TEII

Tecnologías Específicas de la Ingeniería Informática

Virtualización

Jorge Bernal Bernabé

Departamento de Ingeniería de la Información y las Comunicaciones (DIIC)

jorgebernal@um.es

https://webs.um.es/jorgebernal

Grado en Ingeniería Informática - 2021/22

16 Febrero 2022

Facultad de Informática

UNIVERSIDAD DE MURCIA

Virtualización

Introducción

Virtualización: *"Una técnica (normalmente con un software asociado) que permite encapsular una unidad de proceso (ya sea un programa, un sistema operativo o incluso un equipo completo, dependiendo de a que profundidad se sitúe el nivel de virtualización) para su ejecución dentro de un entorno en un equipo anfitrión que emula el entorno real transparentemente"*

La Virtualización sustituye la versión hardware de un recurso por una versión software con idéntica funcionalidad aparente.

Introducción

Virtualización de plataforma

- Esta involucra la simulación de máquinas virtuales. La virtualización de plataforma se lleva a cabo en una plataforma de hardware mediante un software "**host**" (en castellano "anfitrión", que es un programa de control) que simula un entorno computacional (máquina virtual) para su software "**guest**" (en castellano "invitado").
- Este software "guest", que generalmente es un sistema operativo completo, corre como si estuviera instalado en una plataforma de hardware autónoma.
- Muchas máquinas virtuales son simuladas en una máquina física dada.
- Para que el sistema operativo "guest" funcione, la simulación debe ser lo suficientemente grande como para soportar todas las interfaces externas del sistema a virtualizar, las cuales pueden incluir a los drivers de hardware.

Ventajas

- Ahorro **costes** de infraestructura
  - Menor número de dispositivos hardware
  - Menores costes de adquisición y mantenimiento
  - Menor consumo de energía
- La infraestructura se vuelve **flexible**, superando límites geográficos y siendo más reconfigurable.
- Mayor aprovechamiento recursos (p. ej. servidores)
- **Simplifica** la gestión (múltiples equipos: configuración, detección de errores,...)
- Mayor confiabilidad (Sustitución ante caídas)
- Reducen la complejidad de las copias de seguridad
- Capacidad de rollback de un sistema completo.

Ventajas (II)

- Mejora la eficiencia del uso de recursos porque permite:
  - El uso compartido de un recurso no congestionado en el mismo instante
  - Complementar picos y valles en el ciclo anual de servicios
  - El uso complementario de cargas que afectan a distintos aspectos del hardware (CPU, IO, memoria, etc..)
- Aislamiento de cargas de trabajo distintas.
- Permite hacer despliegues automatizados tanto para crecer como para decrecer.
- Extender el datacenter privado a uno público
- Convertir una inversión (compra) en un suministro (pago por uso)

Problemas Virtualización

- Se concentra toda la exigencia (IO, RAM, CPU) en menor cantidad de recursos físicos.
- La cantidad de elementos a gestionar se multiplica .
- Los cuellos de botella que se presentan en la nube privada suelen tener soluciones costosas. Habitualmente se tiene problemas de E/S.
- Es una nueva capa que hay que conocer y gestionar además de las existentes en un datacenter no virtualizado.
- Se necesita hardware de mayor potencia
- Depende del tipo de virtualización
- Menor rendimiento de las aplicaciones
- Servidor puede ser un único punto de fallo
- Tendencia a sobre-provisionar

Conceptos Virtualización

- **Host:** anfitrión en un entorno virtualizado. Como ejemplo concreto podemos pensar en un equipo físico con un sistema operativo donde se crean máquinas virtuales.
- **Guest:** el recurso virtual acogido en el host. Típicamente la máquina virtual.
- **Hypervisor o VMM**(Virtual Machine Monitor/Manager): componente software donde reside (casi toda) la función de emulación y la gestión de las máquinas virtuales. El VMM maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, Memoria, Red, Almacenamiento) y así podrá repartir dinámicamente dichos recursos entre todas las maquinas virtuales definidas en el computador central. De modo que nos permite tener varios ordenadores virtuales ejecutándose sobre el mismo ordenador físico.

Tipos de Hipervisor

- Type-1, native or bare-metal hypervisors
  - VMware ESXi
  - Hyper-V
  - Xen
- Type-2 or hosted hypervisors
  - Virtualbox
  - VMware Workstation / Player /
  - VMware Fusion (soporte Mac adicional)

*Formal Requirements for Virtualizable Third Generation Architectures, Gerald J. Popek and Robert P. Goldberg *Listas no exhaustivas.*

Escenarios de Aplicación

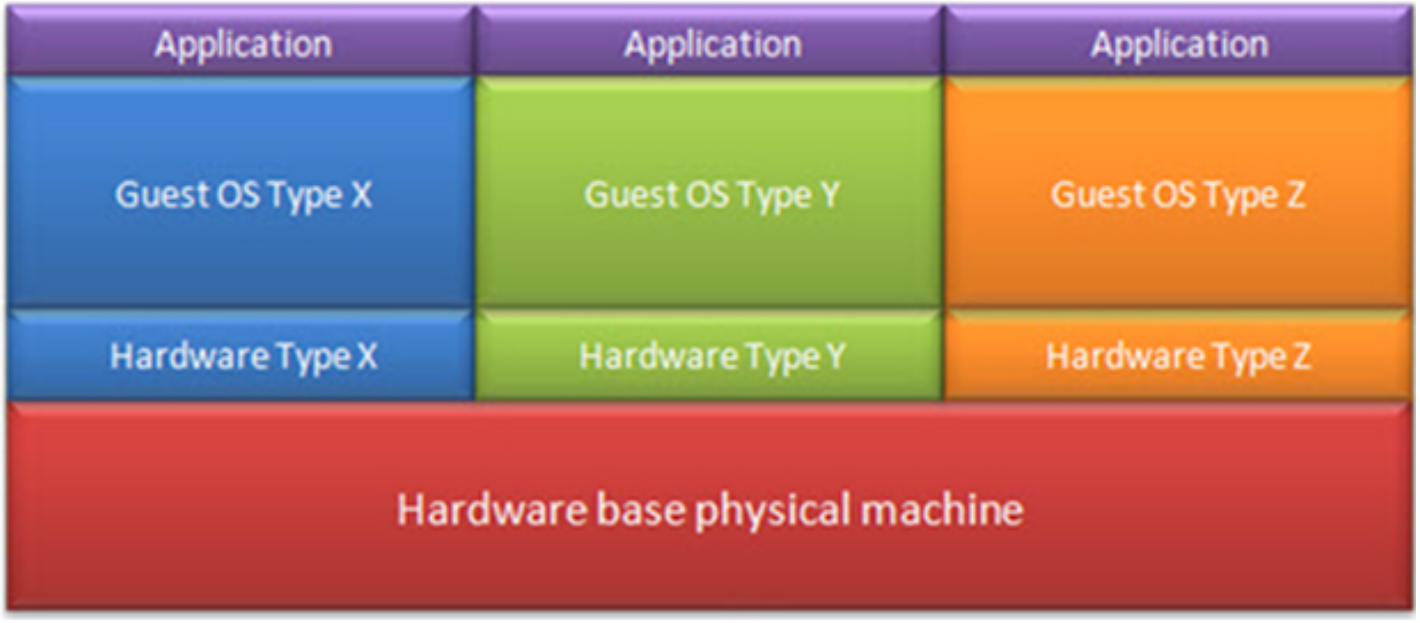
- Optimización de la Infraestructura Tecnológica
  - Uso de recursos infrautilizados
  - Reducción del uso de HW
  - Reducción de dependencias, costes de gestión, etc
- Continuidad/Recuperación ante desastres
  - Pool de máquinas virtuales y recursos de almacenamiento
  - Pueden ser colocados, reubicados y replicados
- Pruebas y desarrollo
  - Pruebas software, sistemas operativos
  - Reduce el número de servidores para test
  - Distintas configuraciones
- Infraestructura de escritorio virtual
  - Simplifica las tareas de gestión de los administradores
  - Mayor control de la información y seguridad.
  - Proyecto EVA (UMU) https://eva.um.es
- Soporte a sistemas operativos y aplicaciones antiguas

