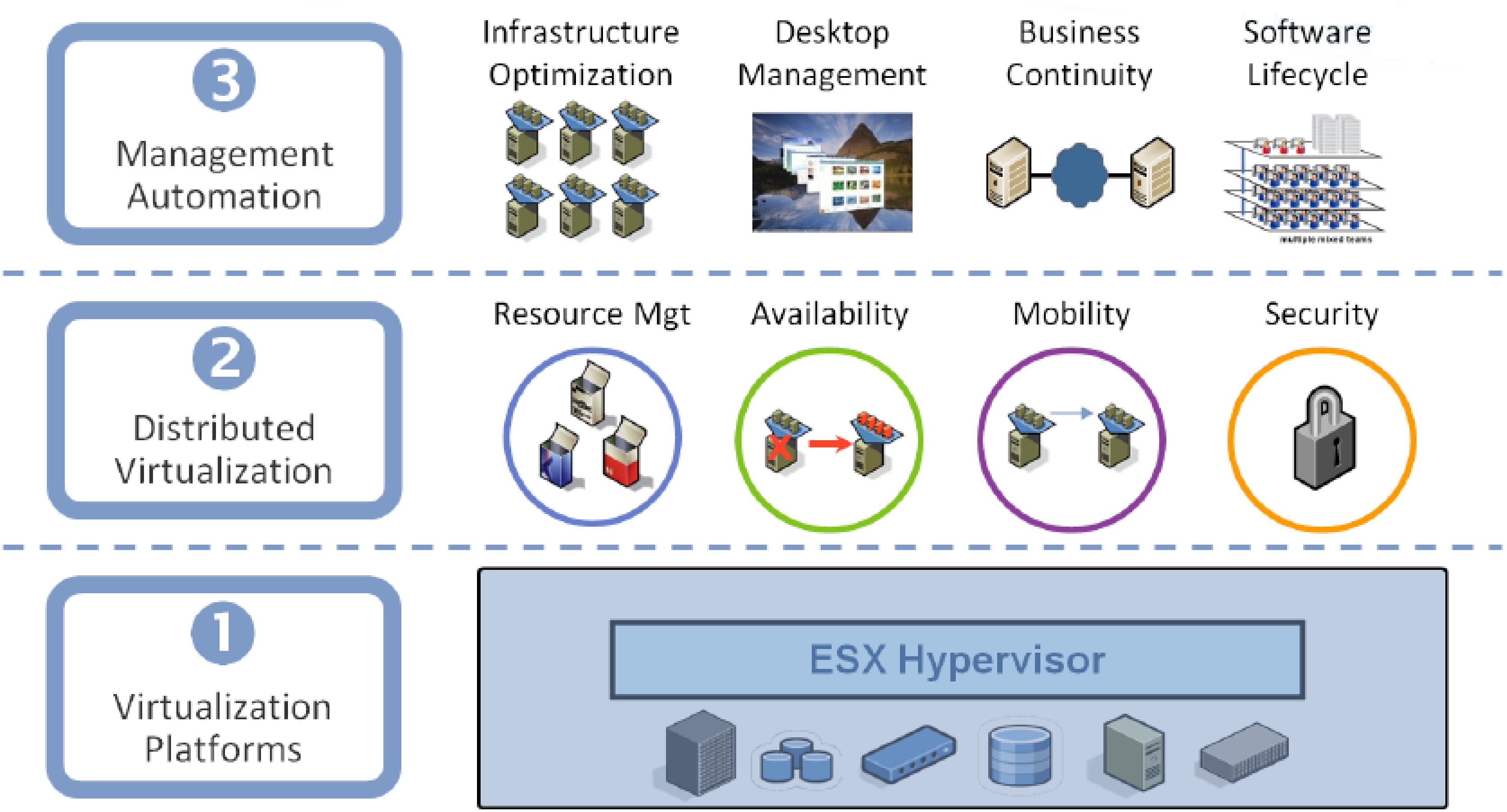
Picture. The architecture of a virtual machine using VMware on a x86 architecture
- 1999 VMware introduces the first virtualization application for x86 systems

[Cloud Computing Virtualization Specialist Complete]

- Core proprieties:
  - Separation of OS and hardware => removing harware dependencies
  - OS and App constitute an unity -> encapsulation
  - The isolation of errors or security issues
  - The HW independent environment may be provided anywhere
  - Flexibility in choosing the best suited OS for the required application

Virtualization

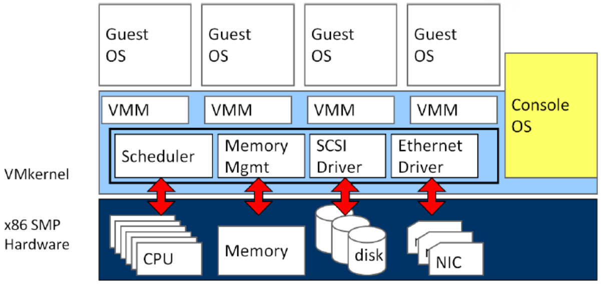
VMWare – Virtualization stack



Virtualization

VMware

- The desktop editions (VMware Workstation, VMWare Fusion, VMWare Player) – require an OS to be installed
- The server editions
  - VMware ESXi Server
    - *bare metal* virtualization solutions
    - Enables high performance



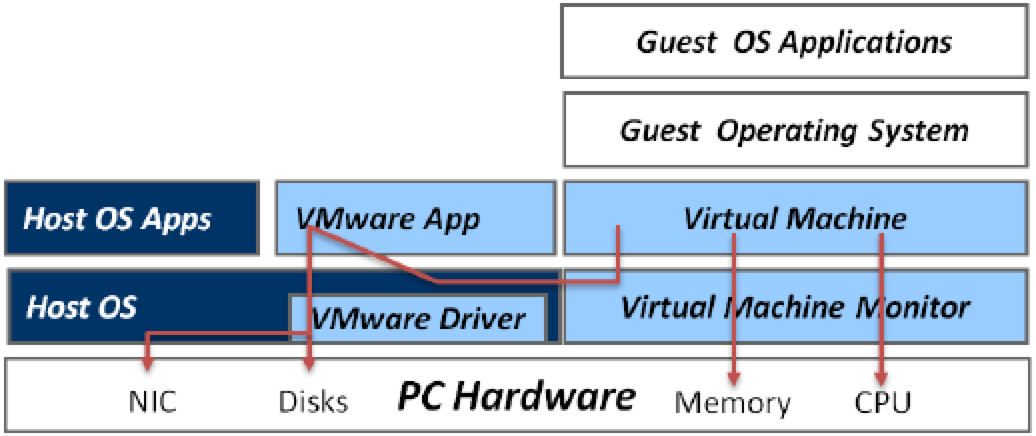
  - VMWare GSX Server (or VMware Server)
    - is a virtualization layer that run on host OS (Linux or Windows), use Host OS drivers and can have same limitation of the host OS environment
    - simplifies computing infrastructure by partitioning and isolating servers in secure and transportable virtual machines, each of which can run standard Windows, Linux or applications

Virtualization

VMware

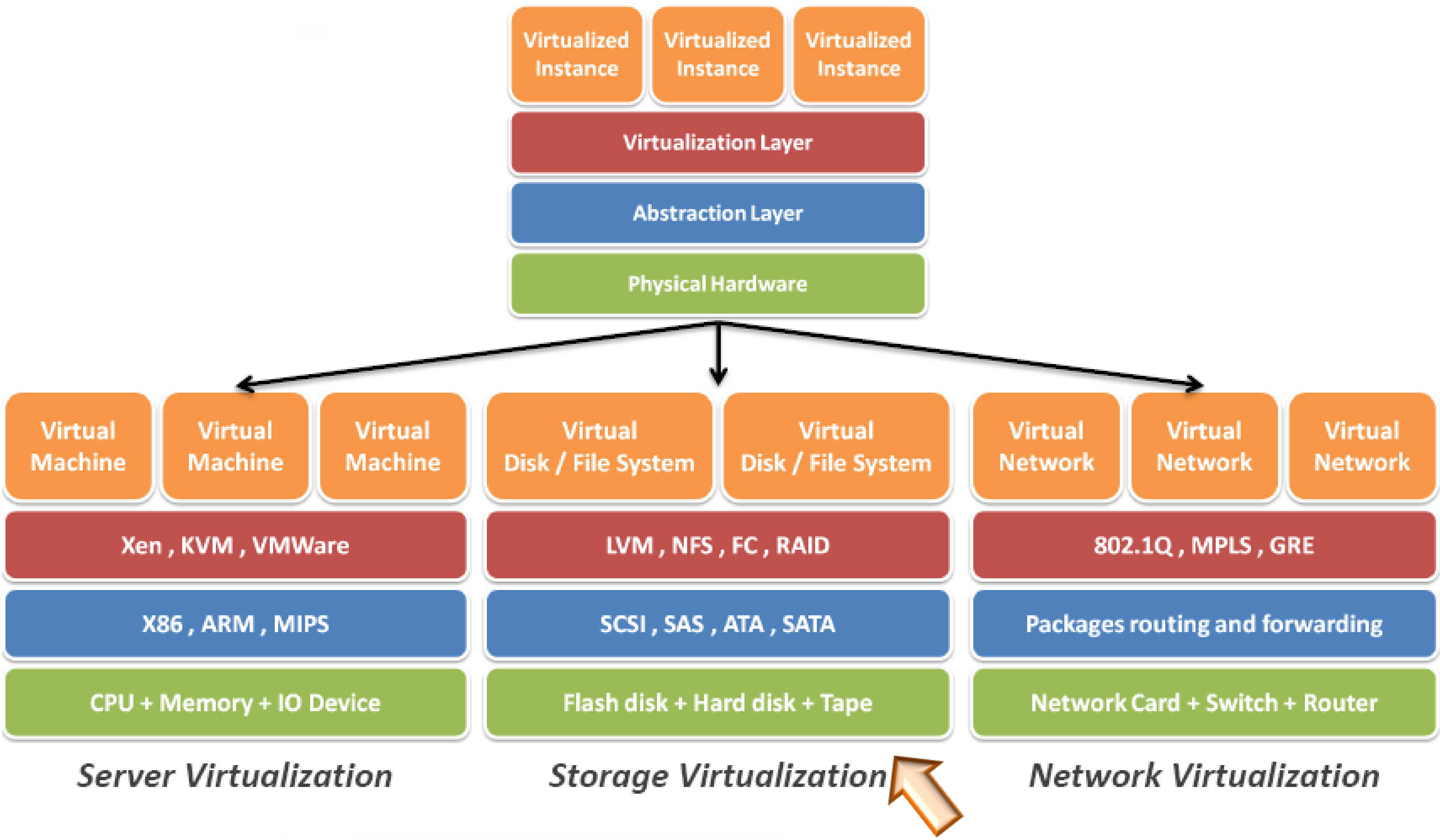
- GSX Server or VMware Server
  - Host Mode – the VMware behaves as an application and uses the host for accessing devices: HDD, NIC,…
  - VMM Mode – enables each *OS guest* to directly access the processor (*direct execution*)
  - => VMWare provides
    - Native execution speed
    - Support for a broad range of devices

