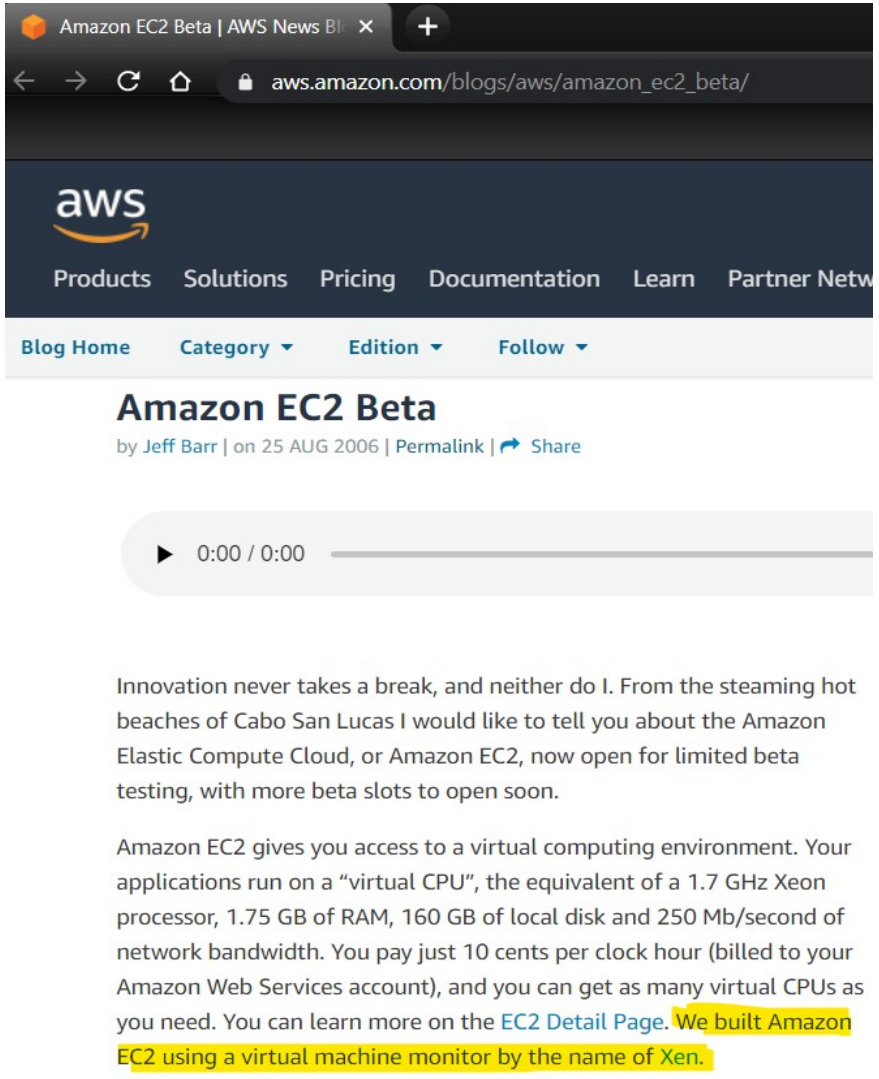
Virtualization

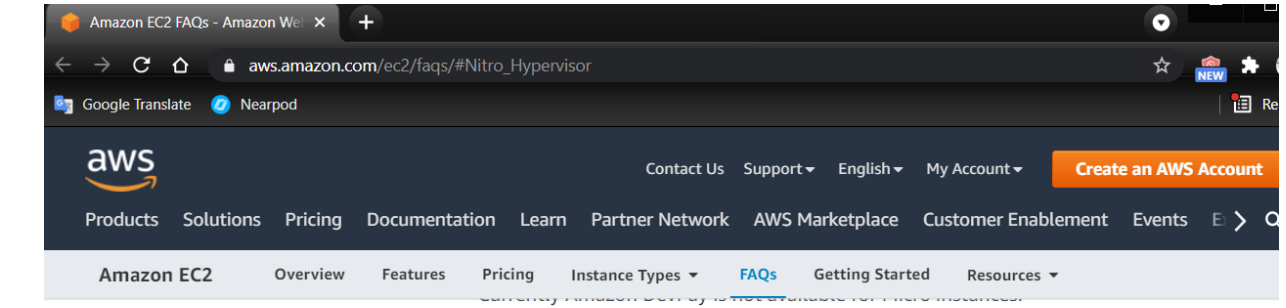
Concepts, types, KVM

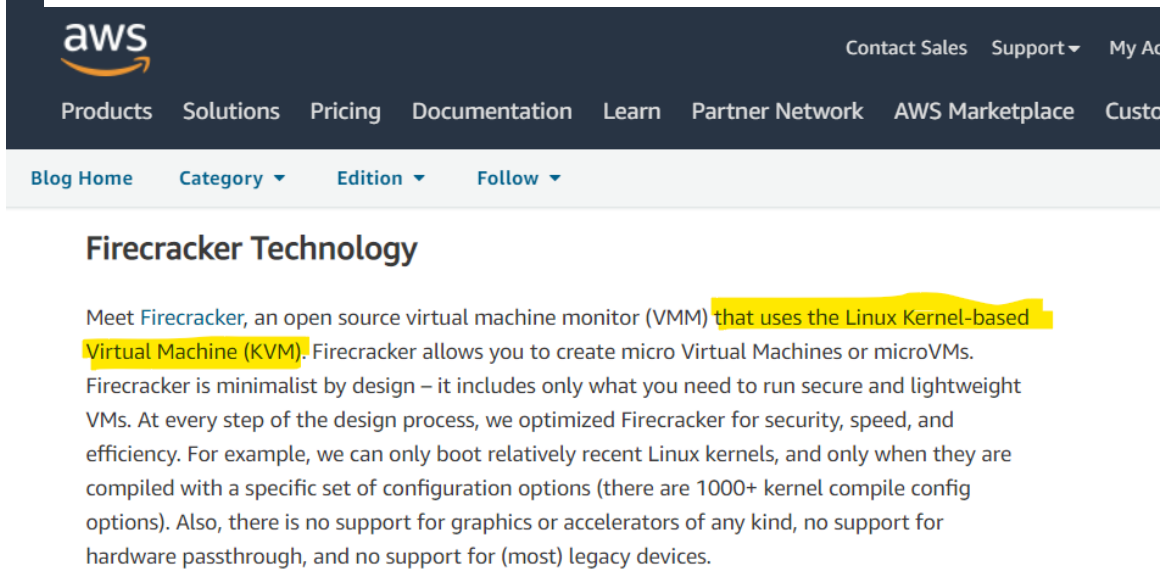
27 y 29 / 04 / 2021

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Motivation


Amazon EC2 Beta | AWS News Bl
aws.amazon.com/blogs/aws/amazon_ec2_beta/
aws
Products Solutions Pricing Documentation Learn Partner Netw
Blog Home Category Edition Follow
Amazon EC2 Beta
by Jeff Barr | on 25 AUG 2006 | Permalink | Share
0:00 / 0:00
Innovation never takes a break, and neither do I. From the steaming hot beaches of Cabo San Lucas I would like to tell you about the Amazon Elastic Compute Cloud, or Amazon EC2, now open for limited beta testing, with more beta slots to open soon.
Amazon EC2 gives you access to a virtual computing environment. Your applications run on a "virtual CPU", the equivalent of a 1.7 GHz Xeon processor, 1.75 GB of RAM, 160 GB of local disk and 250 Mb/second of network bandwidth. You pay just 10 cents per clock hour (billed to your Amazon Web Services account), and you can get as many virtual CPUs as you need. You can learn more on the EC2 Detail Page. We built Amazon EC2 using a virtual machine monitor by the name of Xen.


Amazon EC2 FAQs - Amazon We
aws.amazon.com/ec2/faqs/#Nitro_Hypervisor
Google Translate Nearpod
aws
Contact Us Support English My Account Create an AWS Account
Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events
Amazon EC2 Overview Features Pricing Instance Types FAQs Getting Started Resources
PAGE CONTENT
General
Instance types
Storage
Networking and security
Management
Billing and purchase options
Platform
Nitro Hypervisor
Q. What is the Nitro Hypervisor?
The launch of C5 instances introduced a new hypervisor for Amazon EC2, the Nitro Hypervisor. As a component of the Nitro system, the Nitro Hypervisor primarily provides CPU and memory isolation for EC2 instances. VPC networking and EBS storage resources are implemented by dedicated hardware components, Nitro Cards that are part of all current generation EC2 instance families. The Nitro Hypervisor is built on core Linux Kernel-based Virtual Machine (KVM) technology, but does not include general-purpose operating system components.


aws
Contact Sales Support My Ac
Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Custo
Blog Home Category Edition Follow
Firecracker Technology
Meet Firecracker, an open source virtual machine monitor (VMM) that uses the Linux Kernel-based Virtual Machine (KVM). Firecracker allows you to create micro Virtual Machines or microVMs. Firecracker is minimalist by design – it includes only what you need to run secure and lightweight VMs. At every step of the design process, we optimized Firecracker for security, speed, and efficiency. For example, we can only boot relatively recent Linux kernels, and only when they are compiled with a specific set of configuration options (there are 1000+ kernel compile config options). Also, there is no support for graphics or accelerators of any kind, no support for hardware passthrough, and no support for (most) legacy devices.

Contenido(updated)

- Virtualización
- Tipos de Virtualización
- Visión de la arquitectura KVM
- Libvirtd

Contenido

Parte 1:

- Virtualización
- Tipos de Virtualización
- Visión de la arquitectura KVM

Parte 2:

- Libvirtd
- Visión de la arquitectura Xen
- Virtualización de CPU, memoria, y dispositivos de I/O
- Clusters virtuales e gerenciamento de recursos
- Tarea