Tipos de virtualización

- **Full virtualization** – se considera que la simulación del hardware es completa, tanto que permite que el guest ejecute el software sin modificación alguna, principalmente el sistema operativo.
  - Binary rewriting: cuando no hay soporte hardware en el procesador físico.
  - Hardware-assisted Virtual Machine (HVM): con soporte hardware en el procesador.
- **Paravirtualization** – el hardware no es simulado, sino que es el anfitrión tiene los medios para generar entornos aislados en los que se ejecutan los sistemas guest. El sistema operativo del guest tiene el kernel modificado.
- **Operating-system-level virtualization** - el kernel del sistema operativo host tiene mecanismos para crear contenedores del mismo tipo de sistema operativo.

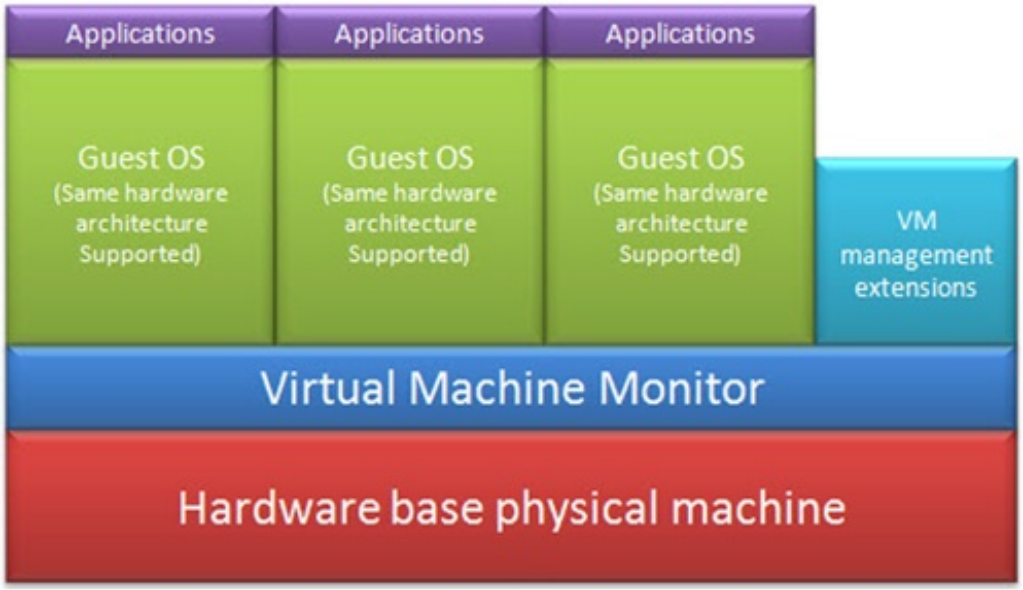
Emulación

- SW replica la funcionalidad al completo del hardware de la máquina
- Bajo rendimiento
- Ej. QEMU



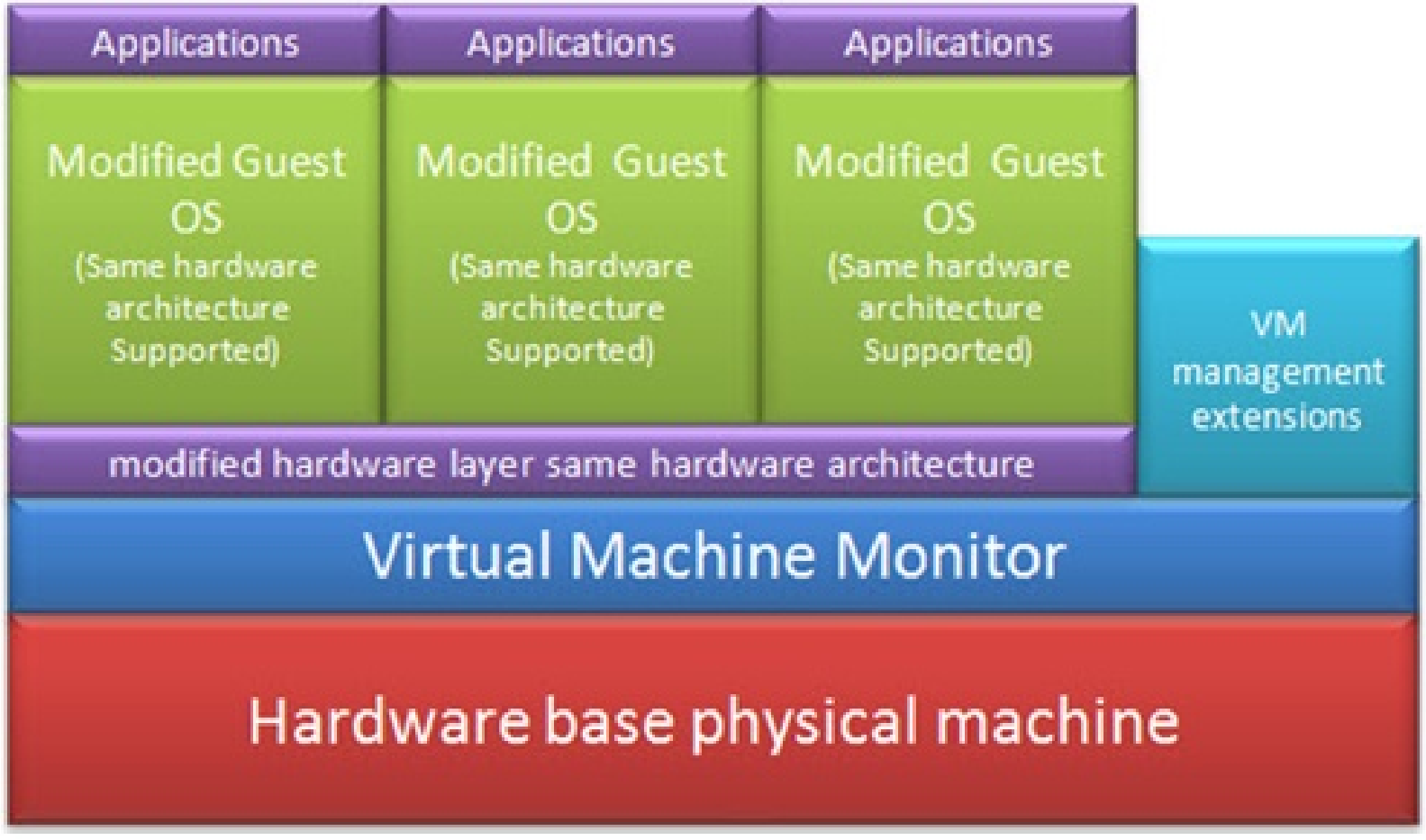
Virtualización Completa

- Proporciona una capa de adaptación a las máquinas invitadas (VMs)
- SO anfitrión, SO invitado y monitor de máquinas virtuales (VMM) o hipervisor
- Permite ejecutuar distintas versiones de SSOO (Windows Server 2003, Windows Server 2008, Linux, etc)
- **El guest no es modificado**
- Permite que muchas instancias corriendo al mismo tiempo