Uses ex: Google Cloud VMware Engine deploys private clouds with the following VMware stack components: **VMware ESXi: hypervisor on dedicated nodes**

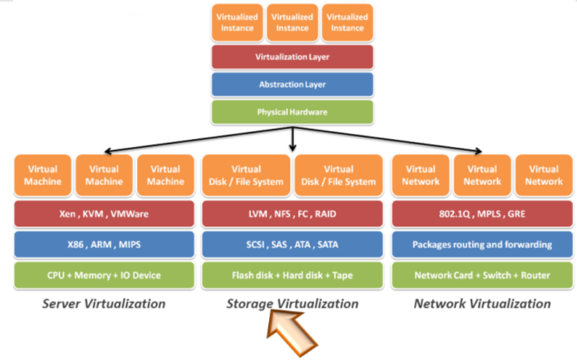


Virtualization

Virtualization techniques

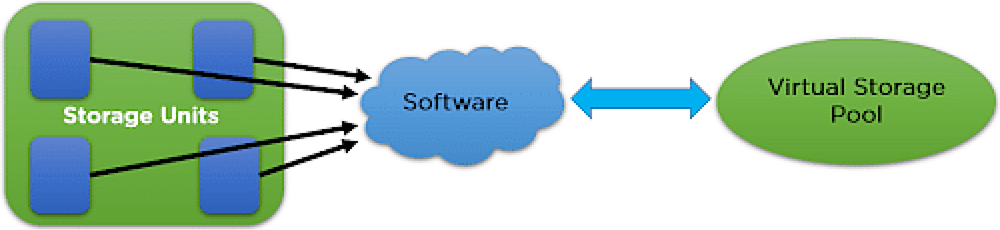


Virtualization



Storage Virtualization

This virtualization enables all the storage devices on the system to be accessed and be managed as a single storage unit pool

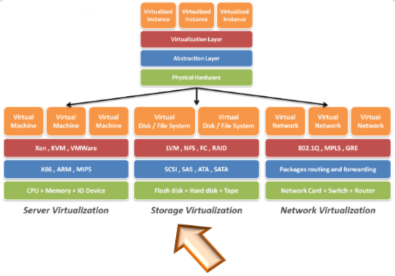


Pooling: Combining storage capacity from multiple (often disparate) storage devices, whether they're disk arrays, network-attached storage (NAS) systems, or other forms of storage

Abstraction: The details of the physical storage, such as location, are hidden from the users. Applications and users interact with a virtual representation of the storage instead of the actual physical devices

Management: Administrators can manage storage volumes, allocate space, and optimize storage utilization from a single interface, without worrying about the physical location of the storage

Virtualization



Storage Virtualization

Techniques employed to efficiently manage and present storage resources:

Block-Level Virtualization: It operates at the block level and involves abstracting block storage, like SAN (Storage Area Network) volumes, making multiple storage devices appear as a single pool of storage.

Examples: useful for database apps, …

File-Level Virtualization: This abstracts file systems from clients, allowing for files to be moved around on different servers or even different storage systems without disrupting the clients accessing them.

Examples: useful for efficient file management

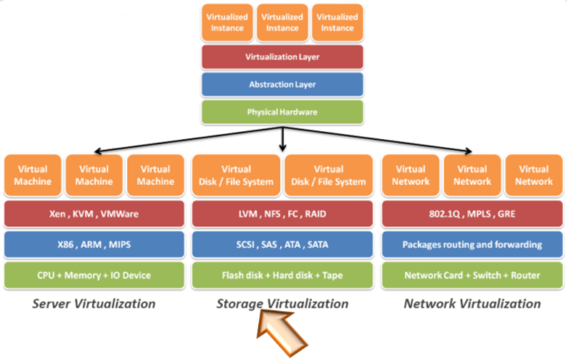
Distributed File Systems and Object Storage: Systems like Ceph or GlusterFS distribute and replicate data across different physical storage devices, but present it as a single, scalable filesystem.

Examples: useful for big data - Apache Hadoop HDFS (for distributed file systems), Ceph and MinIO (for object storage), GFS, …

Software-Defined Storage (SDS): This approach abstracts storage resources and manages them with software, decoupling the storage capabilities from the underlying hardware. SDS often includes automated management features like provisioning, deduplication, replication, and snapshots.

Examples: useful for an efficient use of storage resources - VMware vSAN, Nutanix Acropolis, and OpenStack Cinder

Virtualization



Storage Virtualization

Techniques employed to efficiently manage and present storage resources:

Thin Provisioning: Allocating storage capacity dynamically on an as-needed basis rather than pre-allocating storage, which can result in underutilized resources.

Examples: This feature is available in many storage arrays, like those from Dell EMC, HPE Nimble, and Pure Storage

Storage Hypervisors: Similar to server hypervisors, these provide a virtualization layer that manages and presents storage resources, potentially from different and disparate storage devices.

Examples: DataCore SANsymphony and StarWind Virtual SAN are often referred to as storage hypervisors

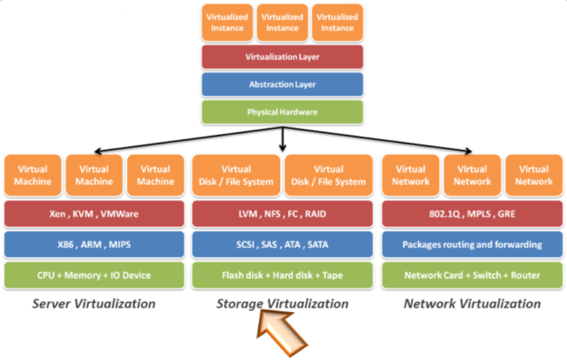
Cloud Storage Gateways: These are devices or services that translate cloud storage APIs such as REST to traditional storage protocols like iSCSI or Fibre Channel so that applications can use cloud storage without modification.

Examples: AWS Storage Gateway, Microsoft StorSimple, and NetApp Cloud Volumes ONTAP

Hybrid Cloud Storage Models: Techniques that integrate on-premises storage with cloud storage services, providing seamless movement of data between local and cloud environments.

Examples: Microsoft Azure StorSimple (integrates on-premises appliances with Azure cloud storage), Google Anthos (for managing data across on-premises and Google Cloud storage), and IBM Spectrum Virtualize for Public Cloud

Virtualization

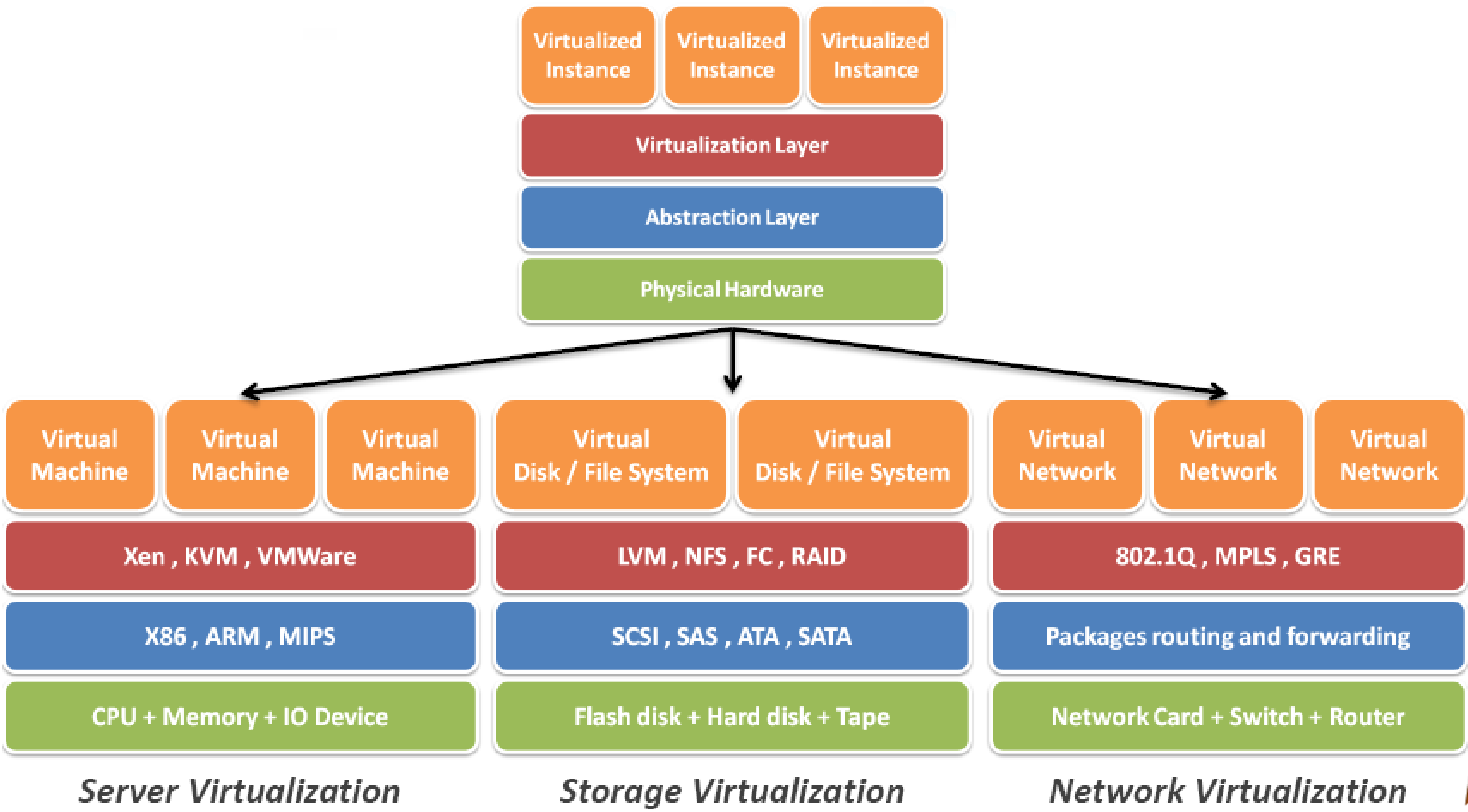


Storage Virtualization

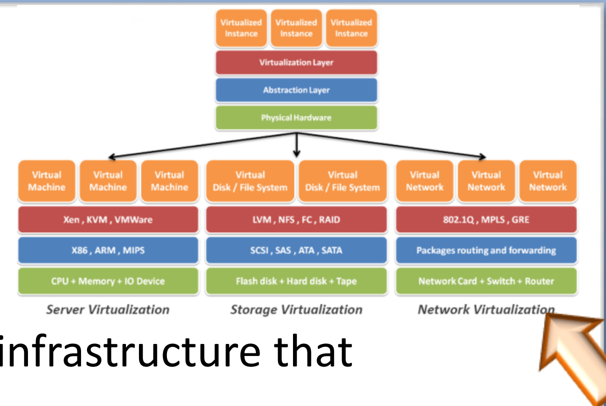
https://www.youtube.com/watch?v=O0D8Ftc44ls