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

1 Virtualization

Contexto

Virtualization

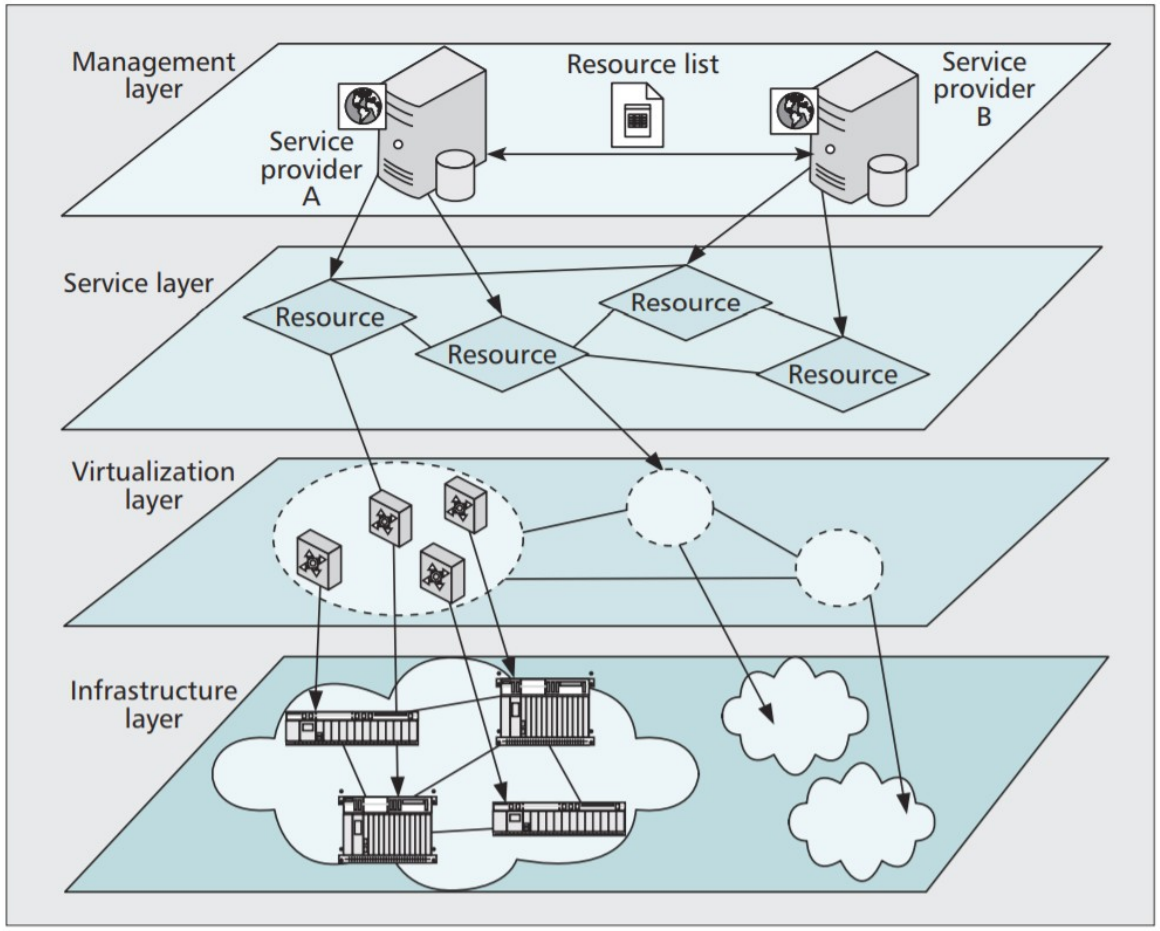


Figure 5. *Layers of IaaS service delivery model.*

Source: Enabling Infrastructure as a Service (IaaS) on IP Networks: From Distributed to Virtualized Control Plane.
Kim-Khoa Nguyen, et. al

Virtualization

- *“The goal of virtualization is to support portability, improve efficiency, increase reliability, and shield the user from the complexity of the system”*.
- The term virtualization is often synonymous with **hardware virtualization**, which plays a fundamental role in efficiently delivering IaaS solutions for cloud computing.
  - Virtualization Layer, Hypervisor, Virtual Machine Monitor(VMM), Virtualization Software

Virtualization

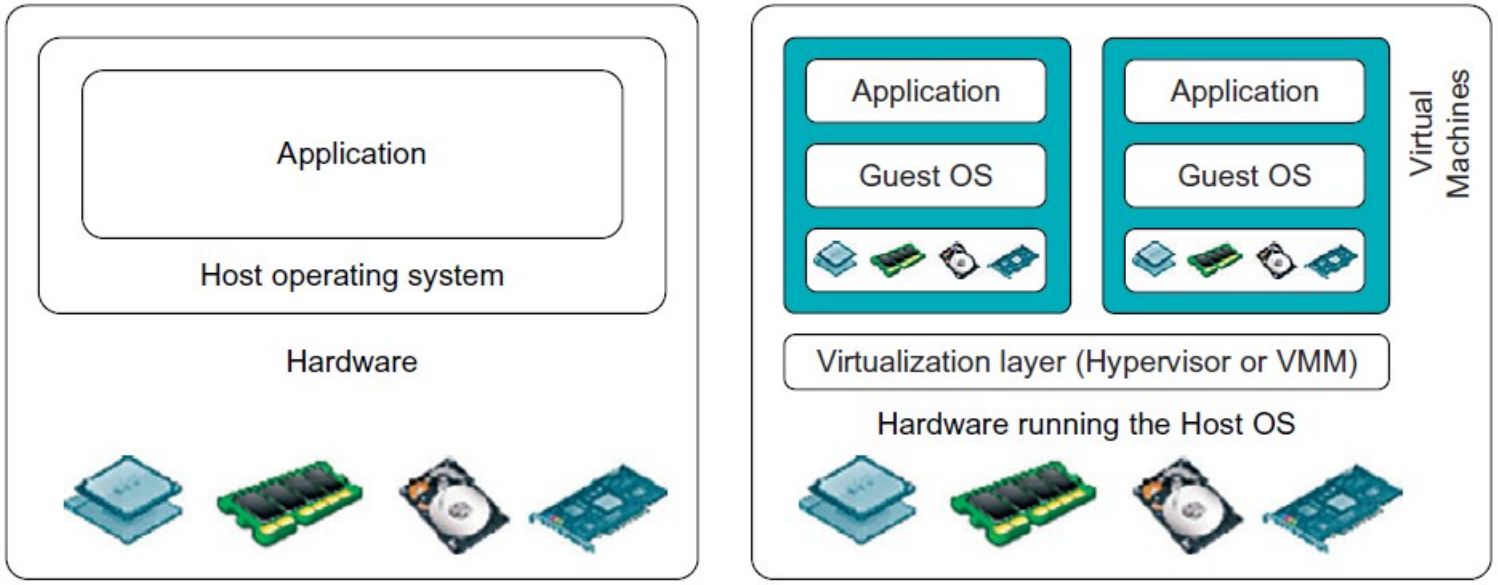


FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.

Virtualization

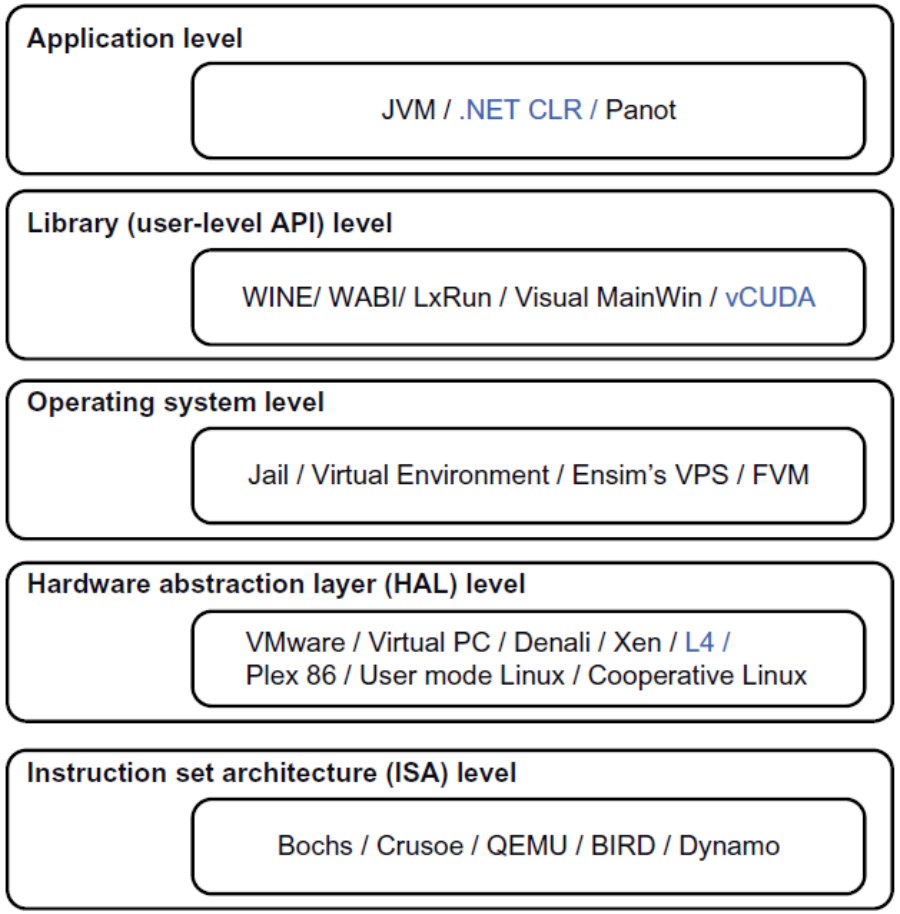


FIGURE 3.2

Virtualization ranging from hardware to applications in five abstraction levels.

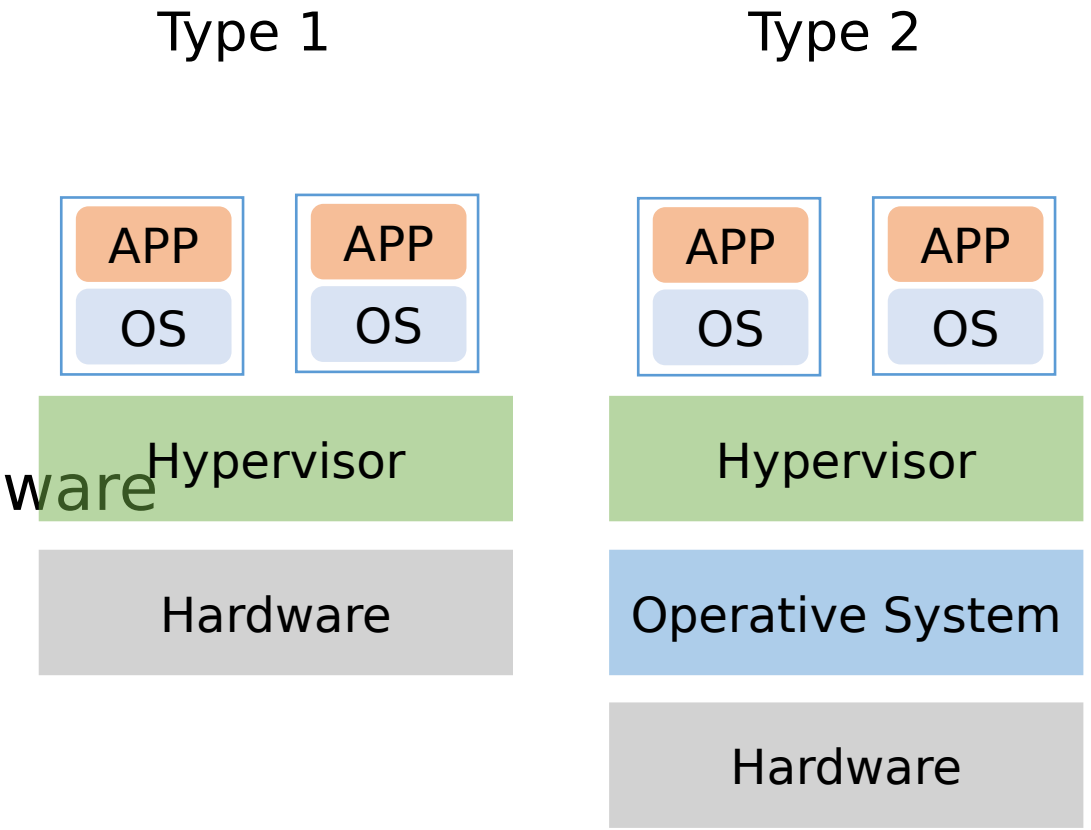
Relative merits of different approaches

Table 3.1 Relative Merits of Virtualization at Various Levels (More "X"'s Means Higher Merit, with a Maximum of 5 X's)

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

Types of Virtualization

- Type 1
  - Bare metal
  - Loaded directly on hardware
  - Example: Xen, VMware ESXi
- Type 2
  - Hosted hypervisor
  - Loaded in an OS running on hardware
  - Example: Virtual Box