Paravirtualización

- **Modificación del SO invitado** para cooperar con el VMM y el host. Llamadas al VMM o a los dispositivos reales (instrucciones hardware especiales)

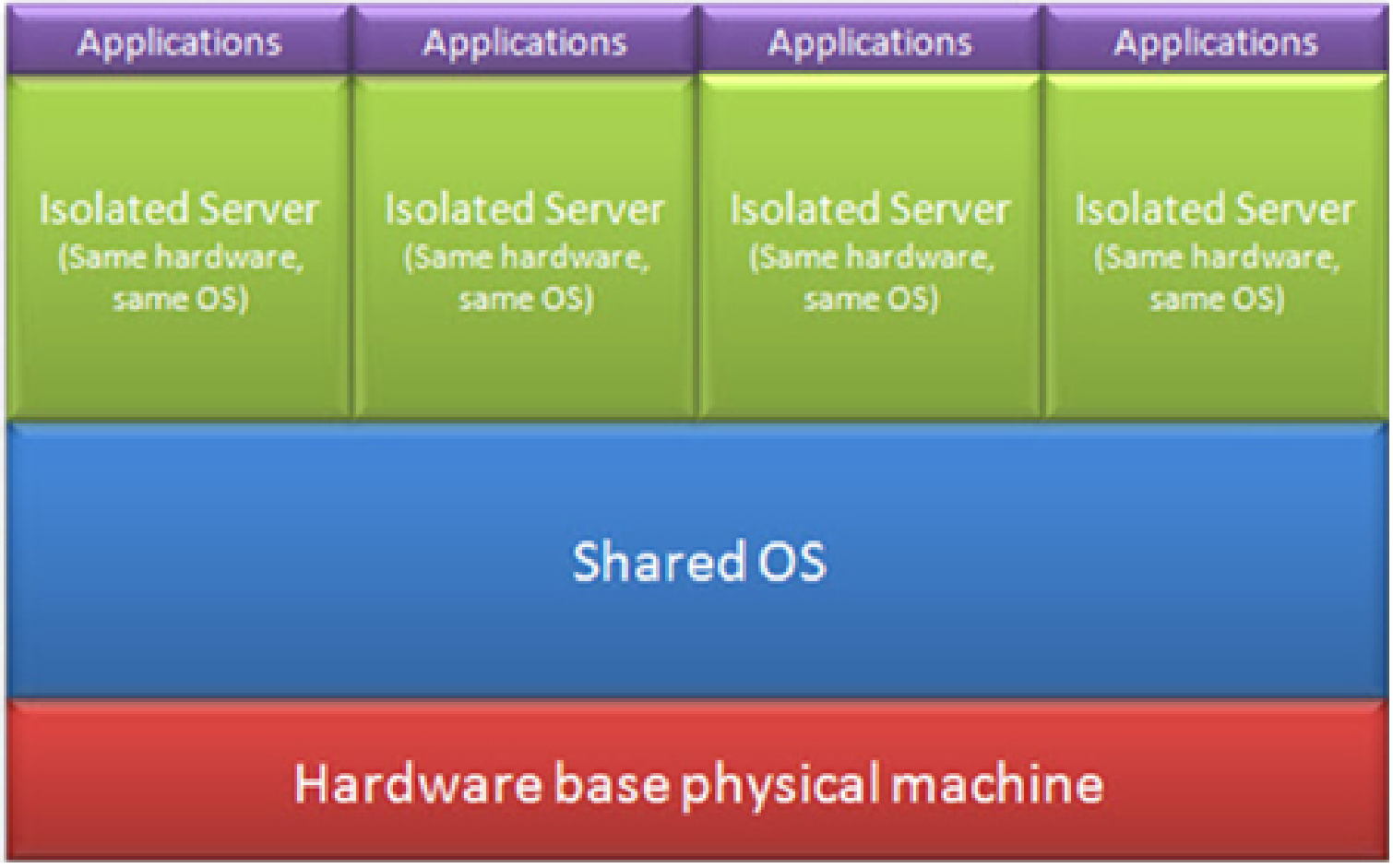


Paravirtualización

- La Paravirtualización es complementaria a la virtualización completa
- Se usa para hacer que un guest sepa que está siendo virtualizado.
- En esos casos hay configuraciones adicionales y un menor overhead que pueden mejorar el rendimiento.
- Se emplean driver de dispositivo para-virtualizados y el objetivo es reducir la sobrecarga motivada por la emulación.
- Rápida y ligera
- Imágenes más pequeñas
- Ejemplos: Xen, VMware, Hyper-V, KVM y Virtualbox.

Virtualización de Sistema Operativo

- Particiona la capa del SO existente
- Tanto anfitrión como invitados comparten el mismo SO/Kernel.