Virtualization

Virtualization techniques



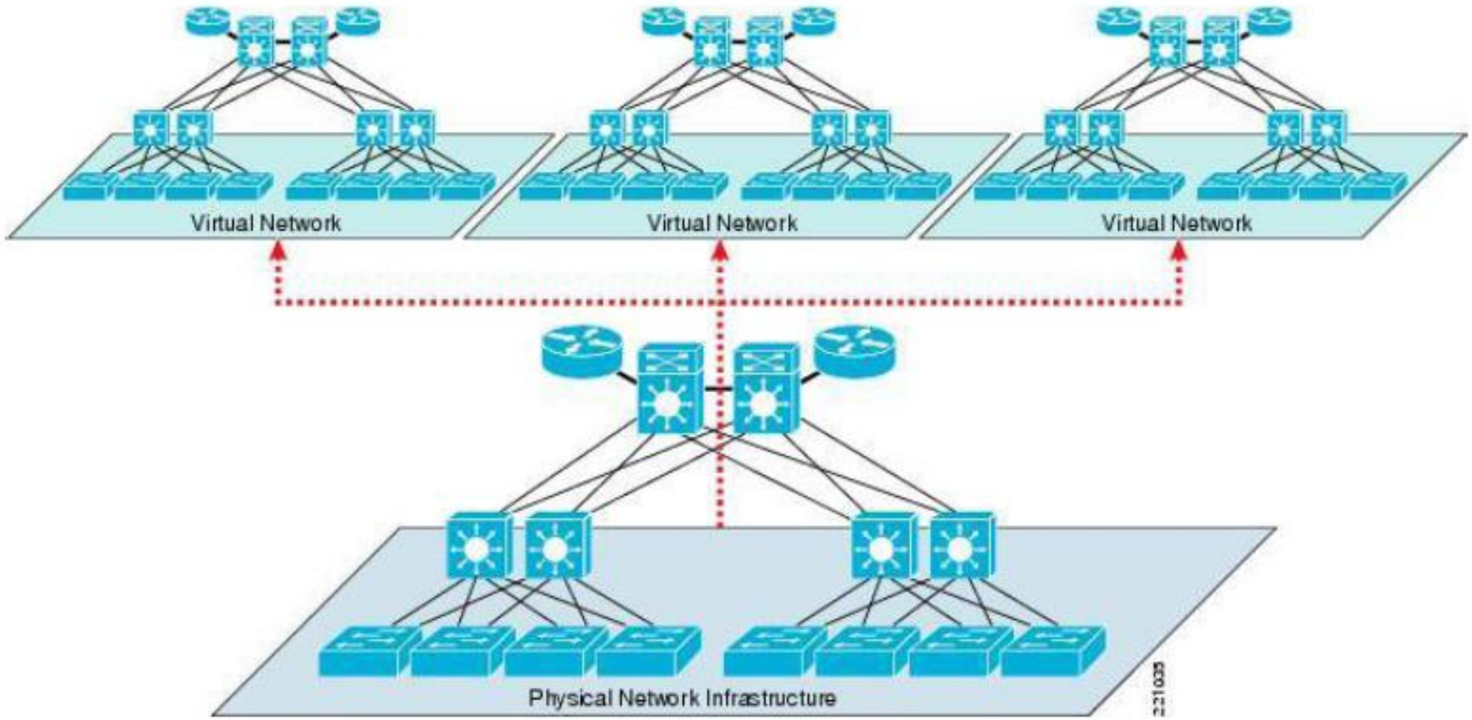
Virtualization



Network virtualization

Network virtualization involves creating a virtualized network infrastructure that operates independently of physical hardware.

It allows multiple virtual networks to coexist on a single physical network, improving flexibility, scalability, and efficiency.



Virtualization

Network virtualization

•Foundational Implementations and Roles:

- **802.1Q** implements *data-path virtualization* from hop to hop, allowing multiple virtual networks to coexist on the same physical network infrastructure
  - Each Ethernet frame is tagged with a VLAN ID, enabling network devices to distinguish between traffic for different virtual networks
- ***MPLS (Multiprotocol Label Switch)*** – implements the virtualization level for *switch* and *router* introducing labels to steer packets through predefined paths across the network.
  - This facilitates efficient management of traffic flows and enables the creation of virtual private networks (VPNs) with improved performance and reliability.
  - MPLS is extensively used in service provider networks and large enterprise networks.
- ***GRE (Generic Routing Encapsulation)*** – is a tunneling protocol that encapsulates packets to facilitate the creation of virtual point-to-point links over heterogeneous networks. This allows for the extension of network virtualization across different network infrastructures, including the internet, which is instrumental in building secure VPNs.

Virtualization

Network virtualization

- **Principles of external and internal virtualization:**
  - Virtualization - external
    - Several networks or network parts are adjoined into a virtual unity
    - Mechanism:
      - Level 1: virtualization is seldom used on this level
      - Level 2 Virtualization (e.g., VLANs): At this level, MAC address modifications enable network segmentation and the creation of isolated broadcast domains. This is critical for limiting broadcast traffic and enhancing security within a network.
      - Level 3 Virtualization (e.g., VPNs): Utilizes tunneling techniques to create secure, encrypted connections over a public network, effectively virtualizing the network by extending private network services over a wide area.
      - Upper Levels Virtualization: Involves creating networks that span various applications, such as peer-to-peer networks, which abstract away from the underlying network infrastructure to provide application-specific connectivity.

Virtualization

Network virtualization

•**Principles of external and internal virtualization:**

- Virtualization - internal
  - Mechanism
    - Level 1: the hypervisor does not require to emulate the physical level
    - Level 2 Devices in Hypervisor (e.g., virtual switches): The hypervisor emulates network devices like switches and bridges, enabling virtual machines on the same host to communicate as if connected to a physical network device.
    - Level 3 Devices in Hypervisor (e.g., virtual routers): Similar to level 2, but for higher-layer devices such as routers, allowing for more complex networking functionalities, including inter-VLAN routing, to be implemented within the virtualized environment.
    - Upper Levels: The guest operating systems can participate in network virtualization, running software-based network functions and services, further abstracting from the physical network.

Virtualization

Network virtualization

Key concepts:

- **Virtual Network (VN):** A software-based network between virtual machines.
- **Virtual Network Function (VNF):** Refers to the implementation of network functions (such as firewalls, load balancers, and routers) in software that can run on one or more virtual machines (VMs) on top of the hardware networking infrastructure
- **Network Functions Virtualization (NFV):** NFV is an architectural framework that aims to virtualize network services traditionally run on proprietary, dedicated hardware. It's about the bigger picture of transforming how network services are deployed and managed, making the network more flexible and agile.
  - Obs. NFV provides the environment and infrastructure for VNFs to operate efficiently in a virtualized network setting
- **Software-Defined Networking (SDN):** SDN provides a way to programmatically initialize, control, change, and manage network behavior dynamically via open interfaces.
- **Network Slicing**: The process of creating multiple unique logical networks on a single physical network infrastructure. Each slice can have its own network characteristics and resources, tailored for specific types of traffic or services.

Virtualization

Network virtualization

•Integrating traditional network virtualization techniques with newer paradigms like Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) significantly enhances the scalability, flexibility, and efficiency of virtual networks

•**SDN** represents an innovative approach to networking that distinctly separates the network's decision-making aspect (control plane) from the component that executes these decisions by moving data packets (data plane)

- Centralizes Network Management: By centralizing control, SDN provides a network-wide view, making it easier to optimize traffic flows and manage resources dynamically
- Increases Flexibility: Administrators can adjust network rules and policies on the fly, adapting to changing needs without altering the physical infrastructure
- Enhances Programmability: Networks can be programmed directly through software applications using open APIs. This is particularly beneficial for automating tasks and integrating with cloud services