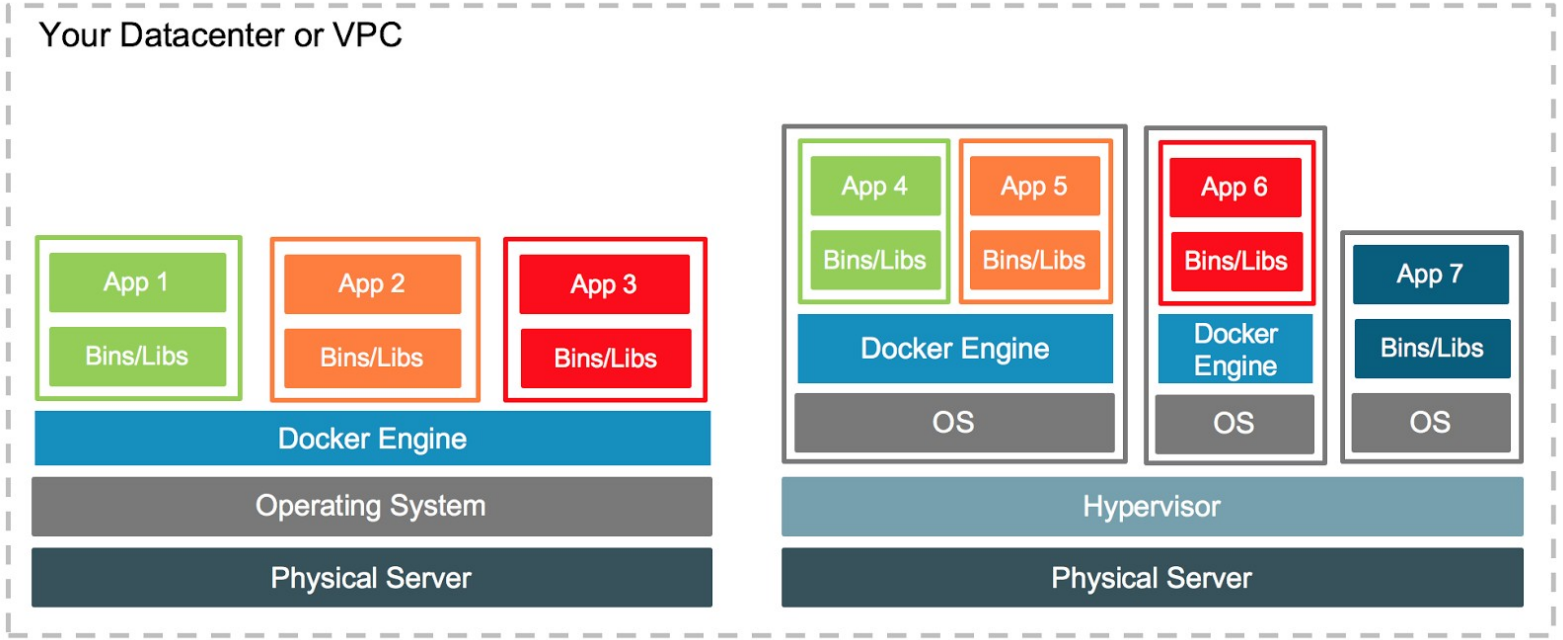


Virtualization at System Level

Full Virtualization (Binary Translation)	• Virtual machine simulates hardware to allow an unmodified guest OS to be run in isolation.
Para-virtualization (Modifed Guest OS)	• The hypervisor is installed on a physical server (host) and a guest OS are aware that it has been virtualized.
Hardware Assisted (Intel VT, AMD-V)	• Processor architecture has special instructions to aid the virtualization of hardware.

OS-Level Virtualization

- Operating system virtualization inserts a virtualization layer inside an operating system to partition a machine’s physical resources. It enables multiple isolated VMs within a single operating system kernel. This kind of VM is often called a virtual execution environment (VE), Virtual Private System (VPS), or simply cont



Examples

- QEMU/KVM or simply KVM
- XEN

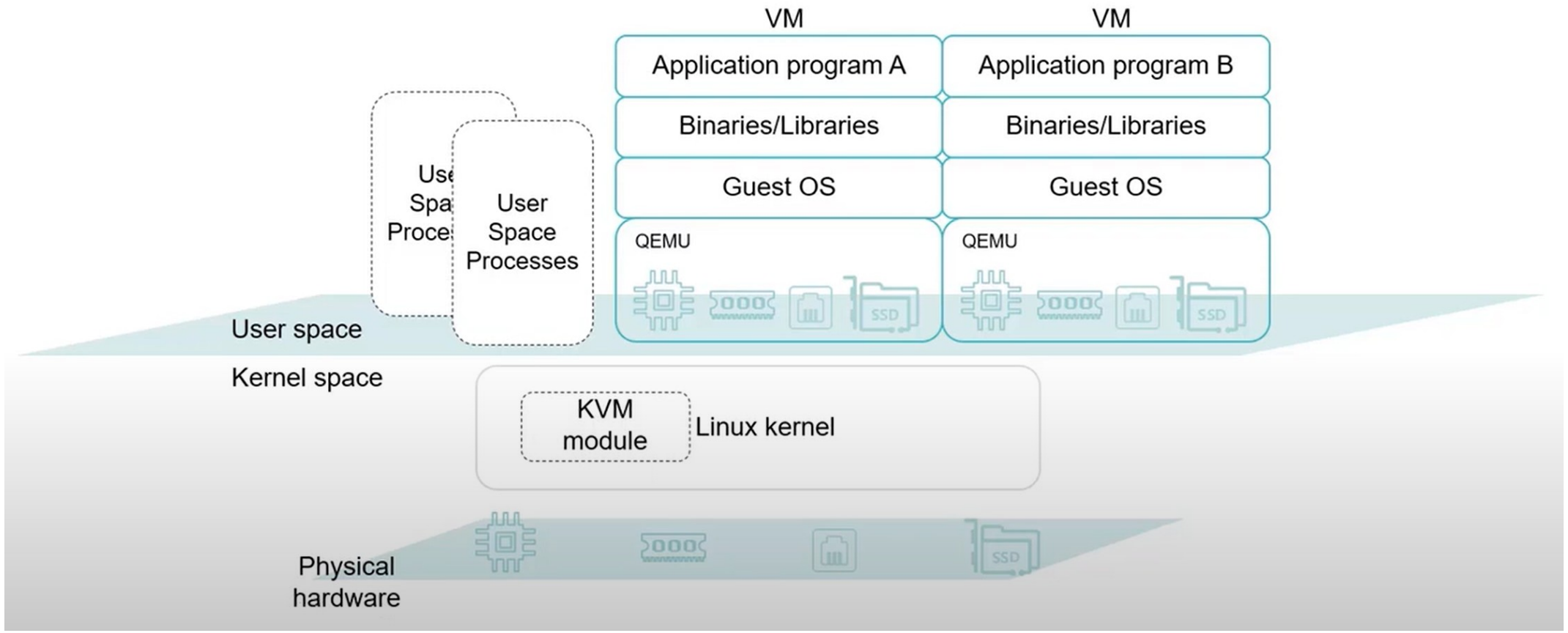
KVM

- KVM is an acronym of “Kernel based Virtual Machine”,
- It is used with QEMU to emulate some peripherals, called QEMU-KVM.
- KVM is a special operating mode of QEMU that uses CPU extensions (HVM) for virtualization via a kernel module.

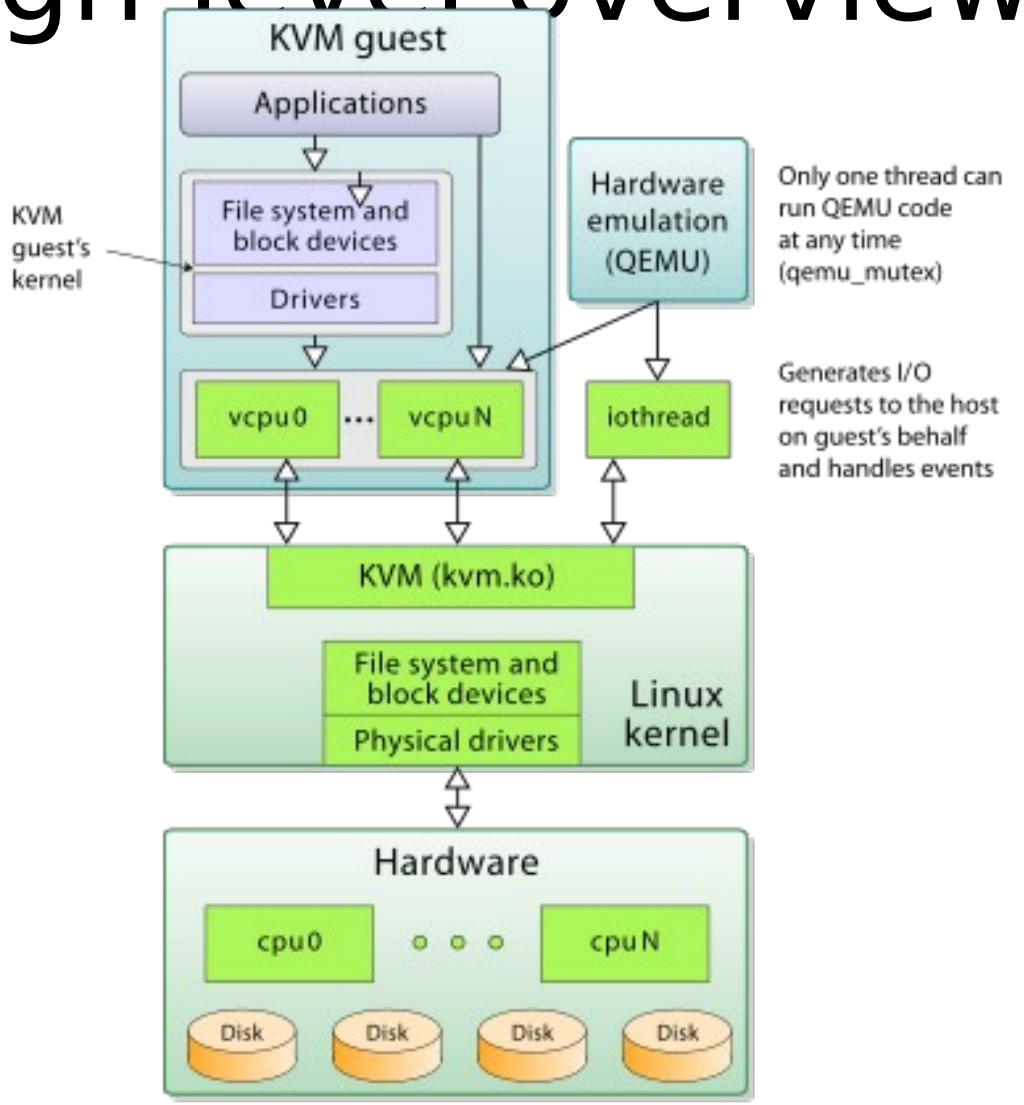
KVM

- KVM itself emulates very little hardware, instead deferring to a higher-level client application such as QEMU, crosvm, or Firecracker for device emulation.
- KVM supports hot plug vCPUs, dynamic memory management, and Live Migration since February 2007, memory write intensive workload impacts in migration process

KVM


VM
VM
Application program A
Application program B
Binaries/Libraries
Binaries/Libraries
Guest OS
Guest OS
QEMU
QEMU
User Space Processes
User space
Kernel space
KVM module
Linux kernel
Physical hardware

KVM: A high-level overview



KVM Execution Model

- Three modes for thread execution instead of the traditional two:
  - User mode
  - Kernel mode
  - Guest mode (executes natively)
- A virtual CPU is implemented using a Linux thread
- The linux scheduler is responsible for scheduling a virtual CPU, as it is a normal thread