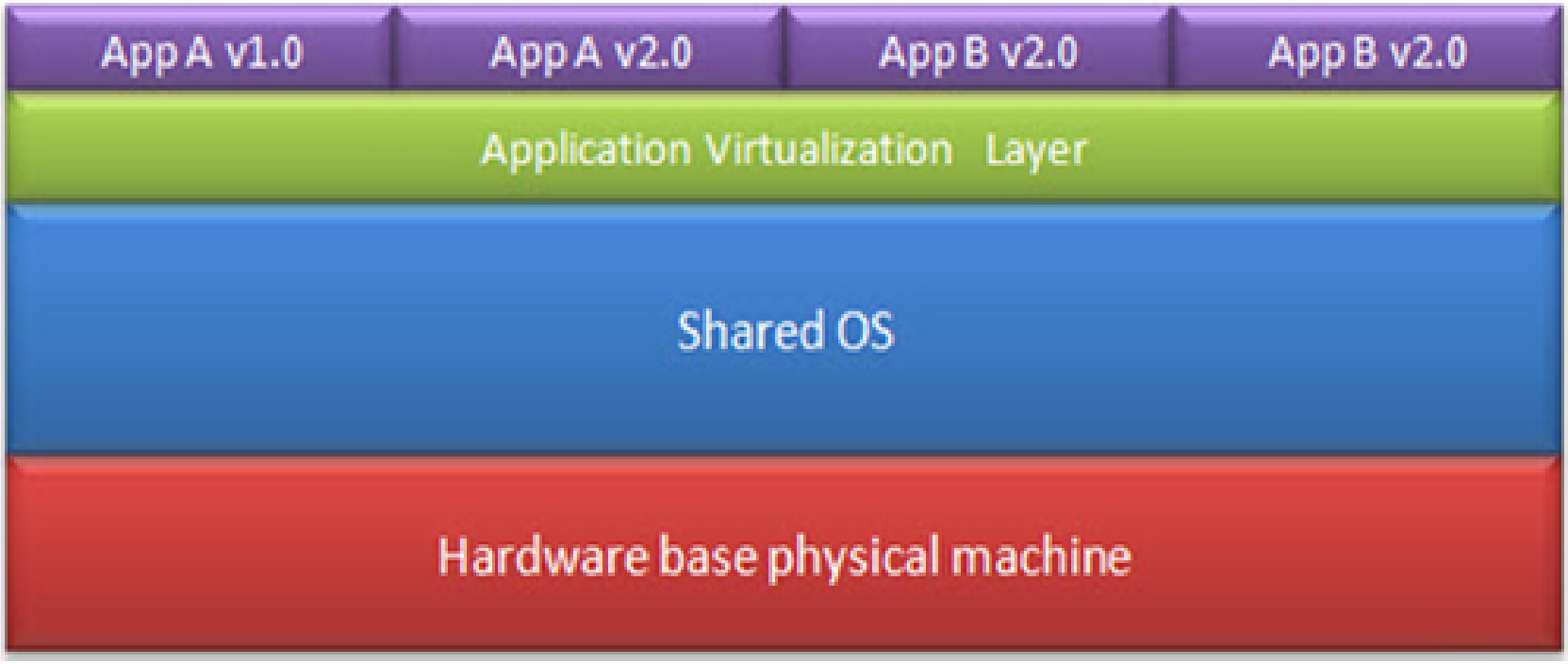


Virtualización a nivel de Sistema Operativo

- Rápida y eficiente
- Soporta un importante número de instancias virtuales (todas del mimo SO)
- El kernel proporciona diferentes espacios de usuario del SO
- La capa de virtualización proporciona un Sistema de archivos propietario y una capa de abstracción de servicio de kernel que garantiza el aislamiento y seguridad de los recursos entre distintos contenedores
- El objeto en el que tiene lugar la virtualización se denomina contenedor
- Ejemplos: LXC, LXD, Docker, OpenVZ, chroot, XenApp...
- Riesgos de seguridad
- Un contenedor no lanza una nueva instancia de sistema operativo, es un proceso del host OS con cierta cantidad de recursos que el contenedor le permite usar como la RAM, el espacio de direcciones “virtual”, puertos de conexión, etc..

Virtualización de Aplicaciones

- Requiere una capa de virtualización de aplicación
- Esta capa genera un registro del sistema y un file system virtual donde se realizan los cambios necesarios para ejecutar la aplicación sin que estos se realicen en los sistemas del SO subyacente.



Virtualización de Aplicaciones

- Las llamadas de la aplicación virtualizada al sistema de ficheros son redirigidas a una localización virtual
- La aplicación se abstrae de la plataforma física
- Mejora la portabilidad de las aplicaciones
- Se pueden ejecutar en distintos sistemas operativos
- Permite la ejecución de aplicaciones incompatibles
- Ejecución más lenta de las aplicaciones
- No todo el software se puede virtualizar

crea un entorno para la aplicación a ejecutar (dll Necesarias, claves de registro y modificaciones en la estructura de archivos).

Formatos de disco

- VDI – formato interno de los discos de VMs VirtualBox.
- qcow2 - formato de KVM comprimido.
- VMDK – El más común porque es el que emplea VMware. Hay varias versiones.
- VHD – el que usa Microsoft en hyper-V
- raw (.img, .raw, etc.) – es disco servido directamente desde un dispositivo local o remoto que no constituye datastore. De esta forma se elimina un intermediario y se favorece el rendimiento.

Formatos de VMs

- Open Virtualization Format (OVF), XML, un OVF describe únicamente a una máquina virtual con todos sus elementos.
- Open Virtual Appliance (OVA), versión empaquetada del OVF
- Vagrant “boxes”, formato Vagrant (devOps) específico para cada hypervisor.

Soporte Hardware Virtualización: AMD-V, Intel-VT-x

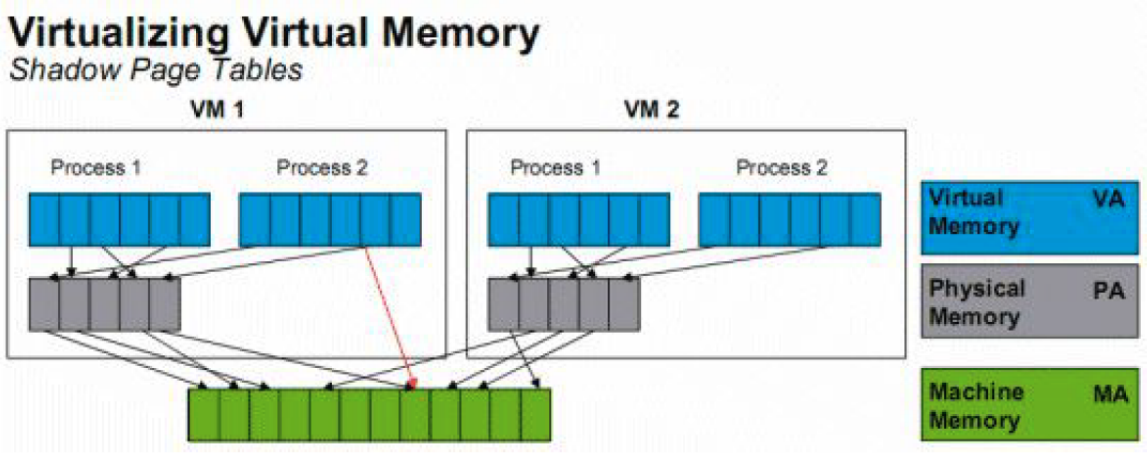
Lógica implementada en hardware específica para mejorar la ejecución de instancias virtuales.

- Permite que el software cree máquinas virtuales de forma más eficiente de tal forma que múltiples sistemas operativos y sus aplicaciones puedan ejecutarse simultáneamente en la misma computadora.
- Extensiones de virtualización al conjunto de instrucciones x86. Ej. en Intel: VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXOFF y VMXON
- La memoria virtual del guest se gestiona accediendo a hardware específico de paginación (TLB).
- VPID, ASID: Permiten identificar sobre el hardware real todos los objetos de las VM
- Virtualización de E/S: acceso directo de VMs al dispositivo, evitando el hipervisor. (AMD-Vi/Intel-VT-d)
- Virtualización de Red: Controlador ethernet con capacidad de virtualización (Intel-VT-c)

Soporte Hardware Virtualización: AMD-V, Intel-VT-x

- Software-based memory virtualization. La TLB (Translation look-aside buffer) es una memoria caché, mantiene mapping entre direción virtual a fisica de la máquina real.