Virtualization

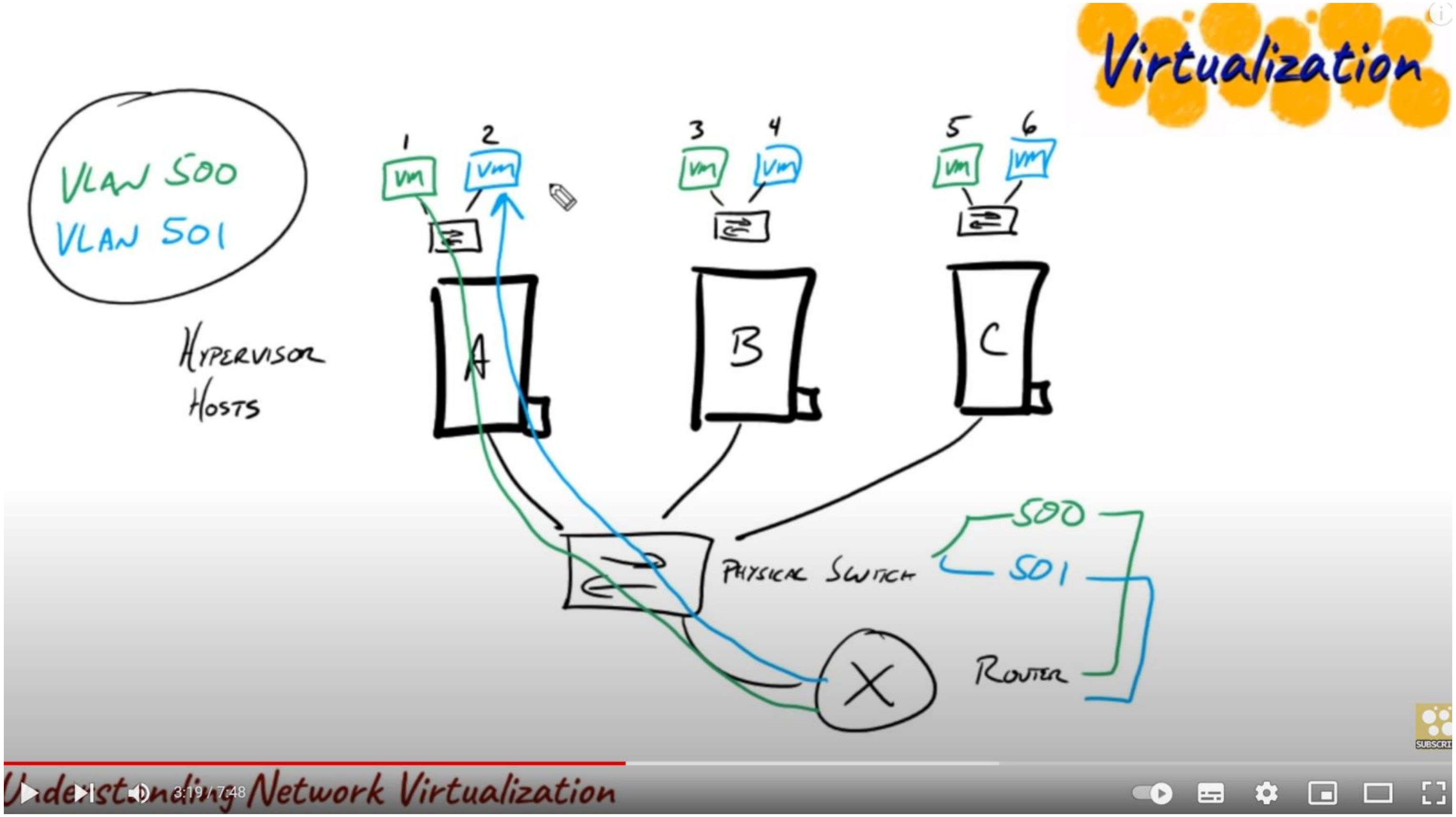
Network virtualization

NFV is a network architecture concept that uses virtualization technologies to manage core networking functions via software. NFV decouples network functions, such as routing, firewalling, and load balancing, from proprietary hardware appliances, so they can run in software on standardized hardware platforms

- **Reduces Hardware Dependence**: By virtualizing services that traditionally required dedicated hardware, NFV lowers capital expenditure and operational costs
- **Improves Scalability and Flexibility**: Network services can be deployed and scaled as virtual instances, allowing for rapid adjustment to demand
- **Facilitates Service Innovation**: With a faster and less costly deployment model, service providers can experiment with and roll out new services more quickly

Virtualization

Network virtualization



https://www.youtube.com/watch?v=u0TgGIn2LIM

Virtualization

Network virtualization

Solutions:

VMware NSX: Offers a full suite of network virtualization capabilities for the data center. It allows for the creation of entire networks in software and embeds them in the hypervisor layer, abstracted from the underlying physical hardware.

Cisco ACI (Application Centric Infrastructure): Integrates software and hardware with the intent to offer a solution that manages the network as a pool of resources, providing a centralized, automated approach with policy-driven management.

Open vSwitch (OVS): An open-source project that provides a multilayer virtual switch, which is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols.

Juniper Contrail: Provides software-defined networking (SDN) solutions to create scalable and agile networks. It is designed to automate and orchestrate virtual networks that connect virtual and physical devices.

Virtualization

Network virtualization

Solutions:

Microsoft Hyper-V Network Virtualization (HNV): Allows for multiple virtual networks, potentially with overlapping IP addresses, to coexist on the same physical network, which is particularly useful in multi-tenant environments like those in cloud services.

Amazon Web Services Virtual Private Cloud (AWS VPC): Lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

Azure Virtual Network: Enables Azure resources to securely communicate with each other, the internet, and on-premises networks in a high-availability and scalable manner.

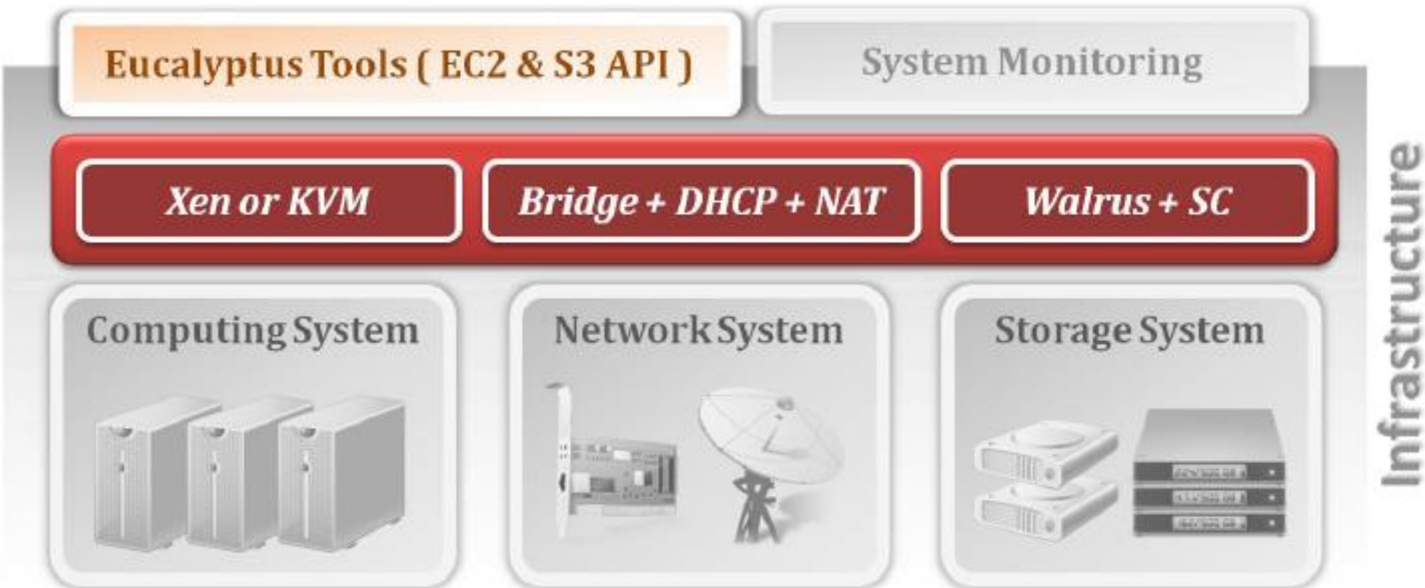
Google Cloud Virtual Networks (VPC): Provides a global, scalable, and flexible virtual network infrastructure within Google Cloud Platform, including services like load balancing, flexible addressing, and network security.

Terraform & Ansible for Network Automation: Although not network virtualization technologies per se, these Infrastructure as Code (IaC) tools are often used to automate the provisioning and management of virtualized network resources.

Virtualization

IaaS – case study

Eucalyptus - *"Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems"*



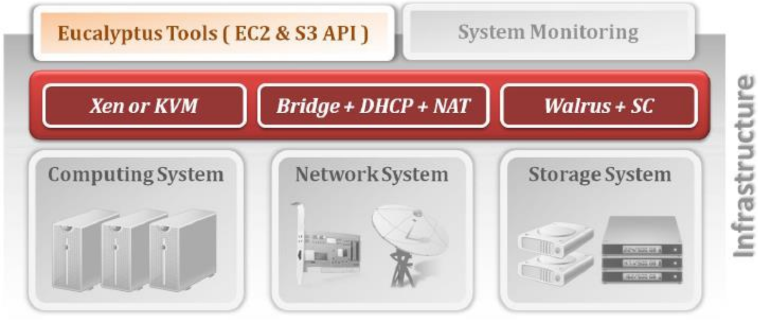
Virtualization

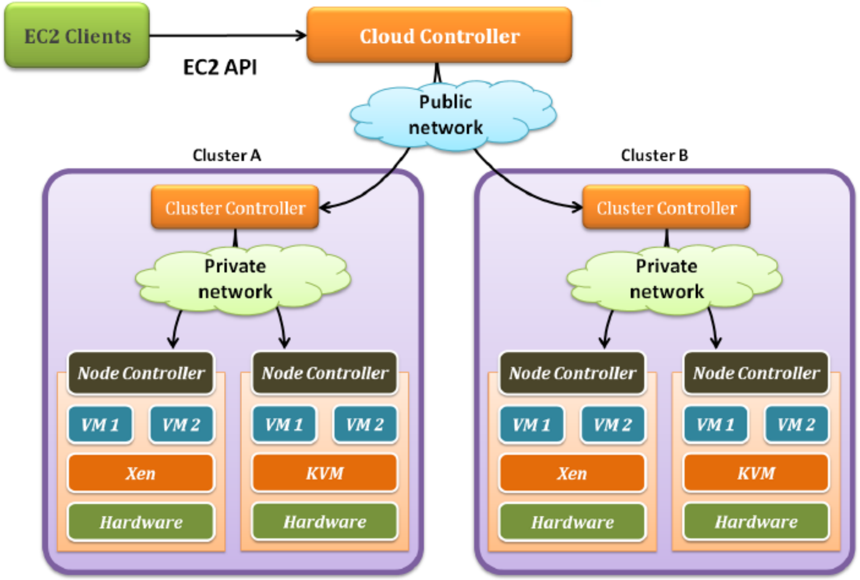
IaaS – case study: Eucalyptus

Server Virtualization

Components:

- CLC (Cloud Controller) – deserializes the user requests to clusters
- CC (Cluster Controller) – assigns the require resources for the virtual machine
- NC (Node Controller) – runs the user virtual machines