KVM Execution Model

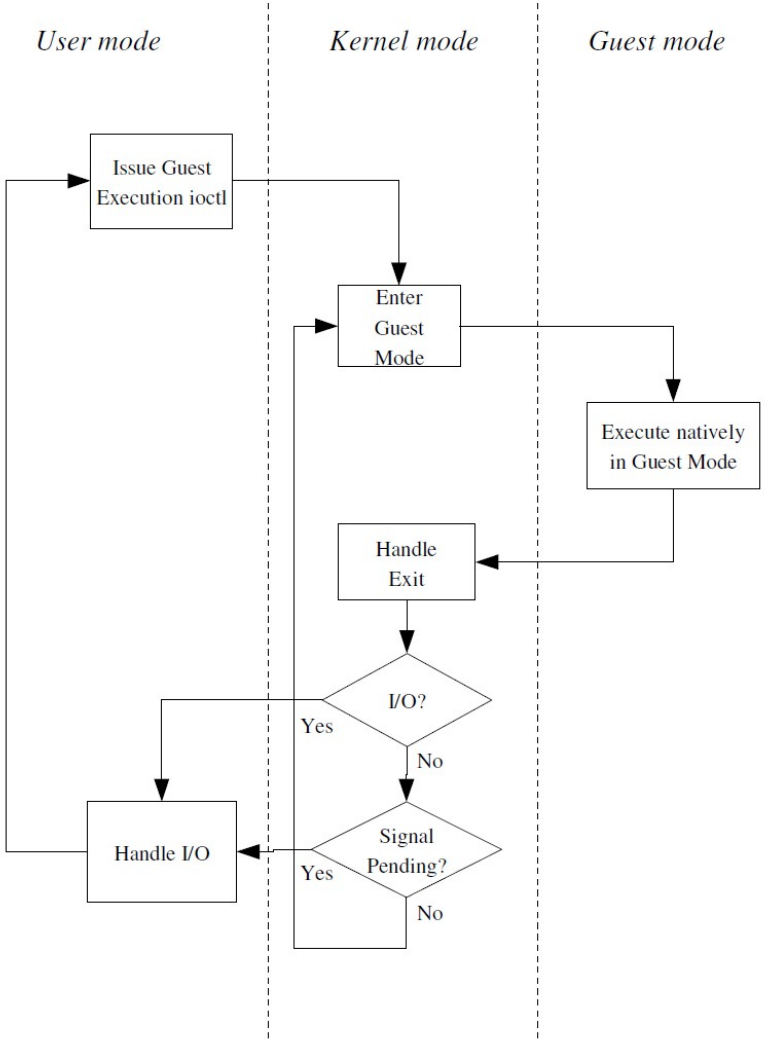


Figure 2: Guest Execution Loop

KVM Memory Model

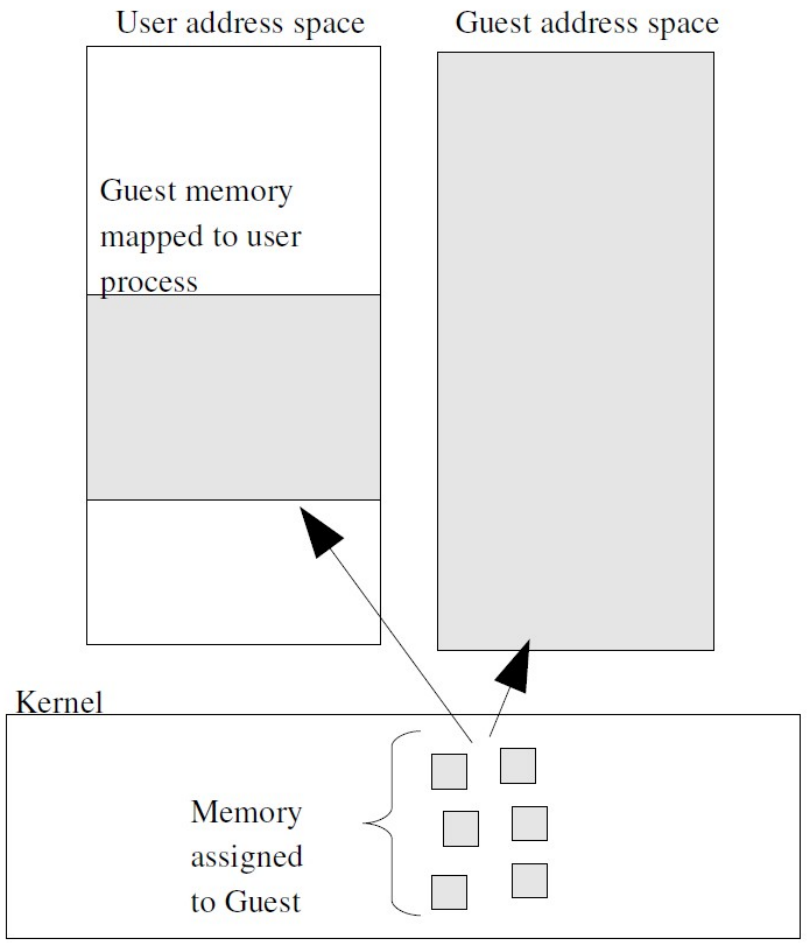


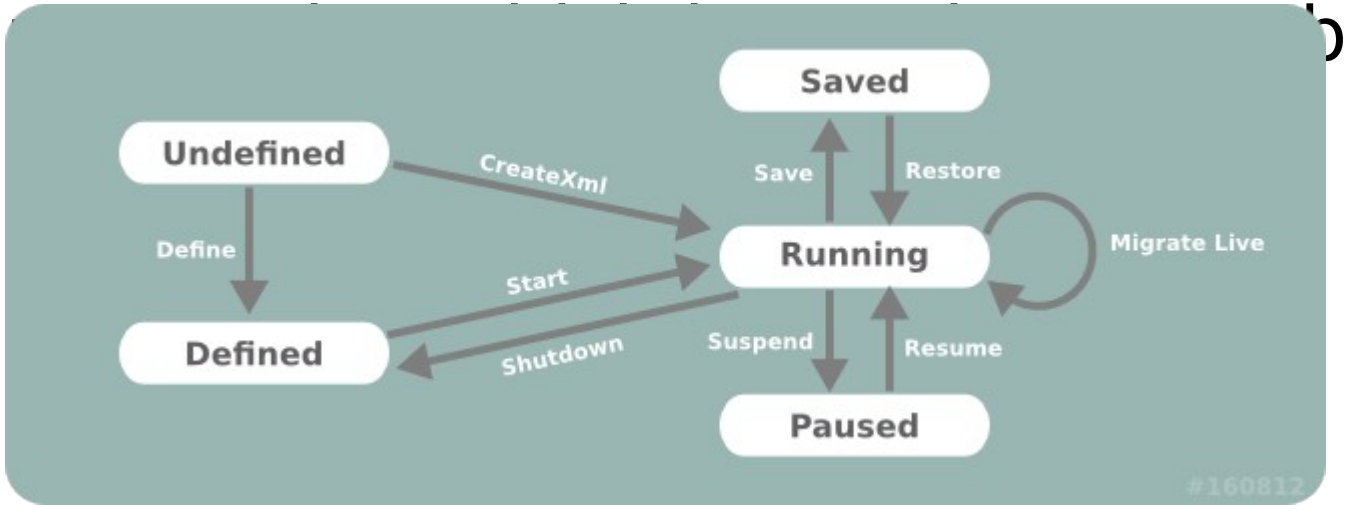
Figure 1: `kvm` Memory Map

Open Question?

- Cual es el flujo para iniciar una VM en KVM?
  - KVM_RUN

Libvirt and KVM

- When you install QEMU/KVM on Linux, **libvirt** is also installed
- A daemon runs on the system and communicates with hypervisors
- Exposes a[illegible]be managed



Xen

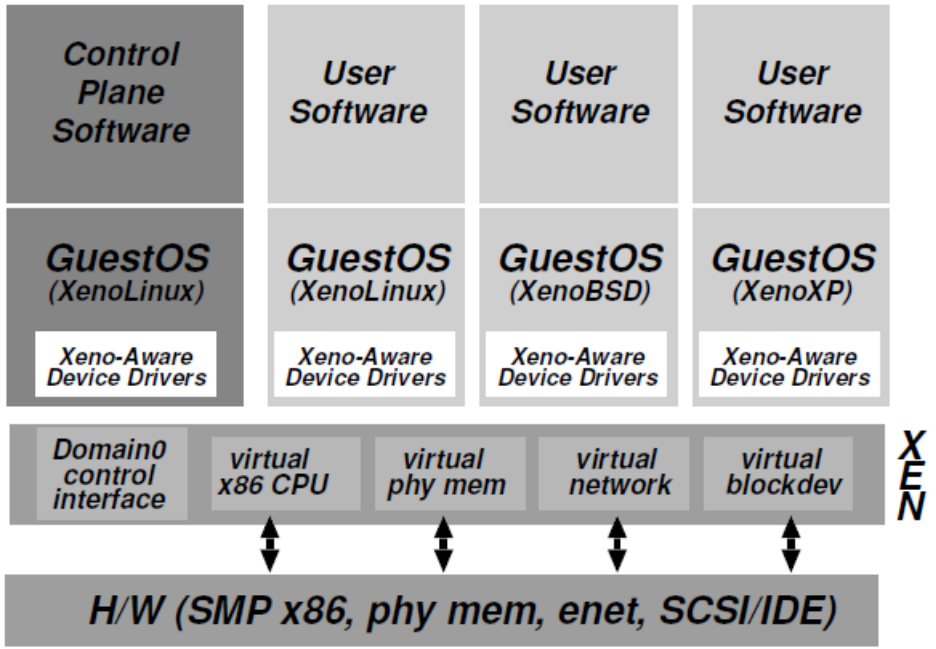
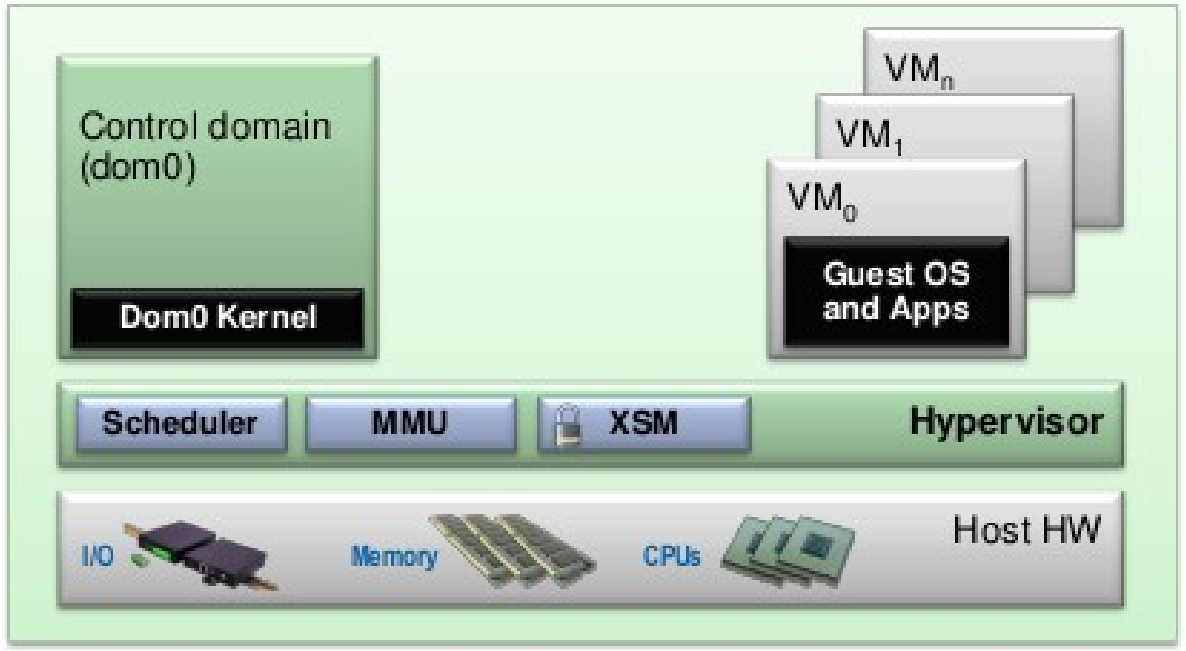


Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including *Domain0* running control software in a XenoLinux environment.