VA: Virtual Address (guest)
PA: Physical Address (guest)
Machine Address (host)



- Hardware-Assisted Memory Virtualization
  - Las CPUs proveen soporte hardware usando 2 niveles de tablas de página. La primera tabla almacena la traducción virtual a física. La segunda la traducción a dirección de máquina real.
- Virtualización anidada
- *cat/proc/cpuinfo | egrep '(vmx|svm)'*

VirtualBox - Introducción

Hypervisor con Virtualización Completa. Con Escritorio y shell.

- Clonar VMs
- Exportar VMs
- Gestionar discos
- Compartir recursos
- Migrar VMs (teleporting)
- Carpeta compartida: host (anfitrión) y Guest VMs
- Snapshots
- Gestionar redes

VirtualBox Networking

Modos de red soportados

- NAT
- Red NAT
- Red solo-anfitrión
- Internal network
- Bridge

	VM ↔ Host	VM1 ↔ VM2	VM → Internet	VM ← Internet
Host-only	+	+	–	–
Internal	–	+	–	–
Bridged	+	+	+	+
NAT	–	–	+	Port forwarding
NAT Network	–	+	+	Port forwarding

VirtualBox Shell

Modo avanzado de configuración. **VBoxManage**

- Operaciónes con VMs: Crear, modificar, listar, clonar, importar, exportar, snapshopt, migrar
- Gestionar estado VMs: Iniciar, Parar, reiniciar, pausar
- Operaciones de Red
- Gestionar discos: cambiar formato, redimensionar...
- Gestionar dispositivos: keyboard, CD, Audio, usb, ..

VBoxManage

Uso Comando

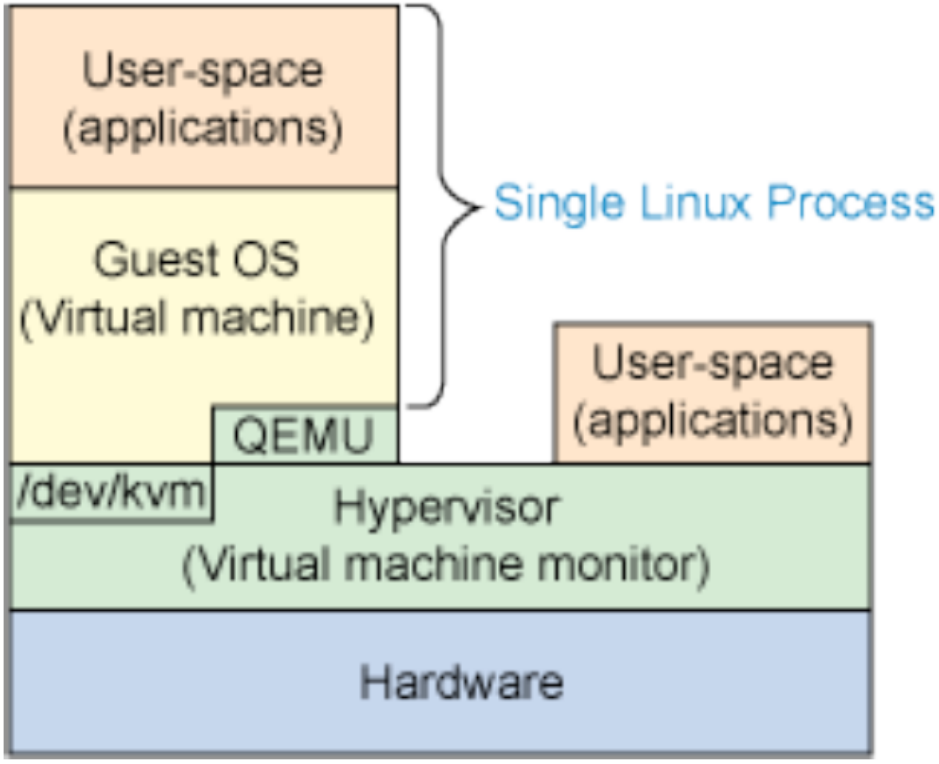
```
VBoxManage [<general option>] <command>
```

Ejemplos

- `VBoxManage createvm --name 'SUSE 10.2' --register`
- `VBoxManage list vms|hdds|natnets`
- `VBoxManage modifyvm 'Windows XP' --memory 512`
- `VBoxManage startvm 'Windows XP'`
- `VBoxManage showvminfo 'Windows XP'`
- `VBoxManage import WindowsXp.ovf --dry-run`
- `VBoxManage createmedium ...`
- `VBoxManage storagectl --add ide|sata|scsi`

KVM

- KVM es una solución de virtualización completa para Linux en hardware x86 que utiliza extensiones de virtualización (Intel VT o AMD-V).



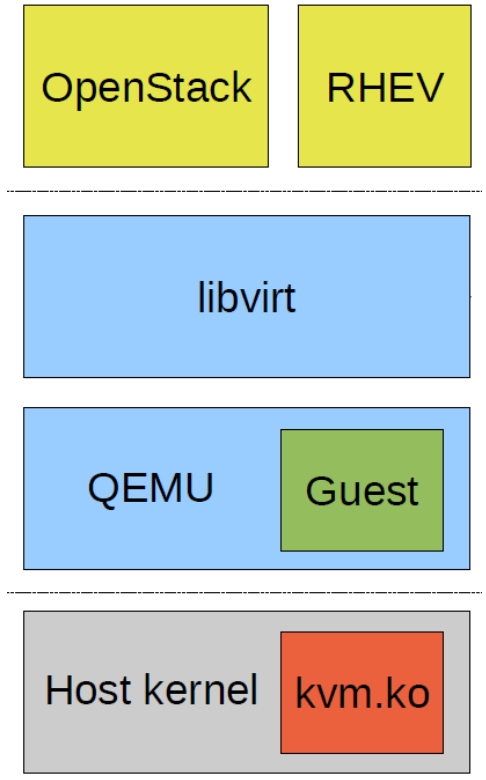
KVM

Kernel-based Virtual Machine

- Consiste en un modulo del kernel, kvm.ko, que proporciona la infraestructura de virtualización principal y un módulo específico del procesador, kvm-intel.ko o kvm-amd.ko.
- El componente del kernel de KVM está incluido desde Linux (2.6.20).
- KVM es open source.
- KVM puede ejecutar huéspedes Linux/Unix/Windows de 32 o 64 bits
- QEMU para emular hardware.
- libvirt para gestionar las maquinas virtuales
- "Virtual Machine Manager" como aplicación de escritorio

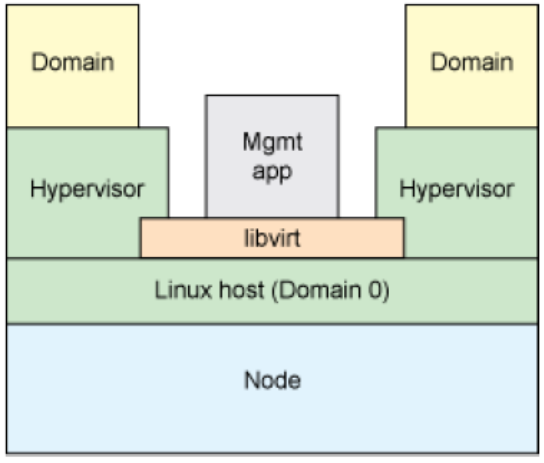
KVM-QEMU-Libvirt

QEMU se despliega en el espacio de usuario de la máquina host y usa el driver kvm.ko para ejecutar los procesos del invitado (guest): Soluciones de Cloud Computing (ej. Openstack) utilizan libvirt como libreria para proporcionar Infrastructure as a Service (IaaS)



Libvirt

- Libvirt es una colección de software (servicio) para administrar máquinas virtuales y otras funcionalidades de virtualización.
- Incluye un API en C, un daemon (libvirtd) y una utilidad de línea de comandos (virsh).
- Un objetivo principal de libvirt es proporcionar una única forma de gestionar múltiples proveedores / hipervisores de virtualización diferentes
- Hipervisores KVM / QEMU, Xen, LXC, OpenVZ o VirtualBox (entre otros).