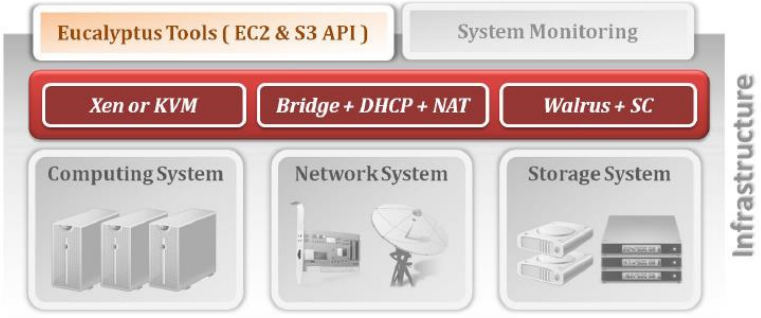
Virtualization

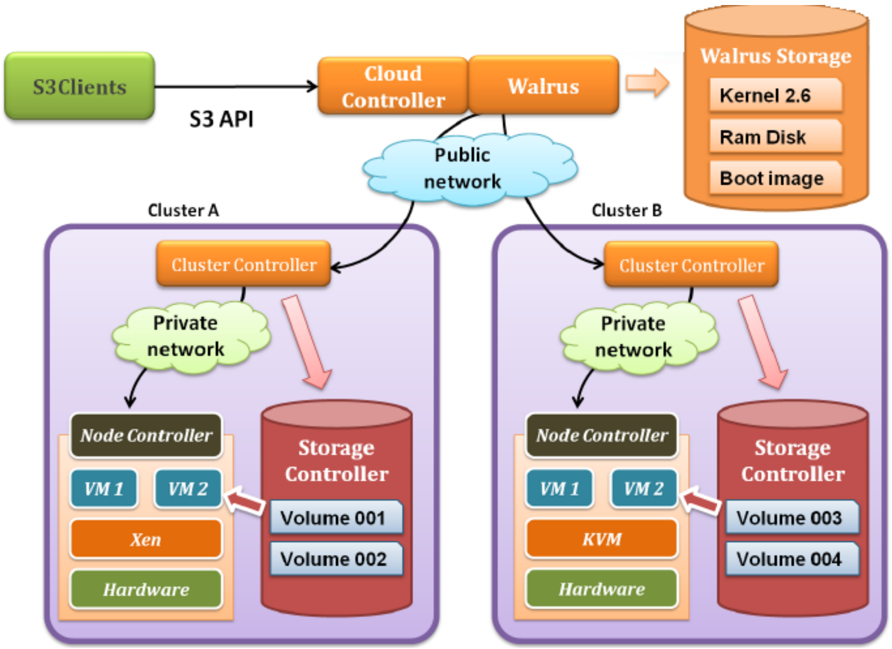
IaaS – case study: Eucalyptus

Storage Virtualization

Components:

- Walrus (compatible with S3) – stores images for VM booting
- Storage Controller
  - Stores, mainly, user created logical volumes, that may be attached to the specified virtual machines at run-time
  - Each SC in cluster is controlled by the CC, and each VM may use these logical volumes through the network





Virtualization

IaaS – case study: Eucalyptus

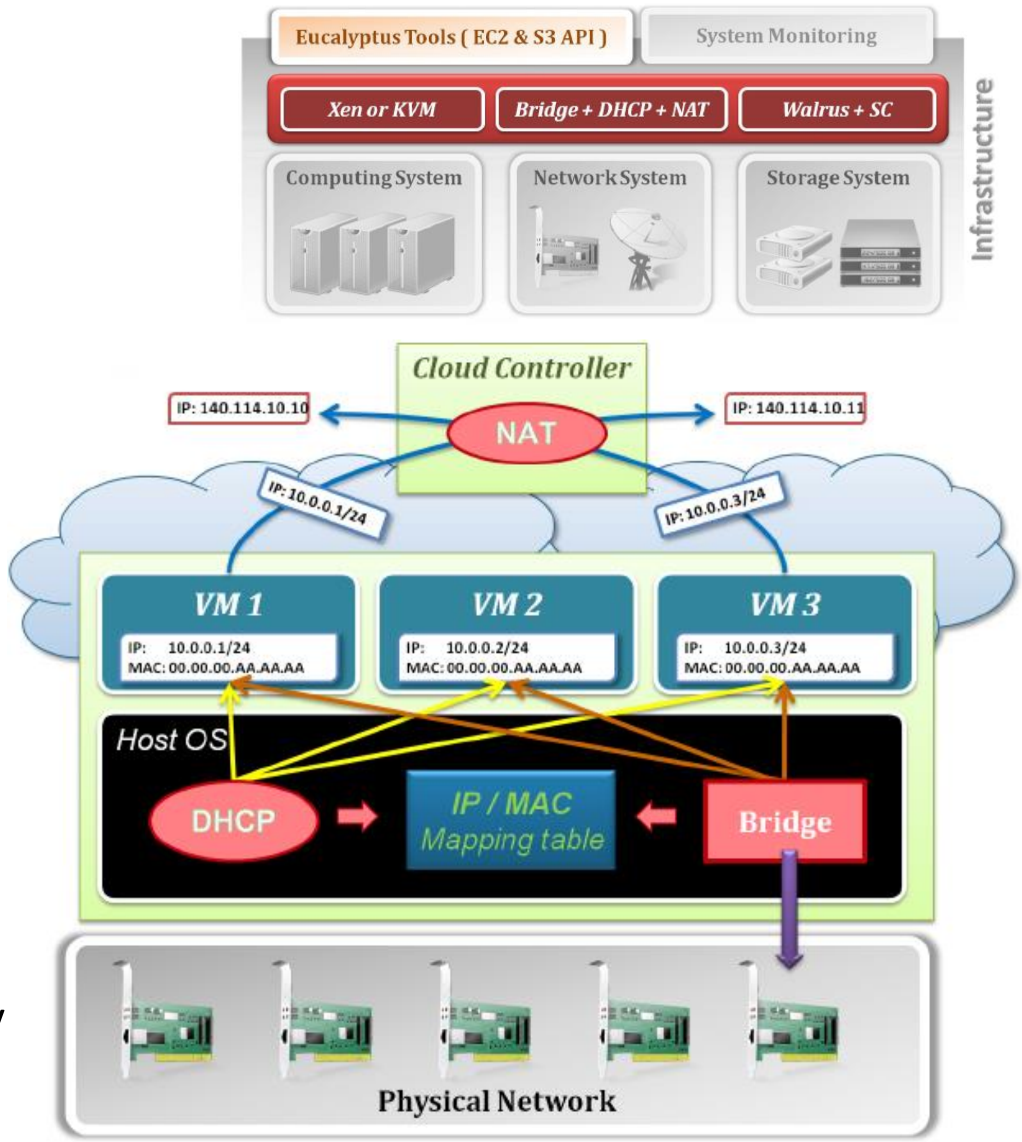
Network Virtualization

- Bridge (Virtual Switch)
  - Enables the virtual machines on a node to share the physical NIC
- DHCP

  Maps the MAC addresses of VMs at private LAN IPs

  NAT
  - Forwards packages to the public network

  (WAN)
- *IP/MAC mapping table*
  - The IP addresses are assigned by Eucalyptus
  - The MAC addresses are assigned by the hypervisor
  - This table is kept by Eucalyptus



Virtualization (see Slide 25)

Classification based on the **implementation approach** and the **level of hardware abstraction** they provide:

Operating System-Based Virtualization:

– ***OS-Level Virtualization***

– doesn’t involve hypervisors at all

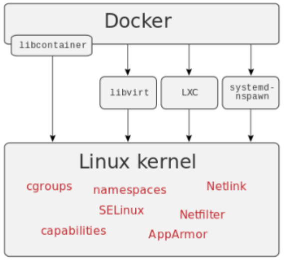
– it works by providing isolated user-space instances, which are called containers

– these containers share the same kernel of the host's operating system but have separate user spaces.

– they're more lightweight than full virtual machines because they don't need to emulate or simulate hardware and don't have the overhead of running separate kernel instances.

– Examples of OS-level virtualization: **Docker**, LXC (Linux Containers), or FreeBSD Jails (see Week 6)

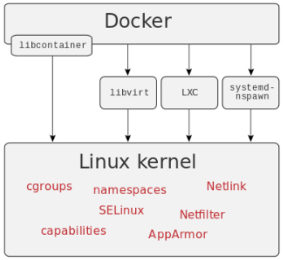
Docker



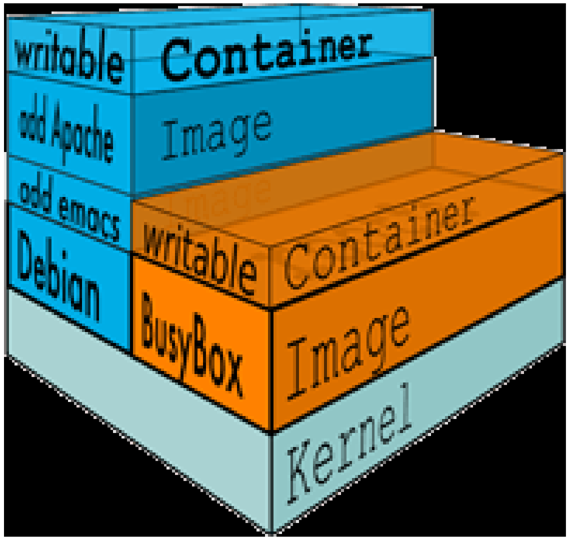
- The project ***dotCloud,*** July 2013 - Solomon Hykes, Andrea Luzzardi and Francois-Xavier Bourlet, Jeff Lindsay
- September 2013 – collaborations with Fedora, Red Hat Enterprise Linux and OpenShift
- October 2014 – Microsoft announced the integrations of a *Docker engine* in Windows Server
- "*Docker is a tool that can package an application and its dependencies in a virtual container that can run on any server. This helps enable flexibility and portability on where the application can run, whether on premise, public cloud, private cloud, bare metal, etc.*"

[http://en.wikipedia.org/wiki/Docker_%28software%29]

Docker

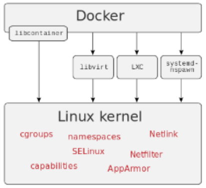


Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in.