Xen



Trusted Computing Base

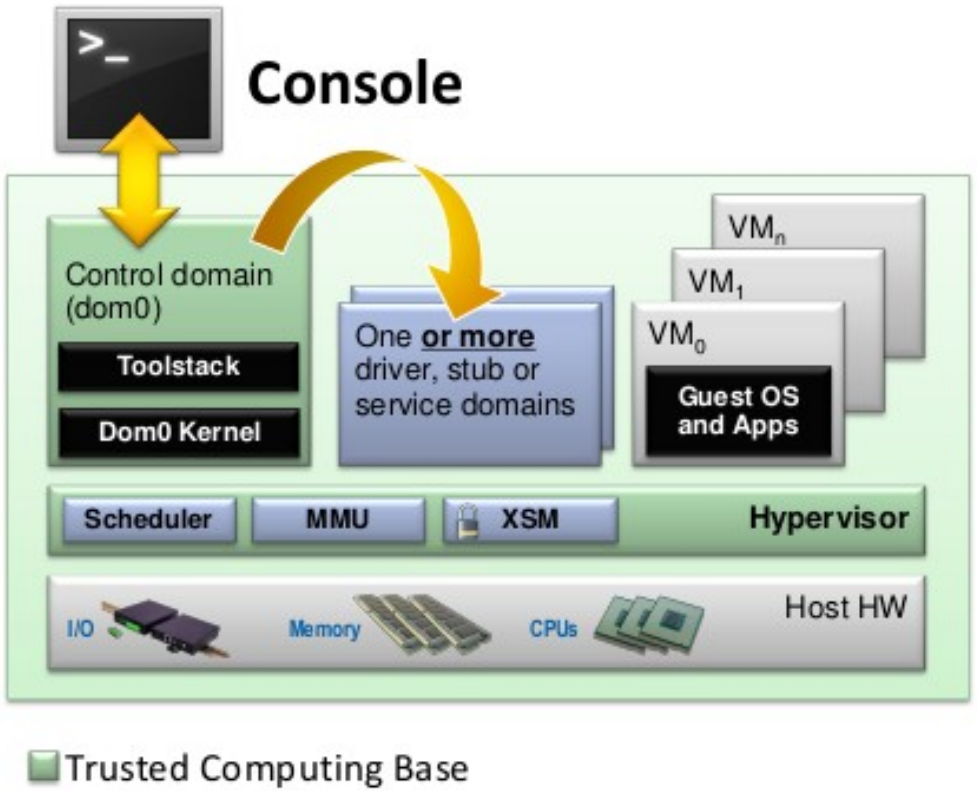
Control Domain aka Dom0

- Dom0 kernel with drivers

Guest Domains

- Your apps

Xen



Console

- Interface to the outside world

Control Domain aka Dom0

- Dom0 kernel with drivers
- Xen Project Management Toolstack

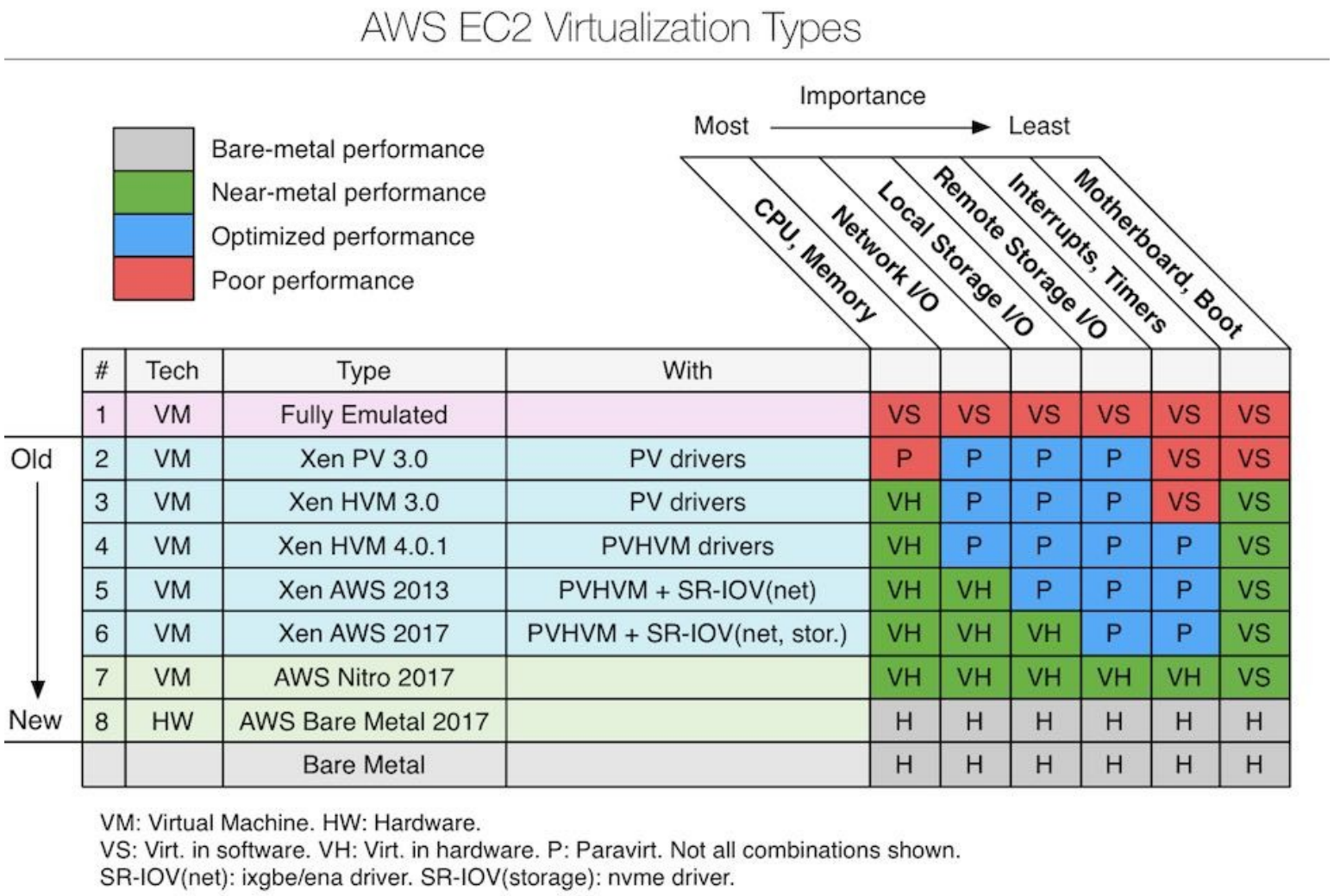
Guest Domains

- Your apps

Driver/Stub/Service Domain(s)

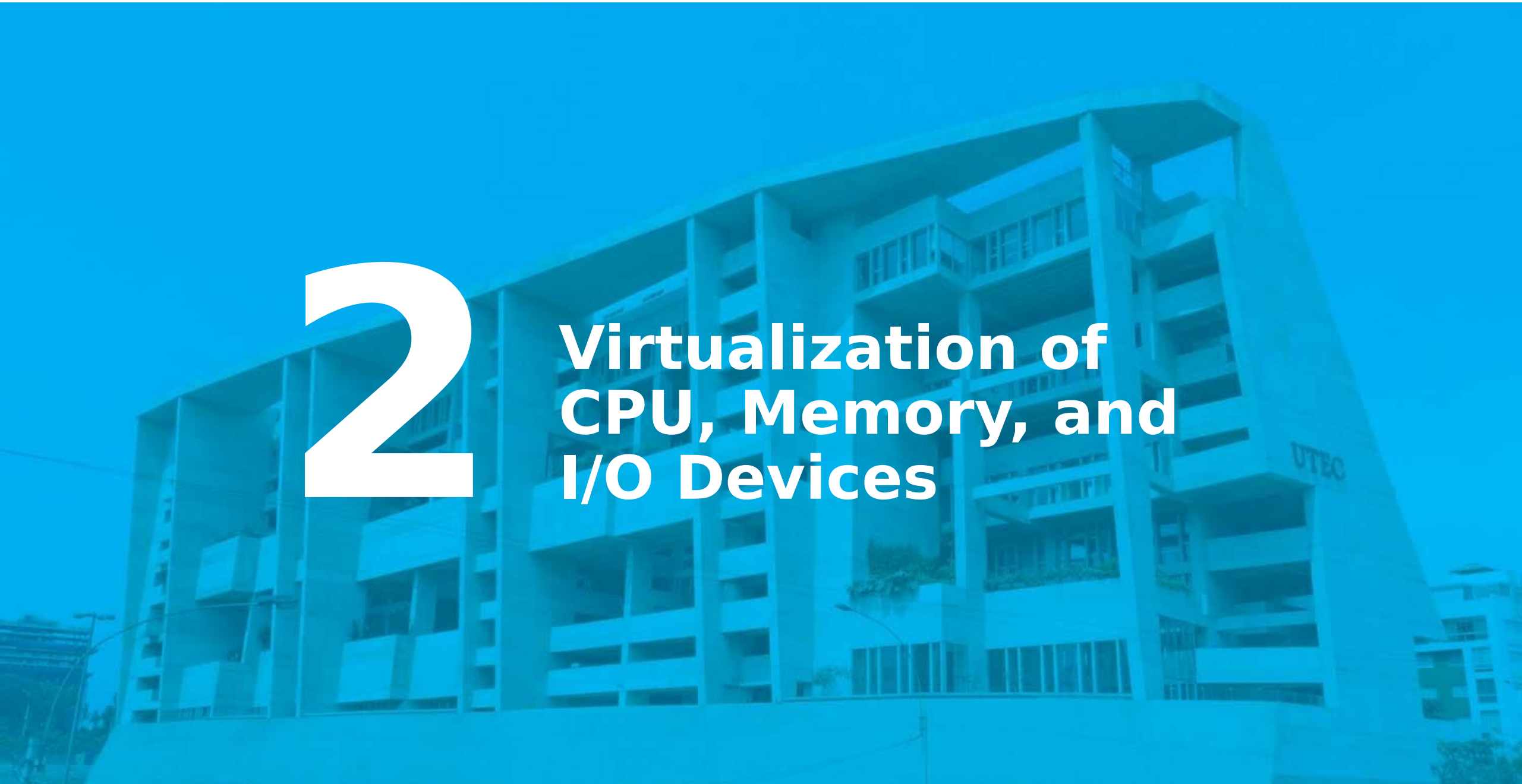
- A "driver, device model or control service in a box"
- De-privileged and isolated
- Lifetime: start, stop, kill