Libvirt

Algunas de las principales características de libvirt:

- Gestión de VM: varias operaciones del ciclo de vida del dominio (VM), como iniciar, detener, pausar, guardar, restaurar y migrar. Operaciones Hotplug para muchos tipos de dispositivos, incluidas las interfaces de disco y de red, la memoria y las CPU.
- brownAdministración remota: se puede acceder a todas las funciones de libvirt en cualquier máquina que ejecute libvirt daemon, incluidas las máquinas remotas. Se admite una variedad de transportes de red para conectarse de forma remota, siendo el SSH el más simple, que no requiere una configuración explícita adicional.
- Administración de almacenamiento: cualquier host que ejecute libvirt daemon se puede usar para administrar varios tipos de almacenamiento: crear imágenes de archivos de varios formatos (qcow2, vmdk, raw, ...), montar recursos compartidos NFS, enumerar grupos de volúmenes LVM

Libvirt

Algunas de las principales características de libvirt:

- Gestión de interfaz de red: cualquier host que ejecute libvirt daemon puede utilizarse para gestionar interfaces de red físicas y lógicas. Enumerar las interfaces existentes, así como configurar (y crear) interfaces, puentes, vlans y dispositivos de enlace.
- Redes virtuales basadas en NAT y rutas: cualquier host que ejecute libvirt daemon puede administrar y crear redes virtuales. Las redes virtuales Libvirt usan reglas de firewall para actuar como un enrutador, proporcionando a las VM acceso transparente a la red de máquinas host.

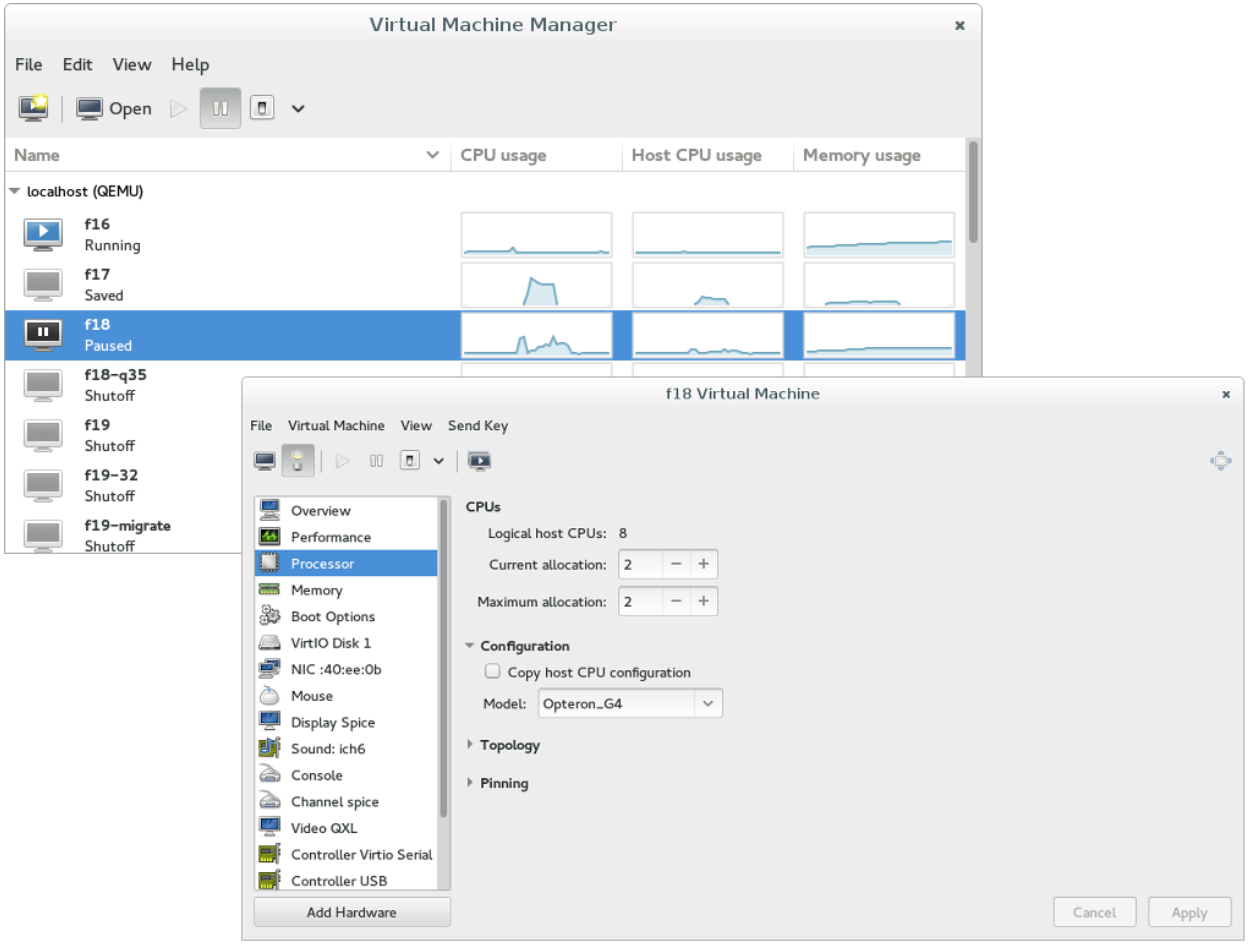
Libvirt - Virsh

Ejemplos comandos virsh:

- `virsh connect qemu:///system`
- `virsh create <path XML file>`
- `virsh -c qemu:///system start PruebaVM`
- `virsh save [domain-name][domain-id | domain-uuid][filename]`
- `virsh shutdown [domain-id | domain-name | domain-uuid]`
- `virsh dominfo [domain-id | domain-name | domain-uuid]`
- `virsh list domain-name [ ||inactive | || -all]`
- `virsh setvcpus [domain-name|domain-id|domain-uuid] [count]`
- `virsh net-list`
- Muchos comandos requiren ejecturarse como root
- Los comandos requieren que el demonio libvirtd este operativo