[https://www.docker.com/what-docker]

Docker

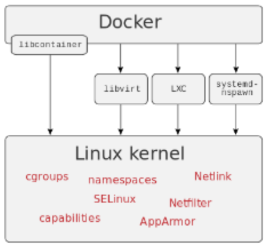


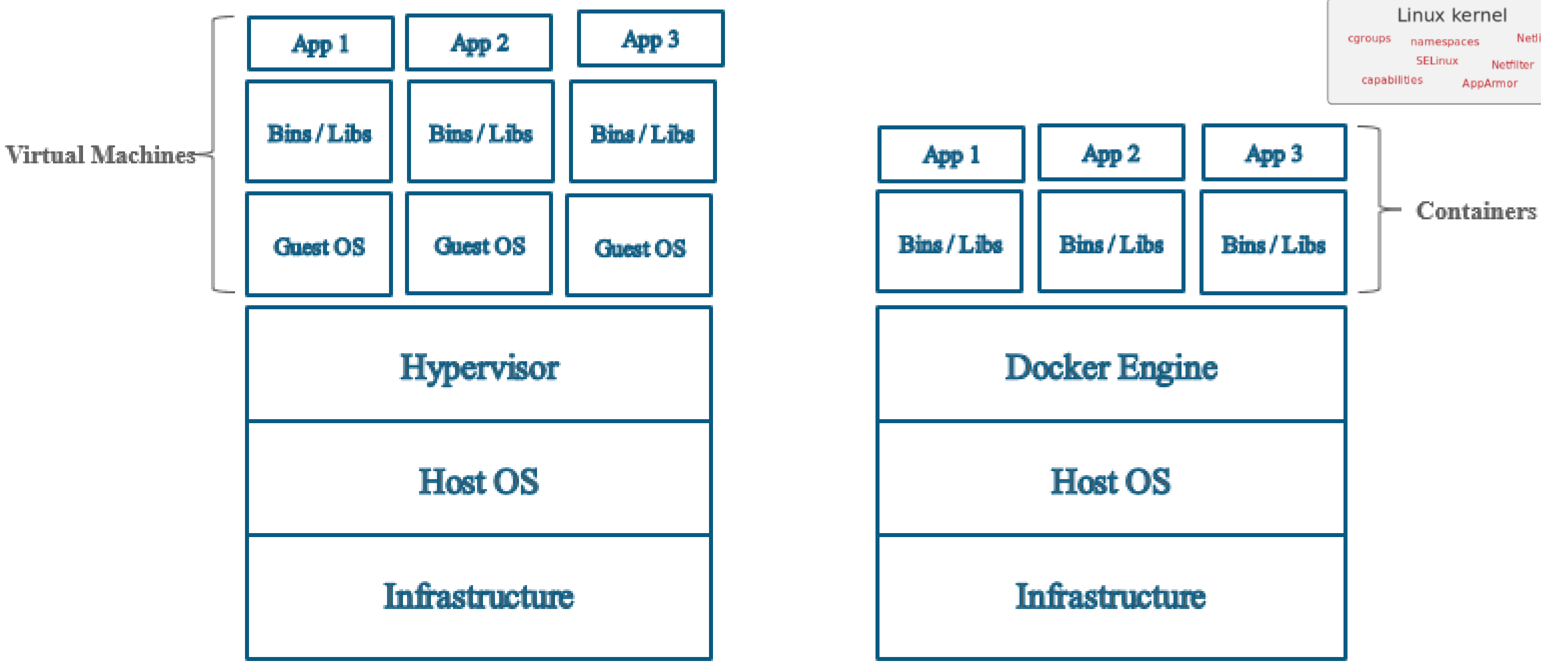
- Implements a high level API that enables the creation of containers ensuring isolated processed running
- It is built over the facilities provided by the operating system (initially Linux kernel) => does not require a discrete operating system, as is the case of virtualization

  => different containers are sharing the same kernel, but are limited by the available resources (CPU, memory, I/O)
- Docker also accesses various virtualization facilities through a series of libraries
- Pros: the containers' creation and management, supports seamless work with distributed systems (e.g. multiple applications, distributed tasks may autonomously run on a single or a range of virtual machines)

[http://sleekd.com/servers/docker-vs-virtualization]

Docker





- Fiecare masina virtuala include aplicatiile, librariile si guest OS => zeci de GB

- *Docker containers* - includ aplicatiile si dependentele dar partajeaza kernelul
- Ruleaza la procese izolate in host OS
- Nu sunt legate de o anumita infrastructura

Docker or Virtualization

So how do you go about deciding between VMs and containers anyway? Scott S. Lowe, a VMware engineering architect, suggests that you look at the "scope" of your work. In other words if you want run multiple copies of a single app, say MySQL, you use a container. If you want the flexibility of running multiple applications you use a virtual machine.

In addition, containers tend to lock you into a particular operating system version. That can be a good thing: You don't have to worry about dependencies once you have the application running properly in a container. But it also limits you. With VMs, no matter what hypervisor you're using -- KVM, Hyper-V, vSphere, Xen, whatever -- you can pretty much run any operating system. Do you need to run an obscure app that only runs on QNX? That's easy with a VM; it's not so simple with the current generation of containers.

[http://www.itworld.com]

Docker or/and Virtualization

- They are used together
- Many providers are running *bare-metal virtualization* technologies(e.g. XEN) and Docker that runs over a virtualized instance (e.g. Ubuntu)
- http://sleekd.com/servers/docker-vs-virtualization/
- http://www.serverwatch.com/server-trends/the-benefits-of-docker-vs.-server-virtualization.html
- http://searchservervirtualization.techtarget.com/feature/Docker-containers-virtualization-can-work-in-harmony
- https://tech.yandex.com/events/yac/2013/talks/14/
- …

Cloud Computing

IaaS

- *scalability and elasticity*
- *availability and reliability*
- *manageability and interoperability*
- *accessibility and portability*
- *performance and optimization*

Cloud Computing

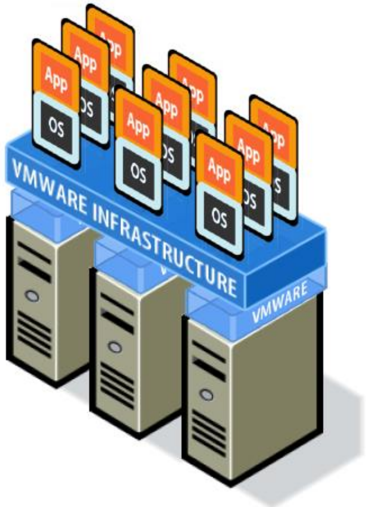
Scalability and *Elasticity* in IaaS

- The clients may dynamically increase or reduce required resources
- Providing large quantity of resources should be available for hours or days
- The system behavior is identical on large or small scale

Example: Scalability implemented by VMWare

- VMWare vMotion enables to move virtual machines, without application interruption

-> dynamic scaling of virtual machines via physical servers



Cloud Computing

Scalability
Elasticity

- **Dynamic provision**
- **Multi-tenant design**

Scalability and *Elasticity* in IaaS

- For computing resources:
  - On demand creation and termination of virtual machines
  - Hypervisors integration between all physical machines in order to manage all virtual machines
- For storage resources:
  - Dynamic storage space allocation and deallocation for the customer Integration of all physical resources in an IaaS system
- For communication resources:
  - On demand login and logout to a virtual network
  - Dynamic split of the network query flux to different physical routers in order to maintain proper bandwidth

Cloud Computing

Scalability
Elasticity

- Dynamic provision
- Multi-tenant design

***Scalability* and *Elasticity* in IaaS | Suppliers**

AWS EC2 Auto Scaling: Adjusts the number of Amazon EC2 instances in response to changing demand.

Azure Virtual Machine Scale Sets: Allows for the creation and management of a group of load balanced VMs.

Google Cloud Instance Groups: Manages groups of instances, providing auto-scaling depending on the load.

OpenStack Heat: Orchestration service that manages the infrastructure lifecycle for both horizontal and vertical scaling.

VMware vMotion: Enables the live migration of running virtual machines from one physical server to another with no impact on availability.

Kubernetes Horizontal Pod Autoscaler: Automatically scales the number of pods in a replication controller, deployment, or replica set based on observed CPU utilization.

Terraform: While not a direct scaling tool, it allows for infrastructure to be expressed as code and can be used to set up the necessary resources for scaling in the cloud.

VMware DRS (Distributed Resource Scheduler): Automatically balances computing workloads with available resources in a virtualized environment.

IBM Cloud Autoscale: Automates horizontal scaling to manage system load changes.

Rackspace Auto Scale: Automatically scales resources in response to the workload demand.

Cloud Computing

Availability
Reliability

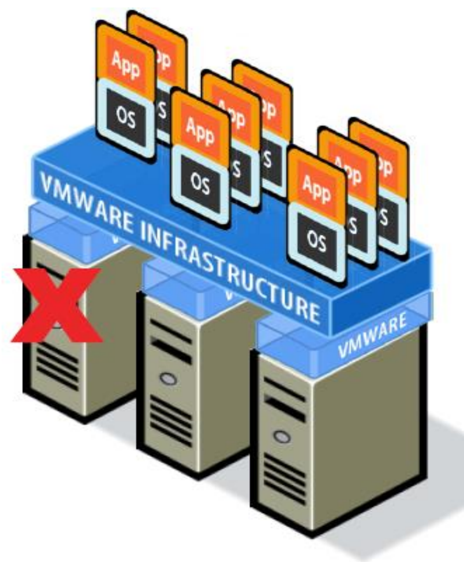
- Fault tolerance
- System resilience
- System security

Availability and Reliability in IaaS

•The client must access computing resources without worry about possible hardware errors

•The data stored in IaaS are available at any time, regardless of natural calamities

•The communication and storage capacity must be maintained regardless of physical equipment deficit



•Example: Availability implemented by VMWare

– In case error occurs VMWare allows automated restart on a different physical server

– VMWare Site Recovery Manager allows rapid transition from a *production site* to a *Disaster Recovery site*

Cloud Computing

Availability and Reliability in IaaS

- For computing resources:
  - Monitoring of each physical and virtual machine
  - Permanent *Backup* of virtual machines to enable recovery
  - Moving virtual machines on various physical machines to prevent potential failure
- For storage resources:
  - Keeping copies on different storage devices
- Scheduled data *Backup* of virtual storage devices and saving on different geographical locations
- For communication resources:
  - Building redundant connections in order to increase system's fail-proofness

Cloud Computing

Availability
Reliability

- Fault tolerance
- System resilience
- System security

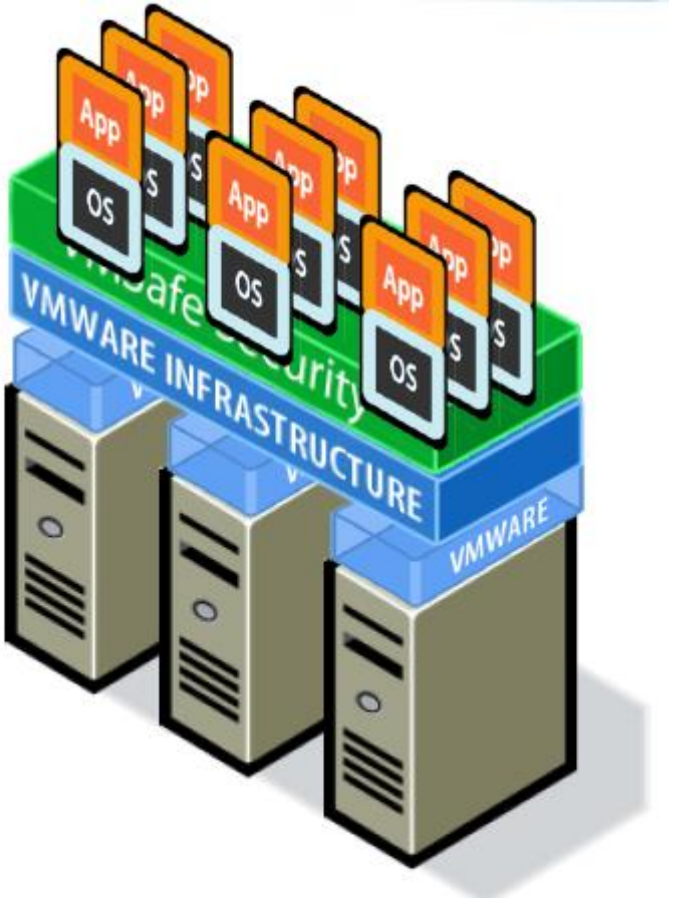
Availability and Reliability in IaaS

•Example:

VMSafe vService enables choosing a higher security level in a VMWare infrastructure

symantec. Checkpoint McAfee

TREND MICRO radware IBM



Cloud Computing

Availability
Reliability

- Fault tolerance
- System resilience
- System security

Availability and Reliability **in IaaS**

- Examples:

Amazon Web Services (AWS):

Elastic Load Balancing (ELB) distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses.