Lectura Complementaria



#	Tech	Type	With						
1	VM	Fully Emulated		VS	VS	VS	VS	VS	VS
2	VM	Xen PV 3.0	PV drivers	P	P	P	P	VS	VS
3	VM	Xen HVM 3.0	PV drivers	VH	P	P	P	VS	VS
4	VM	Xen HVM 4.0.1	PVHVM drivers	VH	P	P	P	P	VS
5	VM	Xen AWS 2013	PVHVM + SR-IOV(net)	VH	VH	P	P	P	VS
6	VM	Xen AWS 2017	PVHVM + SR-IOV(net, stor.)	VH	VH	VH	P	P	VS
7	VM	AWS Nitro 2017		VH	VH	VH	VH	VH	VS
8	HW	AWS Bare Metal 2017		H	H	H	H	H	H
		Bare Metal		H	H	H	H	H	H

http://www.brendangregg.com/blog/2017-11-29/aws-ec2-virtualization-2017.html



2 Virtualization of CPU, Memory, and I/O Devices

Hardware Support for Virtualization

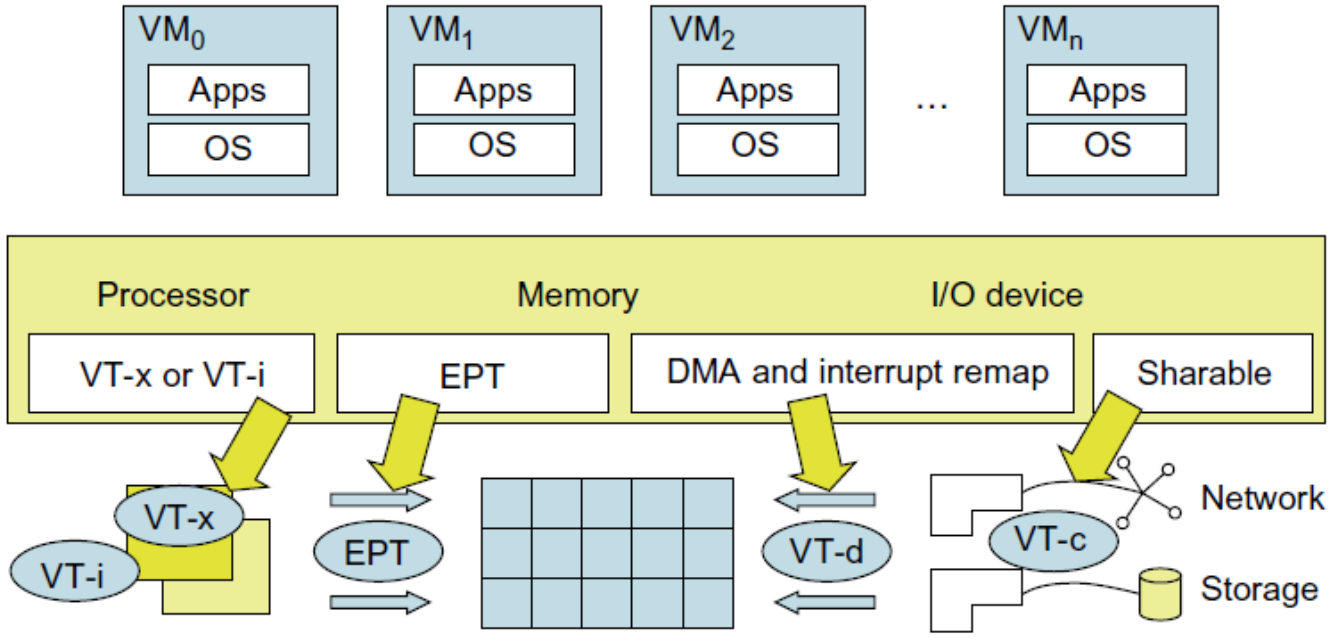


FIGURE 3.10

Intel hardware support for virtualization of processor, memory, and I/O devices.

(Modified from [68], Courtesy of Lizhong Chen, USC)

CPU Virtualization

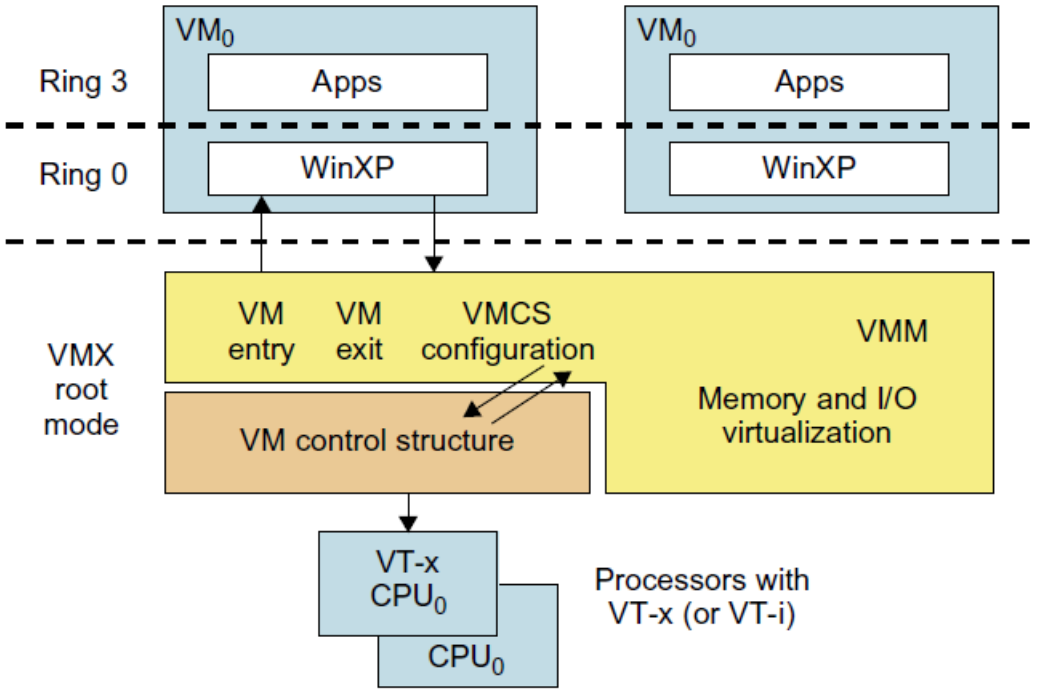
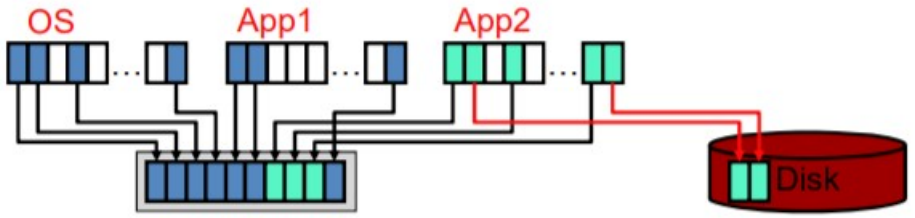


FIGURE 3.11

Intel hardware-assisted CPU virtualization.

(*Modified from [68], Courtesy of Lizhong Chen, USC*)

Virtual Memory



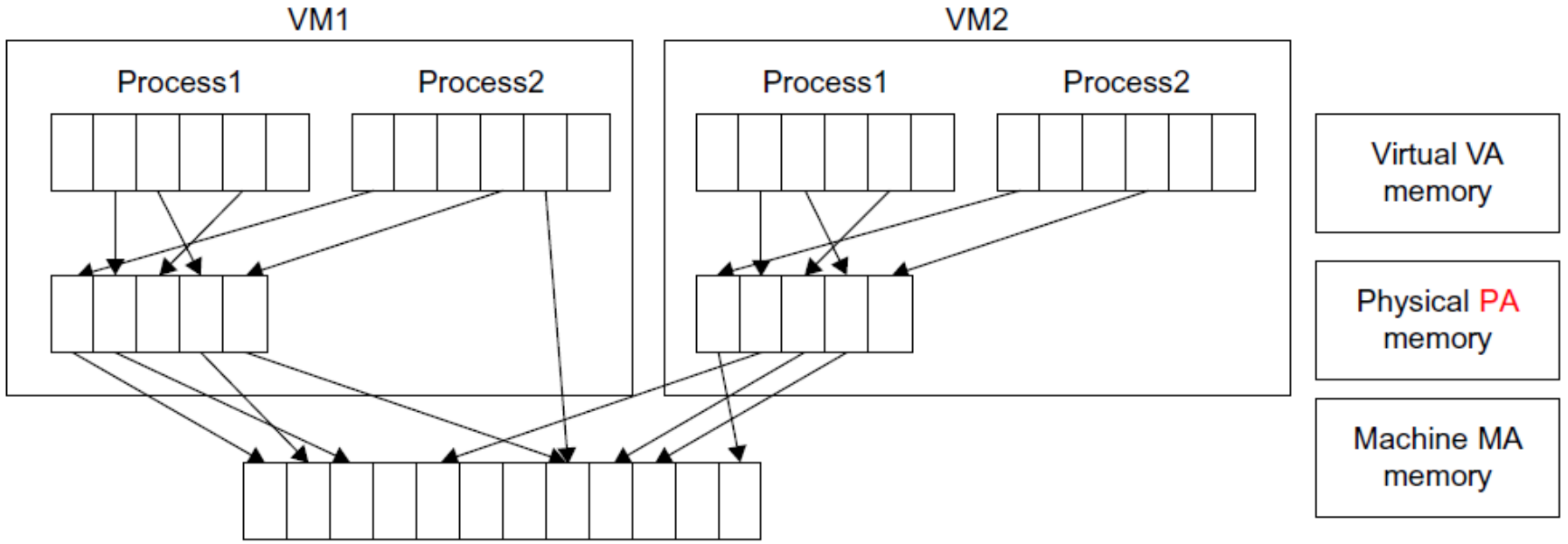


FIGURE 3.12

Two-level memory mapping procedure.

(Courtesy of R. Rblig, et al. [68])

Storage

Types of I/O virtualization:
- full device emulation
- para-virtualization
- direct I/O

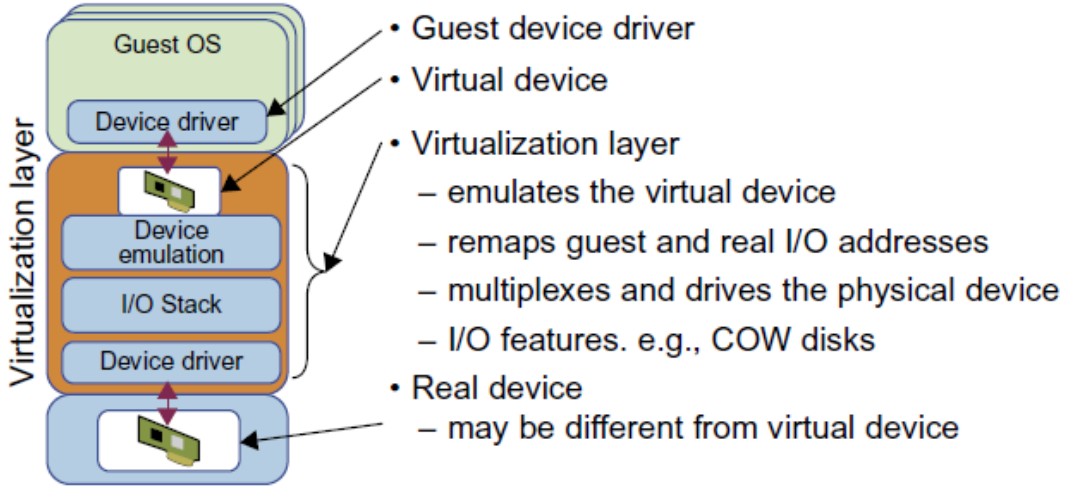


FIGURE 3.14

Device emulation for I/O virtualization implemented inside the middle layer that maps real I/O devices into the virtual devices for the guest device driver to use.