Virtual Machine Manager

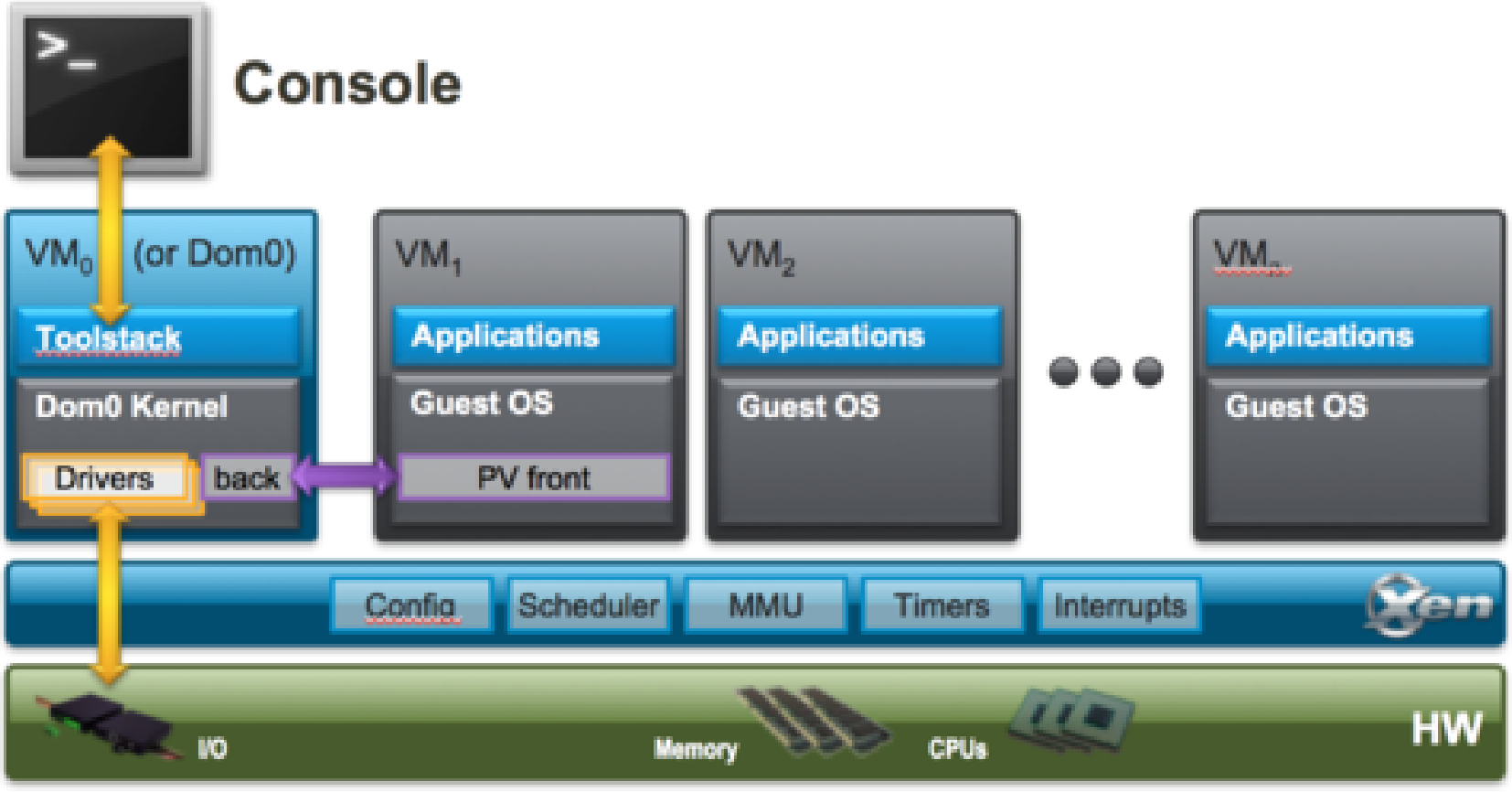
- Gestión de VMs de tipo KVM principalmente y también Xen y LXC (linux containers)
- virt-viewer: interfaz gráfica para connectar a la VM por VNC



XEN

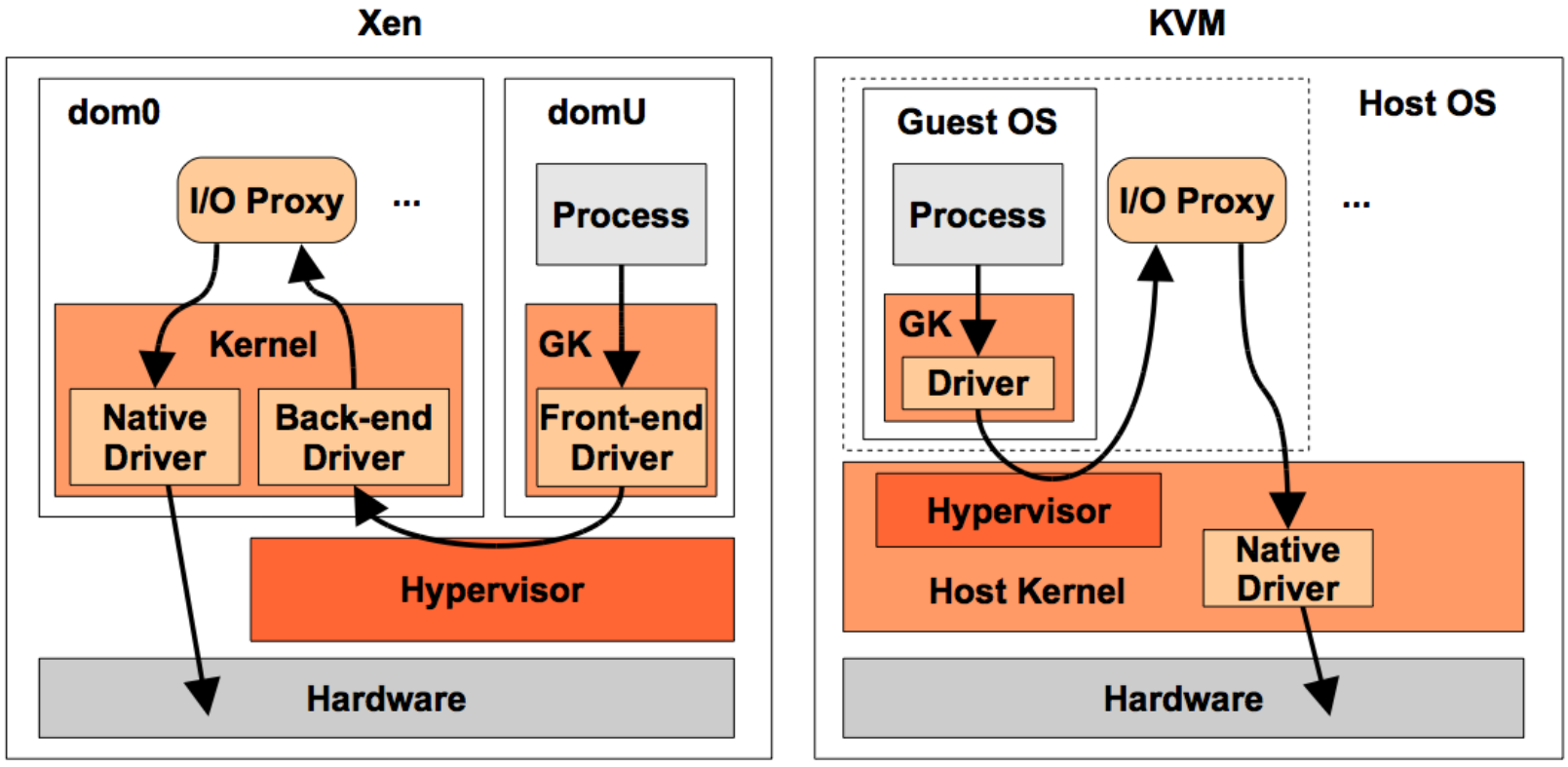
- El hypervisor se ejecuta directamente en el hardware y es responsable del manejo de la CPU, la memoria y las interrupciones. Es el primer programa que se ejecuta después de salir del gestor de arranque. Encima del hipervisor ejecuta varias máquinas virtuales.
- El hypervisor en sí no tiene conocimiento de las funciones de E/S, como las redes y el almacenamiento.
- Una instancia en ejecución de una máquina virtual se llama dominio o invitado.
- Un dominio especial, llamado dominio 0 contiene los controladores para todos los dispositivos en el sistema.
- El dominio 0 también contiene una pila de control para administrar la creación, destrucción y configuración de la máquina virtual.
- UNIX-based como el sistema anfitrión