Amazon Route 53 effectively connects user requests to infrastructure running in AWS, such as EC2 instances, and can also be used to route users to infrastructure outside of AWS.

AWS Availability Zones are multiple, isolated locations within each AWS Region that allow you to run resources in physically separate locations for resilience.

Microsoft Azure:

Azure Load Balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances.

Azure Traffic Manager uses DNS to direct client requests to the most appropriate service endpoint based on traffic-routing methods and the health of endpoints.

Azure Availability Sets ensure that the VMs you deploy on Azure are distributed across multiple isolated hardware nodes in a cluster.

Cloud Computing

Availability and Reliability **in IaaS**

• Examples:

Google Cloud Platform (GCP):

Google Cloud Load Balancing distributes load-balanced compute resources in single or multiple regions, closer to the end-users, and enables you to put your resources behind a single anycast IP.

Google Cloud DNS is a scalable, reliable, and managed authoritative Domain Name System (DNS) service running on the same infrastructure as Google.

Google Compute Engine's live migration helps keep your virtual machines running even when a host system event occurs, such as a software or hardware update.

VMware:

VMware High Availability (HA) provides easy to use, cost-effective high availability for applications running in virtual machines.

VMware Fault Tolerance provides continuous availability for applications by creating a live shadow instance of a VM that mirrors the primary VM.

IBM Cloud:

IBM Cloud Load Balancer helps distribute traffic among your application servers residing locally within data center or across global data centers.

IBM Cloud Internet Services provides tools to manage DNS, monitor end-user performance, and implement firewall rules.

Cloud Computing

Manageability and Interoperability in IaaS:

- The client has total control over the allocated virtual infrastructure
- The virtualized resources may be automatically allocated as per per-configured policy
- The virtualized resources status must be permanently monitored
- The use of resources is recorded and subsequently converted by a *billing system*
- For computing resources:
  - Providing core operations on virtual machines: create, terminate, suspend, *snapshot*
- For storage resources:
  - Monitoring and recording of used space, as well as read/write access for each virtual storage resource
  - Automated allocation/ deallocation of physical space depending on the use level
- For communication resources:
  - Monitoring and recording of used bandwidth for each virtual connection
  - Automated data rerouting when computing or storage resources are duplicated

Cloud Computing

Manageability
Interoperability
- Control automation
- System monitoring
- Billing system

Manageability and Interoperability in IaaS:

In IaaS environments, manageability and interoperability are crucial for the seamless operation of services and applications across different infrastructures and platforms

- **Ansible**: An open-source automation platform for software provisioning, configuration management, and application deployment, which enhances manageability across diverse systems.
- **Terraform**: An infrastructure as code software tool that provides a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.
- **Kubernetes**: An open-source container orchestration platform that provides interoperability within a multi-cloud environment, ensuring that containerized applications run where and when you want, and helps manage them effectively.
- **OpenStack**: An open-source software platform for cloud computing, usually deployed as infrastructure-as-a-service, where virtual servers and other resources are made available to customers.
- **Apache CloudStack**: An open-source cloud management platform for delivering Infrastructure-as-a-Service (IaaS) in cloud computing environments. It helps manage and deploy cloud services by providing interoperability and manageability.

Cloud Computing

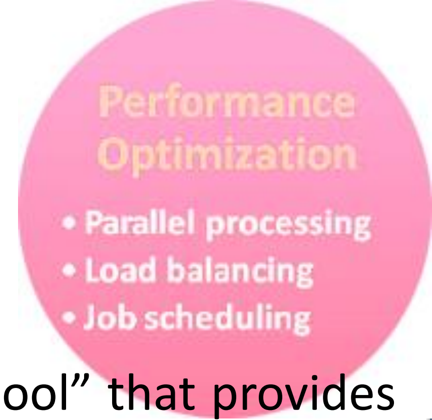
Manageability
Interoperability
- Control automation
- System monitoring
- Billing system

Manageability and Interoperability in IaaS:

In IaaS environments, manageability and interoperability are crucial for the seamless operation of services and applications across different infrastructures and platforms

- **VMware vRealize Operations**: Delivers self-driving IT operations management from apps to infrastructure to optimize, plan and scale VMware Cloud, whether deployed on-premises, hybrid, or multi-cloud.
- **Microsoft Azure Resource Manager**: Allows you to manage your Azure infrastructure and sets dependencies between resources so they're deployed in the correct order.
- **AWS CloudFormation**: Enables you to create and provision AWS infrastructure deployments predictably and repeatedly using a template file for resource management.
- **Chef**: A powerful automation platform that transforms infrastructure into code. This allows you to manage and automate more than just servers in your network.
- **Puppet**: It provides a standardized way to automate delivery and operation of the software no matter where it runs.

Cloud Computing



Performance and Optimization in IaaS

- The physical resources are gathered in a "large resource pool" that provides computing power for various processing types
- The virtual infrastructure will be dynamically configured for optimal physical resources usage

•For computing resources:

- "Creating" a virtual machine considering *load balancing*
- Real time migration of virtual machines between physical machines to achieve load balancing
  - Example: VMWare Distributed Resource Scheduler – performs load balancing

Cloud Computing

Performance and Optimization in IaaS

- For storage resources:
  - Real time migrating of virtual storage between physical machines to increase performance
- For communication resources:
  - Migrating virtual machines or storage in order to balance the network flow
  - VMWare and Cisco (Nexus 1000V) cooperate to reduce mobility load and increase management simplicity

Cloud Computing

Performance and Optimization in IaaS

•Tools that help with performance and optimization in IaaS:

AWS Trusted Advisor: Analyzes your AWS environment and provides recommendations for saving costs, improving system performance, and closing security gaps.

Azure Advisor: Provides personalized recommendations to optimize Azure deployments for performance, high availability, security, and cost.

Google Cloud Operations Suite (formerly Stackdriver): Offers monitoring, logging, and diagnostics to ensure optimal performance and efficient resource utilization.

VMware vRealize Operations Manager: Automates operations management using AI and predictive analytics, providing visibility and insights to optimize performance and capacity.

Dynatrace: A monitoring tool that provides full-stack insights using advanced observability across the infrastructure, allowing for automatic performance optimization.

Cloud Computing

Performance and Optimization in IaaS

•Tools that help with performance and optimization in IaaS:

Datadog: A monitoring service for cloud-scale applications, providing insights into servers, databases, tools, and services through a SaaS-based data analytics platform.

New Relic: Provides full-stack observability and analytics to help engineers improve software performance.

SolarWinds: Offers a range of performance monitoring and management tools designed for various IT infrastructure components.

Nagios: An open-source monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes.

LogicMonitor: A cloud-based performance monitoring platform for modern IT infrastructure, providing comprehensive visibility and actionable data.

Cloud Computing

Performance Optimization

- Parallel processing
- Load balancing
- Job scheduling

Performance and Optimization in IaaS

•**Job scheduling** is a critical aspect of managing and automating operations in IaaS environments:

AWS Batch: Enables developers, scientists, and engineers to easily and efficiently run hundreds of thousands of batch computing jobs on AWS.

Azure Automation: Offers job scheduling capabilities, among other automation features, allowing you to create, monitor, manage, and deploy resources in your Azure environment.

Google Cloud Scheduler: A fully managed cron job scheduler that allows you to run batch jobs on a recurring schedule.

Apache Airflow: An open-source platform to programmatically author, schedule, and monitor workflows. It integrates well with cloud services and local data centers.

Jenkins: An open-source automation server used for continuous integration and delivery, it can schedule and manage jobs for version control and deployment.

Kubernetes CronJobs: For scheduling time-based jobs in a Kubernetes environment, this feature allows you to run jobs on a repeating schedule.

Cloud Computing

Accessibility Portability

- Uniform access
- Thin client

Accesibility and Portability in IaaS

•Clients must gain access and control to the infrastructure without locally installing software or use a special hardware device

•The infrastructure resources must be easily reallocated or duplicated

•For computing resources:

– The Cloud provider offers a Web portal for resources management

– Complies with virtual machines standards to ensure portability

•For storage resources:

– The Cloud provider offers a Web portal for storage resources management

•For communication resources:

– The Cloud provider offers a Web portal for management and access of communication resources

Virtualizare

"A cloud's servers need to be virtualized "

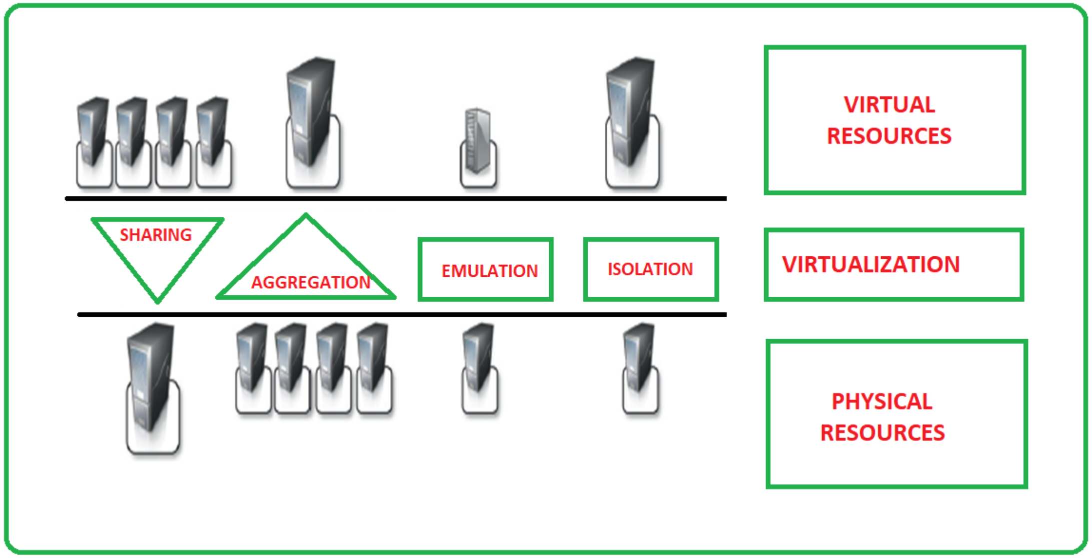
The virtualization impact on data centers:

- Users uncoupling – implementation:
  - Virtualization => users do not concern with physical servers or their location, they are entirely focused on the applications
- Fast resources providing (from months -> hours, minutes)
  - Inside an organization the acquiring, installation, and configuration may take between 60 to 90 days; a virtual server may be installed in several hours or even minutes depending on requirements ;
- The *pay-per-use* model*:*
  - The provider may not require payment for an entire server or for each server the software runs on. Payment is due depending on usage → a novel IT model

Virtualizare

"A cloud's servers need to be virtualized "

1. **Increased Security:** Virtual machines (VMs) provide a controlled environment for executing guest programs, isolating them and preventing harmful operations
2. **Managed Execution –** Virtualization includes sharing, aggregation, emulation, and isolation which are key for managed environments.