(*Courtesy of V. Chadha, et al. [10] and Y. Dong, et al. [15]*)

3 Virtual clusters and resource management

Virtual Cluster

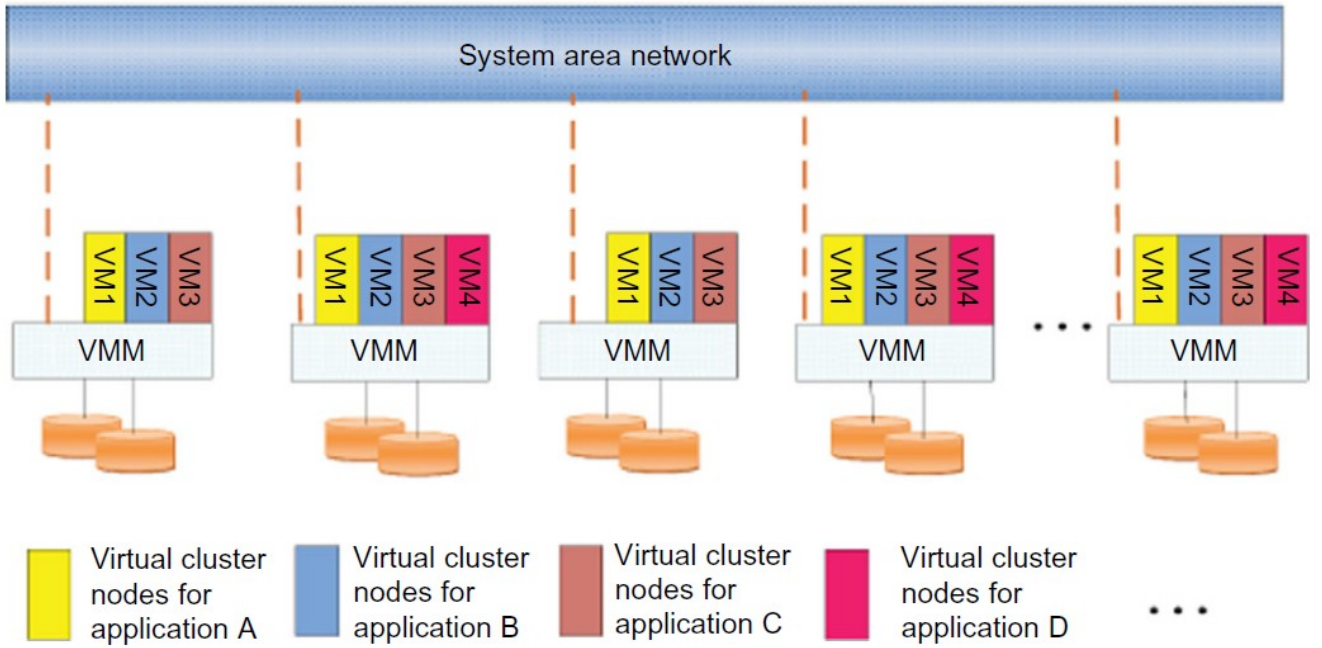


FIGURE 3.19

The concept of a virtual cluster based on application partitioning.

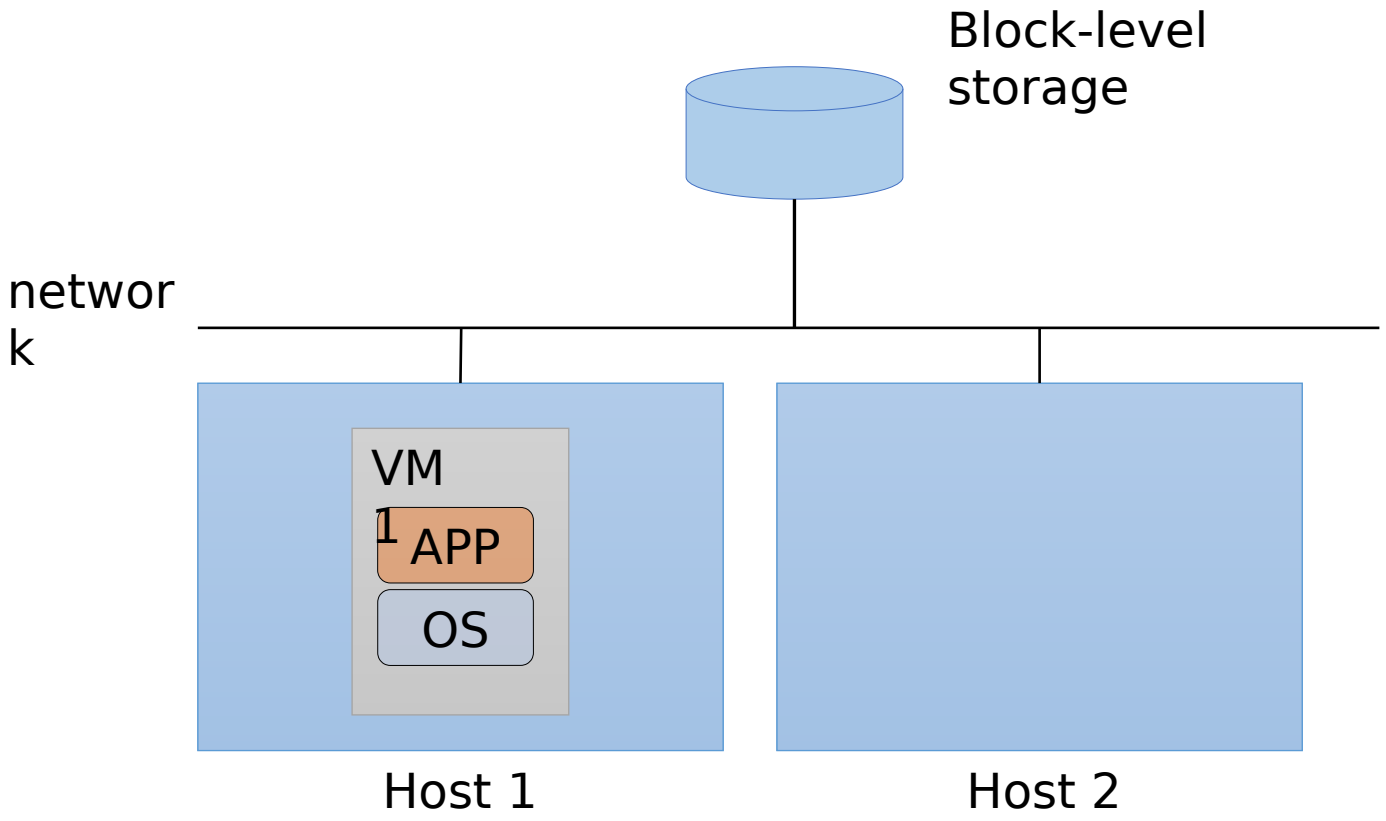
(*Courtesy of Kang Chen, Tsinghua University 2008*)

Virtual Clusters Operations

- Migration of VMs
  - Live migration
- Storage migrations
- Consolidacion de servidores