XEN


Console
VM0 (or Dom0)
Toolstack
Dom0 Kernel
Drivers
back
VM1
Applications
Guest OS
PV front
VM2
Applications
Guest OS
VMn
Applications
Guest OS
Config
Scheduler
MMU
Timers
Interrupts
Xen
I/O
Memory
CPUs
HW

KVM vs XEN

Figura: Drs. Albert Spijkers

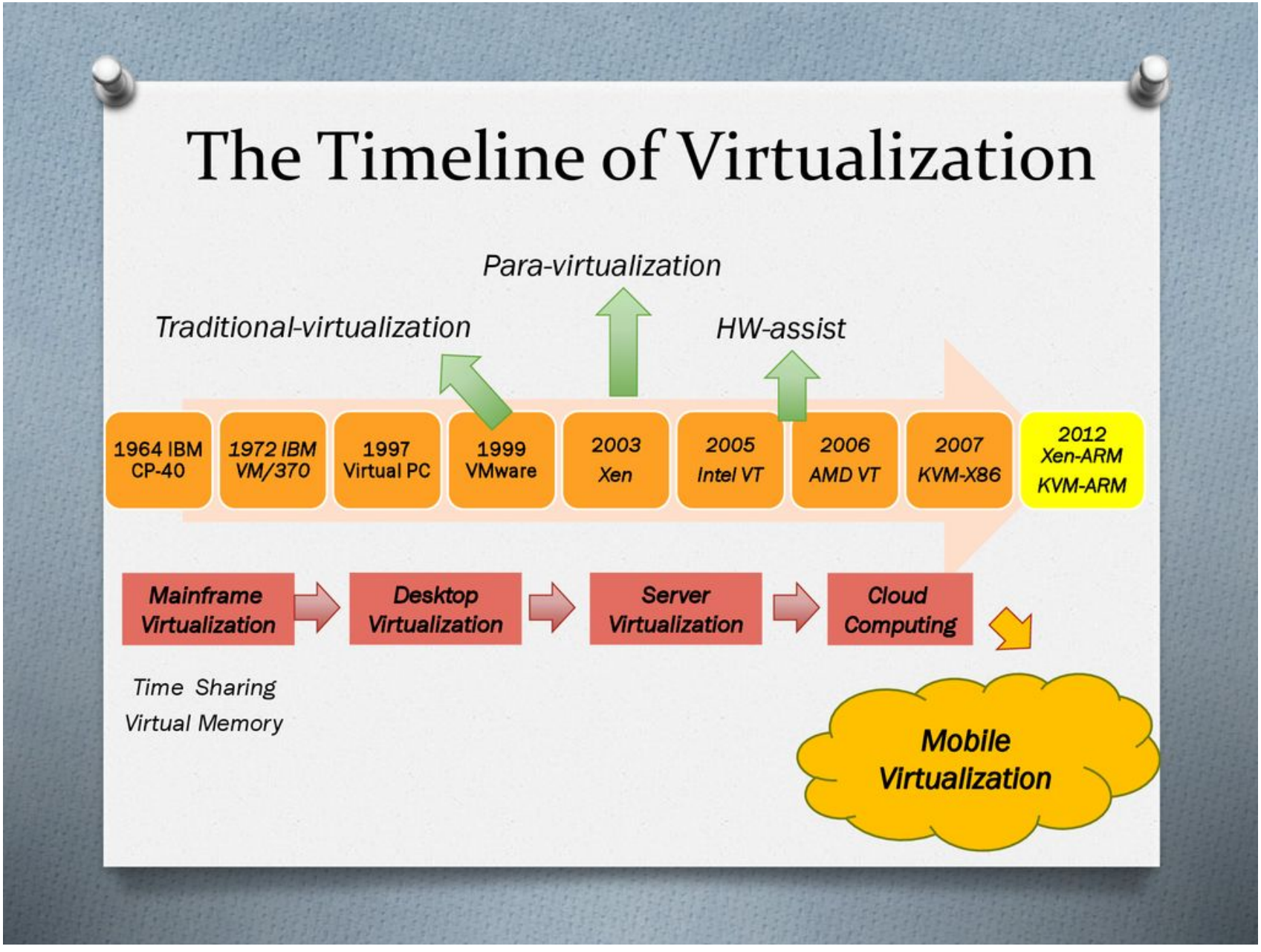


Correr un hipervisor XEN exige un kernel XEN. Linux soporta ser guest XEN pero no ser host XEN. Toca recompilar o comprar (Citrix)

VMWARE

La empresa VMware fue la pionera en popularizar la virtualización aplicada al PC y al servidor corporativo.

- VMware Workstation/Fusion/Player: versiones de escritorio de su VMM de tipo 2, cuyo ámbito es la virtualización en un host.
- VMware ESXi: hypervisor de tipo 1
  - Full virtualization: Asistida o No asistida por hardware
  - Paravirtualización de dispositivos (no de CPU)
  - Gratuita el primer mes, luego se desactivan algunas funcionalidades avanzadas.
  - Networking con vSwitch donde se crear redes que conentan VMs
- VMware vSphere: la consola universal de gestión del DataCenter VMware


The Timeline of Virtualization
Para-virtualization
Traditional-virtualization
HW-assist
1964 IBM CP-40
1972 IBM VM/370
1997 Virtual PC
1999 VMware
2003 Xen
2005 Intel VT
2006 AMD VT
2007 KVM-X86
2012 Xen-ARM KVM-ARM
Mainframe Virtualization
Desktop Virtualization
Server Virtualization
Cloud Computing
Time Sharing
Virtual Memory
Mobile Virtualization

Referencias

- "Mastering Cloud Computing", Foundation and applications Programming, Rajkumar Buyya, Editorial: Morgan Kaufmann
- VirtualBox `https://www.virtualbox.org/wiki/Documentation`
- VirtualBoxManage `https://www.virtualbox.org/manual/ch08.html`
- Virsh Command Reference `https://libvirt.org/sources/virshcmdref/html-single/`
- Virsh documentation in Ubuntu `http://manpages.ubuntu.com/manpages/xenial/en/man1/virsh.1.html`
- KVM `www.linux-kvm.org`
- XEN `http://www.xen.org/`

FIN