[https://www.geeksforgeeks.org/characteristics-of-virtualization/]

Virtualizare

"A cloud's servers need to be virtualized "

3. **Sharing –** Virtualization allows the creation of a separate computing environment within the same host. This basic feature is used to reduce the number of active servers and limit power consumption.
4. **Aggregation –** separate hosts can be combined and presented as a single resource via cluster management software.
5. **Emulation –** Enables the execution of guest programs that require specific host characteristics not physically present.
6. **Isolation**: Offers guests separate execution environments, protecting the host from harmful guest activities.
7. **Performance Tuning**: Advances in hardware and software allow for fine-tuning the performance of guest programs, supporting quality-of-service (QoS) features.
8. **Portability:** In hardware virtualization, VM images can be moved and executed on different machines, while in programming-level virtualization, code like JARs or assemblies can run on any VM without recompilation.

[https://www.geeksforgeeks.org/characteristics-of-virtualization/]

Virtualizare

"A cloud's servers need to be virtualized "

9. **Resource Sharing**: Multiple VMs can share the resources of a single physical machine.
10. **Flexibility**: IT administrators can easily manage VMs, adapting quickly to changes without new physical hardware.
11. **Hardware Independence**: VMs can run on different hardware and moved between servers easily.
12. **Scalability**: Resources can be scaled according to business needs.
13. **Centralized Management**: Simplifies monitoring and management of VMs.
14. **Disaster Recovery**: Simplifies implementation of disaster recovery by replicating VMs to remote sites.
15. **Testing and Development**: Facilitates creation of test environments without additional hardware.
16. **Energy Efficiency**: Reduces energy consumption by server consolidation.
17. **Increased Uptime**: Enables VM migration to avoid downtime during hardware failures.
18. **Cost Savings**: Reduces the need for physical hardware and operational costs.
19. **Improved Security Management**: Isolates applications and workloads, enhancing security.
20. **Cloud Migration**: Facilitates transition to cloud services by virtualizing infrastructure.

[https://www.geeksforgeeks.org/characteristics-of-virtualization/]

Bibliography

- Massimo Cafaro, Givani Aloisio, Grids, Clouds and Virtualization, 2011
- Katarina Stanoevska Slabeva, Thomas Wozniak, Grid and Cloud Computing - A Business Perspective on Technology and Applications, 2010, Editors Santi Ristol, Springer-Verlag Berlin Heidelberg
- Open Cloud Computing Interface - http://occi-wg.org/
- Lecture slides of “Virtual Machine” course (5200) in NCTU
- VmwareOverview Openline presentation slides http://www.openline.nl
- Xenpresentation http://www.cl.cam.ac.uk/research/srg/netos/papers/2006-xen-fosdem.ppt
- LustreFile System. http://www.oracle.com/us/products/servers-storage/storage/storage-software/031855.htm
- Logical Volume Management (LVM). http://www.tldp.org/HOWTO/LVM-HOWTO/
- TomClark, Storage Virtualization: Technologies for Simplifying Data Storage and Management, Addison Wesley Professional, 2005.
- Linux Bridge http://www.ibm.com/developerworks/cn/linux/l-tuntap/index.html
- Xennetworking http://wiki.xensource.com/xenwiki/XenNetworking
- VMware Virtual Networking Concepts http://www.vmware.com/files/pdf/virtual_networking_concepts.pdf
- TUN/TAP wiki http://en.wikipedia.org/wiki/TUN/TAP

Bibliography

- NIST (National Institute of Standards and Technology). http://csrc.nist.gov/groups/SNS/cloud-computing/
- http://www.ibm.com/developerworks/java/library/j-gaestorage/index.html?ca=drs-
- M. Armbrustet. al., “Above the Clouds: A Berkeley View of Cloud Computing,” Technical Report No. UCB/EECS-2009-28, University of California at Berkeley, 2009.
- James Smith and Ravi Nair, “Virtual Machines: Versatile Platforms for Systems and Processors”.
- Xen. http://www.xen.org/
- Kernel-based Virtual Machine (KVM). http://www.linux-kvm.org/page/Main_Page
- Xenproject http://www.xen.org
- KVM project http://www.linux-kvm.org/page/Main_Page

Bibliography

- IBM VirtIOsurvey https://www.ibm.com/developerworks/linux/library/l-virtio
- PCI-SIG IO virtualization specification http://www.pcisig.com/specifications/iov
- A. Menon, A. Cox, and W. Zwaenepoel. Optimizing Network Virtualization in Xen. Proc. USENIX Annual Technical Conference(USENIX 2006), pages 15–28, 2006.
- Multiple materiale si imagini au fost preluate de pe Internet
- https://www.idc.com/getdoc.jsp?containerId=prUS40960516
- https://www.docker.com/what-docker
- https://docs.docker.com/windows/step_one/
- https://wiki.xenproject.org/wiki/Xen_Project_Software_Overview
- https://www.vmware.com/products/vsphere.html
- Stoarge Virtualization: https://www.youtube.com/watch?v=O0D8Ftc44ls
- Network Virtualization: https://www.youtube.com/watch?v=hDgG34IFGp4&list=PPSV
- Virtual Networking Explained: https://www.youtube.com/watch?v=u0TgGIn2LIM

Abstract

- IaaS
  - Necessity
  - Definition
  - Example: Eucalyptus
- Virtualization
  - General overview
  - Concepts
  - Taxonomy
  - Examples
- Docker
- IaaS - proprieties

Cloud Computing
-IaaS-

Questions?

July 2024

Cloud Computing (from slide 10 ☺)

IaaS provides infrastructure for the cloud as a service, enabling a virtualized environment.

Virtualization stands for the techniques employed for creating an abstract level represented by logical resources, while based on physical resources.

The virtualization techniques that plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions:

•Computing resources

- *Virtual Machine techniques*

• Storage resources

- *Virtual Storage techniques*

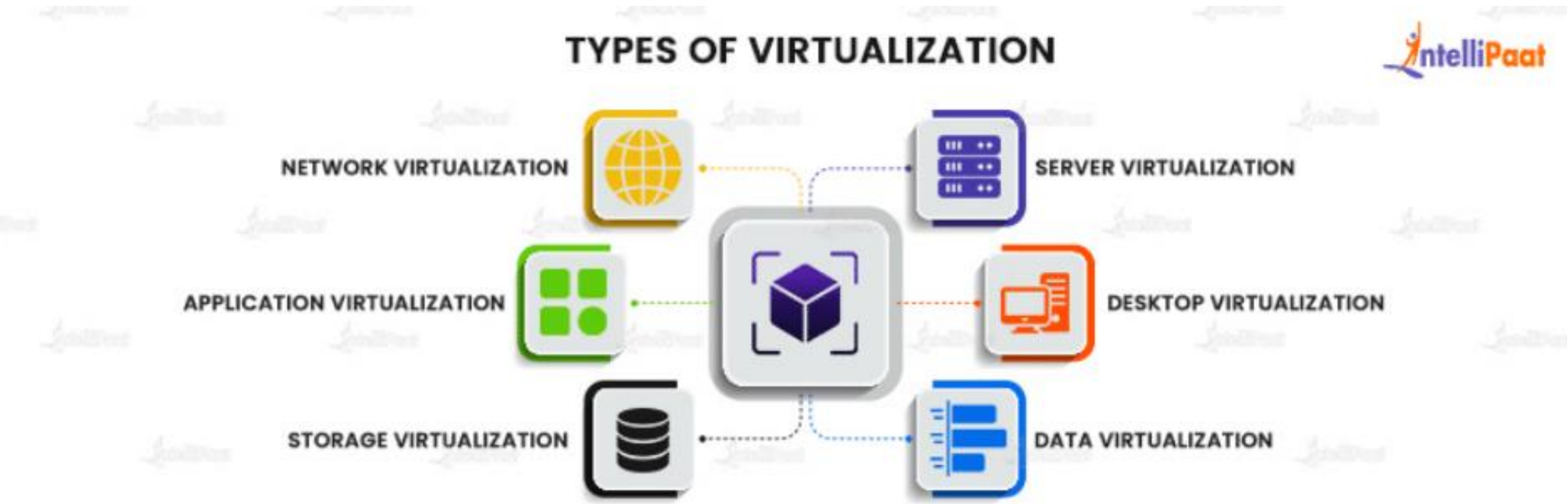
• Communication resources

- *Virtual Network techniques*

… *Application Virtualization, Desktop Virtualization, Data Virtualization, ….*

Cloud Computing (from slide 10 ☺)

… *Application Virtualization, Desktop Virtualization, Data Virtualization, ….*



https://intellipaat.com/blog/virtualization-in-cloud-computing/

Cloud Computing (from slide 10☺)

Application virtualization enables users to remotely utilize applications hosted on a server, which retains all user-specific data and settings. It is a process of virtualizing the applications that may or may not be on the guest OS which has been hosted. It isolates an application from the operating system on which it is running. It offers access to an application without requiring the target device to have it installed.

Desktop virtualization The benefits of desktop virtualization include user portability, user mobility, and simple application installation, updates, and patches. It is the ability to store the OS on a server, allowing the users to access their desktop virtually from any location and even from a different machine.

Data virtualization Data collection from numerous sources and management or manipulation done in one location fall under this virtualization.

It can be used to carry out a wide range of tasks, including data integration, business integration, service-oriented architecture, and data services.