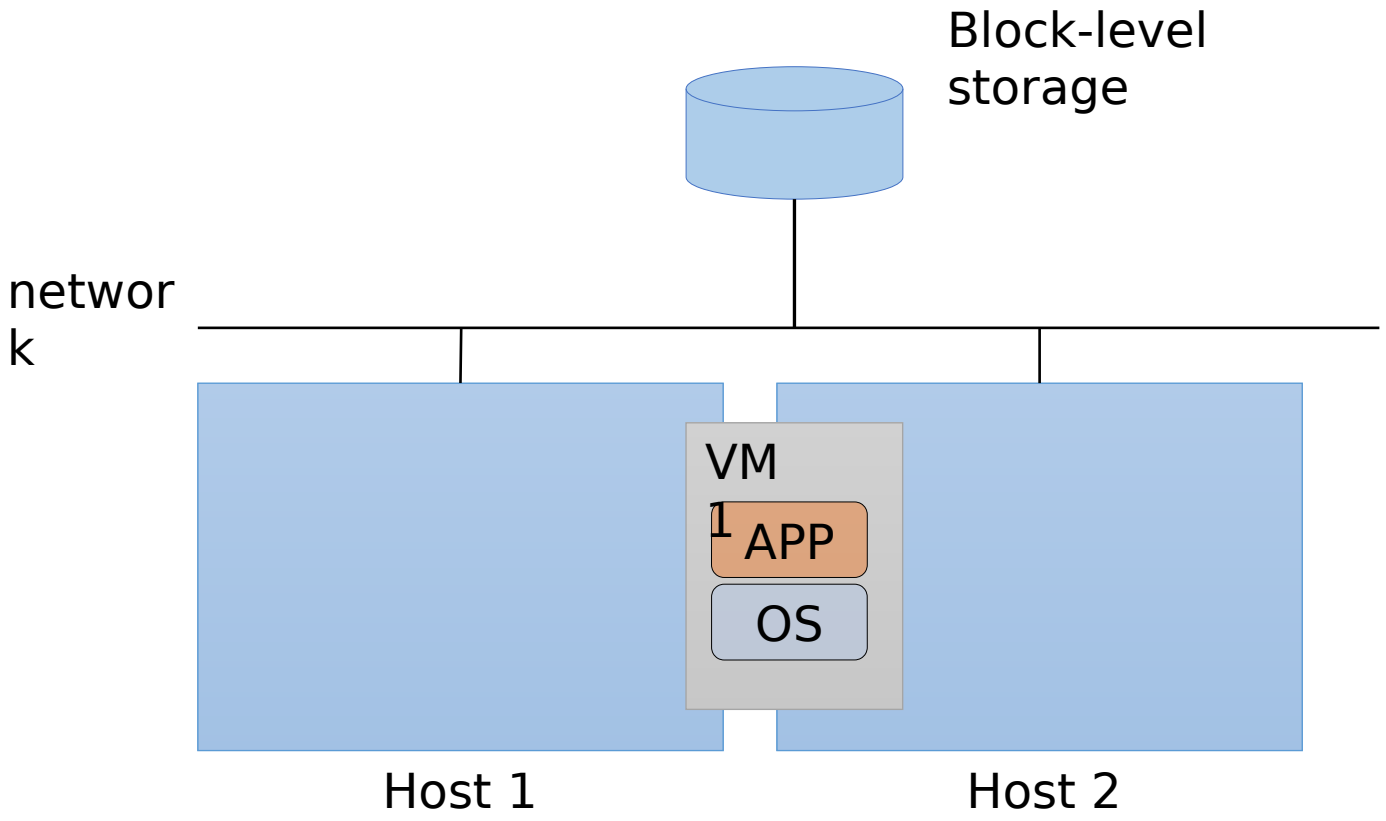
Live migration of VMs

- Concept



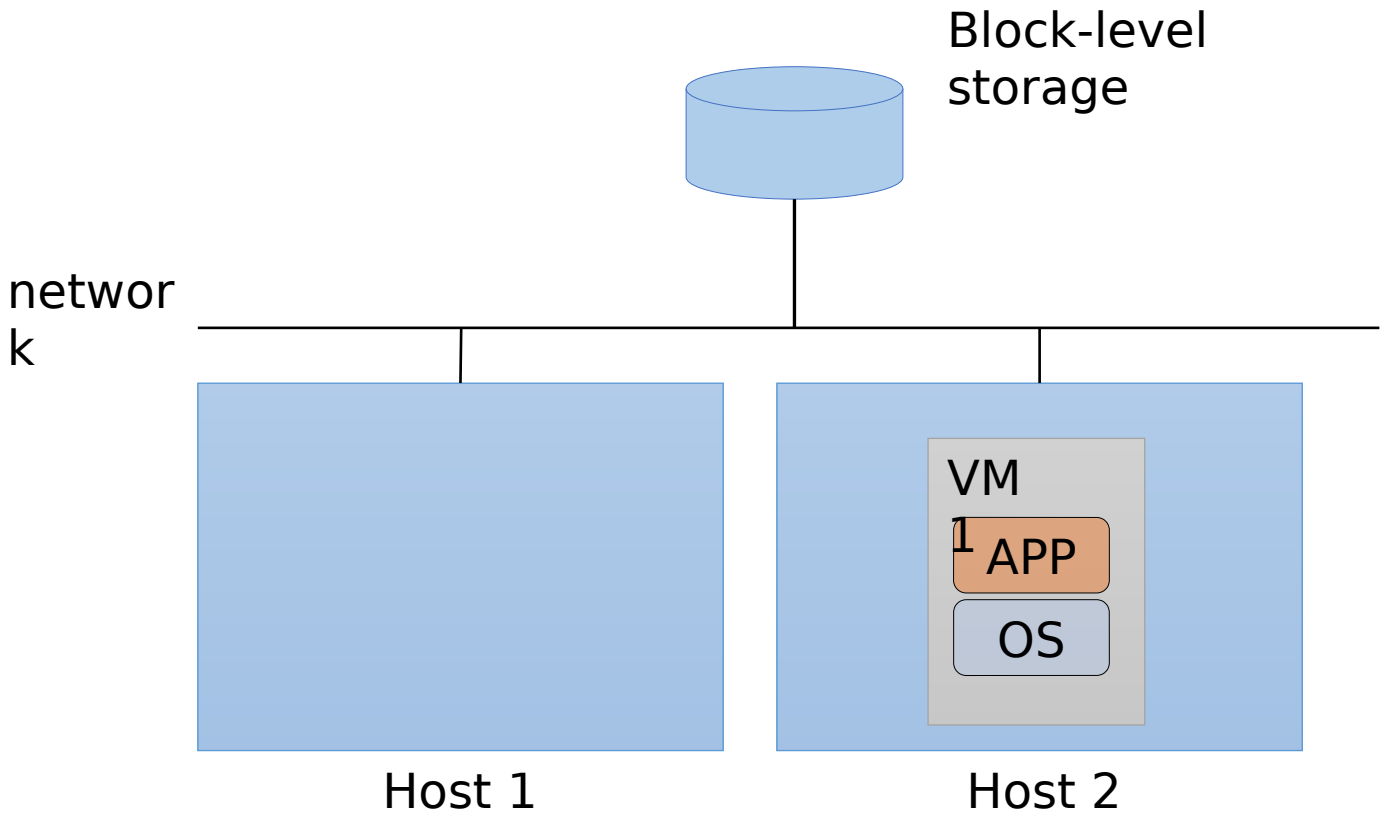
Live migration of VMs

- Concept



Live migration of VMs

- Concept



Live migration of VMs

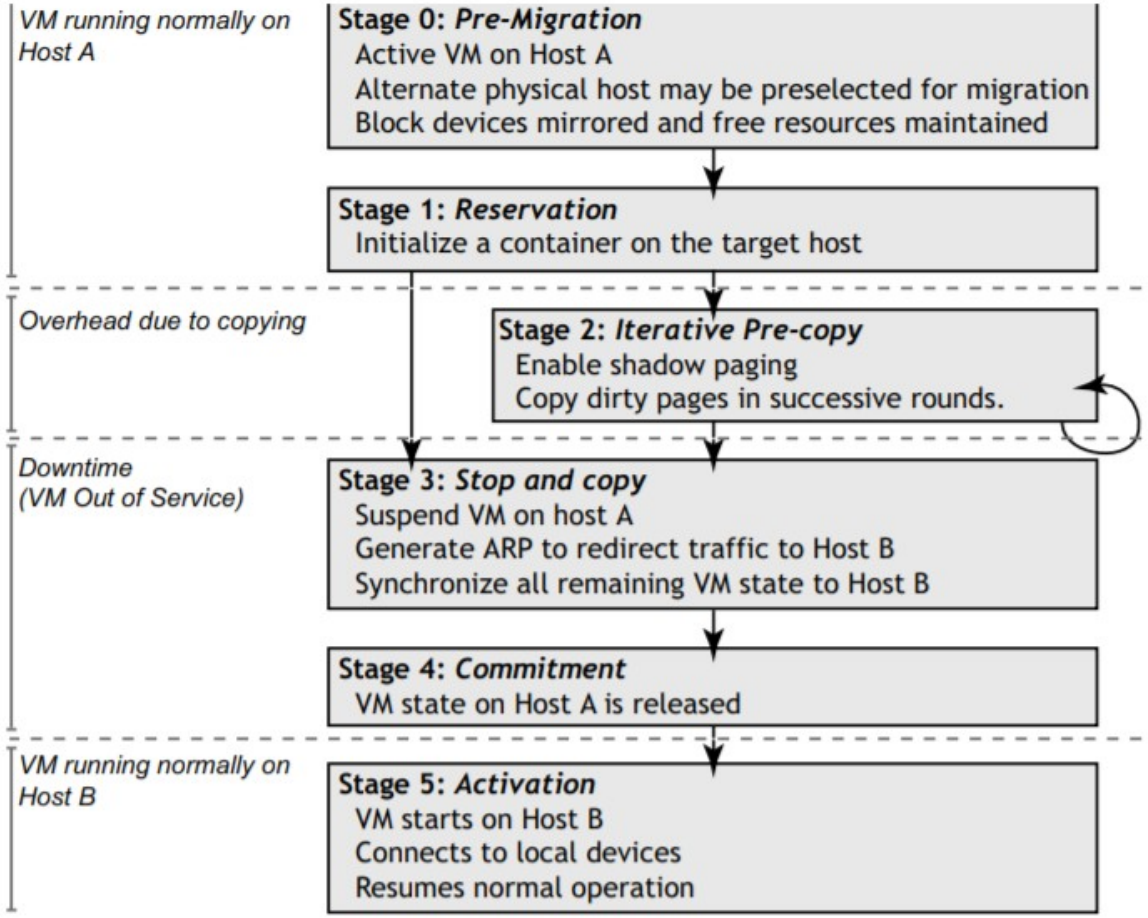
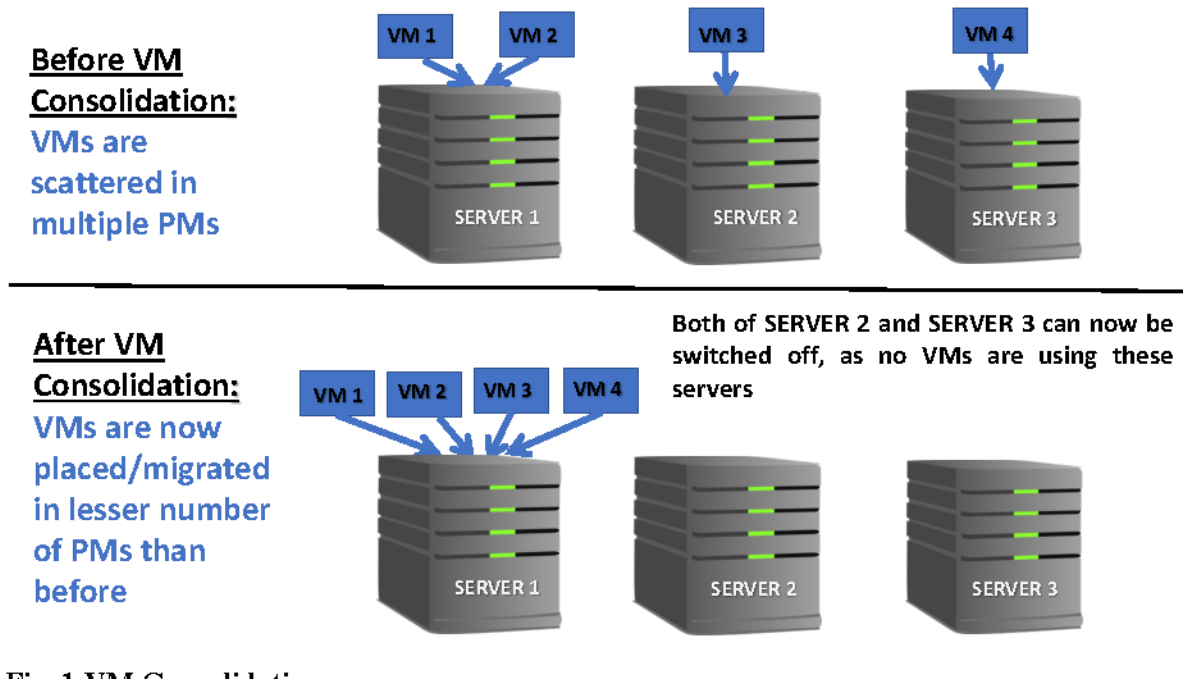


Figure 1: Migration timeline

Consolidación de servidores



PRODUCT DATASHEET

VMware VMotion

Live migration of virtual machines without service interruption

AT A GLANCE

VMware® VMotion™ enables the live migration of running virtual machines from one physical server to another with zero downtime, continuous service availability, and complete transaction integrity. VMotion is a key enabling technology for creating the dynamic, automated, and self-optimizing data center.

BENEFITS

- Continuously and automatically allocate virtual machines within resource pools.
- Improve availability by conducting maintenance without disrupting business operations

How Is VMware VMotion Used?

VMotion allows users to:

- Automatically optimize and allocate entire pools of resources for maximum hardware utilization, flexibility and availability.
- Perform hardware maintenance without scheduled downtime.
- Proactively migrate virtual machines away from failing or under-performing servers.

How Does VMotion work?

Live migration of a virtual machine from one physical server to another with VMotion is enabled by three underlying technologies.

First, the entire state of a virtual machine is encapsulated by a set of files stored on shared storage such as Fibre Channel or iSCSI Storage Area Network (SAN) or Network Attached Storage (NAS). VMware's clustered Virtual Machine File System (VMFS) allows multiple installations of ESX Server to access the same virtual machine files concurrently

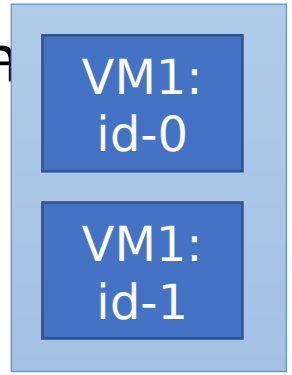
4 Tarea

Tarea

1. Crear um script (bash, libvirt, etc.) que ejecute la operacion de Live Migration.
2. Crear uma **politica** simple que me permita controlar operaciones de live migration
3. (bonus) Como parte de un plan de consolidacion de servidores se ha pedido monitorar el impacto de la operacion live migration para los usuarios de una aplicacion web (o otra similar). Proponga una escenario de comparacion.
    1. obligatorio para grupos de 3

Tarea

1. Crear um script (bash, libvirt, ~~powershell~~, etc.) que ejecute la operacion de Live Migration.
    1. VirtualBox: Teleport
    2. KVM: live migration
    3. ~~(VMWare ESXi: vMotion~~
    4. Xen: live migration:



Tarea

2. Crear uma **politica** simple que me permita controlar operaciones de live migration

2.1 Ejemplo de politica

CPU > 80% -✉ trigger Live Migration,, : id-1

CPU 85% -✉ Live Migration : id-2

2.2 CPU, memoria

> Ejemplo de aplicacion para generar carga

$ apt get install stress-ng

$ sleep 30; stress-ng --cpu 2 –memory : 250MB – i/o

Tarea

(bonus) Como parte de un plan de consolidacion de servidores se ha pedido monitorar el impacto de la operacion live migration para los usuarios de una aplicacion web (o otra similar). Proponga una escenario de comparacion.

Ejemplo

1. Aplicacion web o otra similar:
2. Monitoreo: CPU performance in python, Memoria, (I/O)
3. Politica: (CPU > 60% y Memoria 75%) VM. -> Live migration
4. Resultados (figura): tiempo x CPU, etc.