

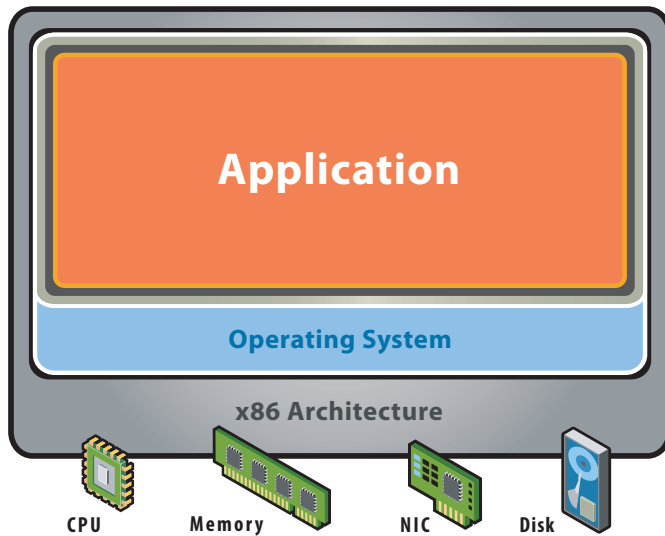
Computación en la Nube

Virtualización

Que es Virtualización?

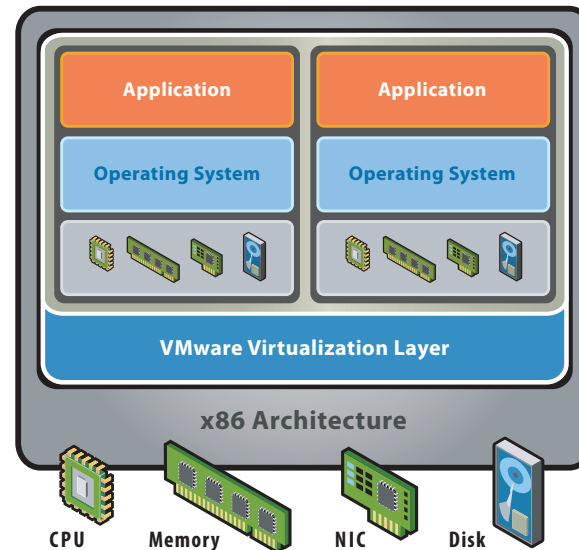
- En filosofía, virtual significa "algo que no es real".
- En informática, virtual significa "un entorno de hardware que no es real".
- En virtualización se duplican (virtualizan) las funciones del hardware físico y se le presentan a un sistema operativo
- El sistema físico que ejecuta el software de virtualización (hipervisor o Virtual Machine Monitor - VMM) se denomina host y las máquinas virtuales instaladas en la parte superior del hipervisor se llaman invitados (guests).

Nivel de abstracción entre recursos físicos y aplicaciones



Before Virtualization:

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure



After Virtualization:

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

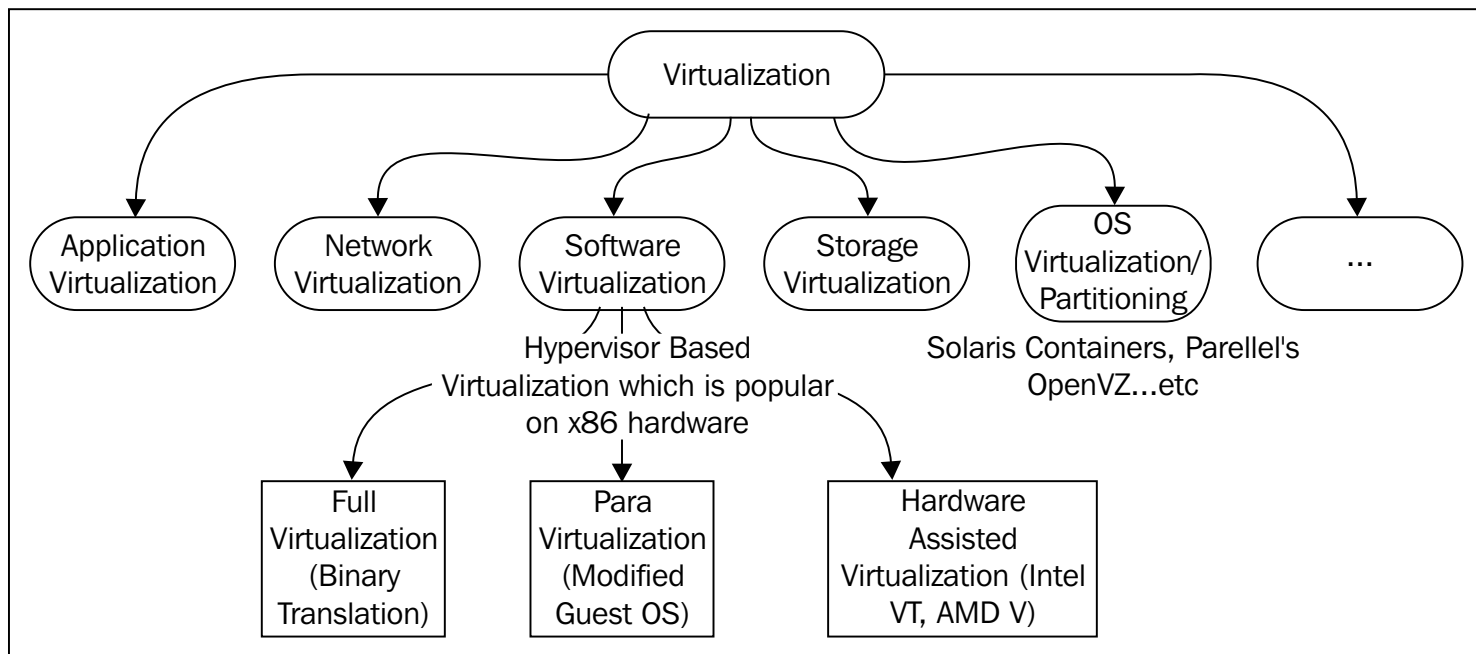
Por que usar virtualización?

- Consolidación de servidores
 - Tasas de utilización de servidores se incrementan de 5-15% a 60-80%
- Aislamiento de Servicios
 - Se eliminan problemas de compatibilidad entre aplicaciones
- Optimización de procesos de desarrollo y pruebas
 - Reúso de sistemas pre configurados
- Continuidad del negocio frente a fallas
 - Alta disponibilidad y soluciones para recuperación ante desastres
- Balanceo de carga dinámico
 - Migración en caliente

Por que usar virtualización?

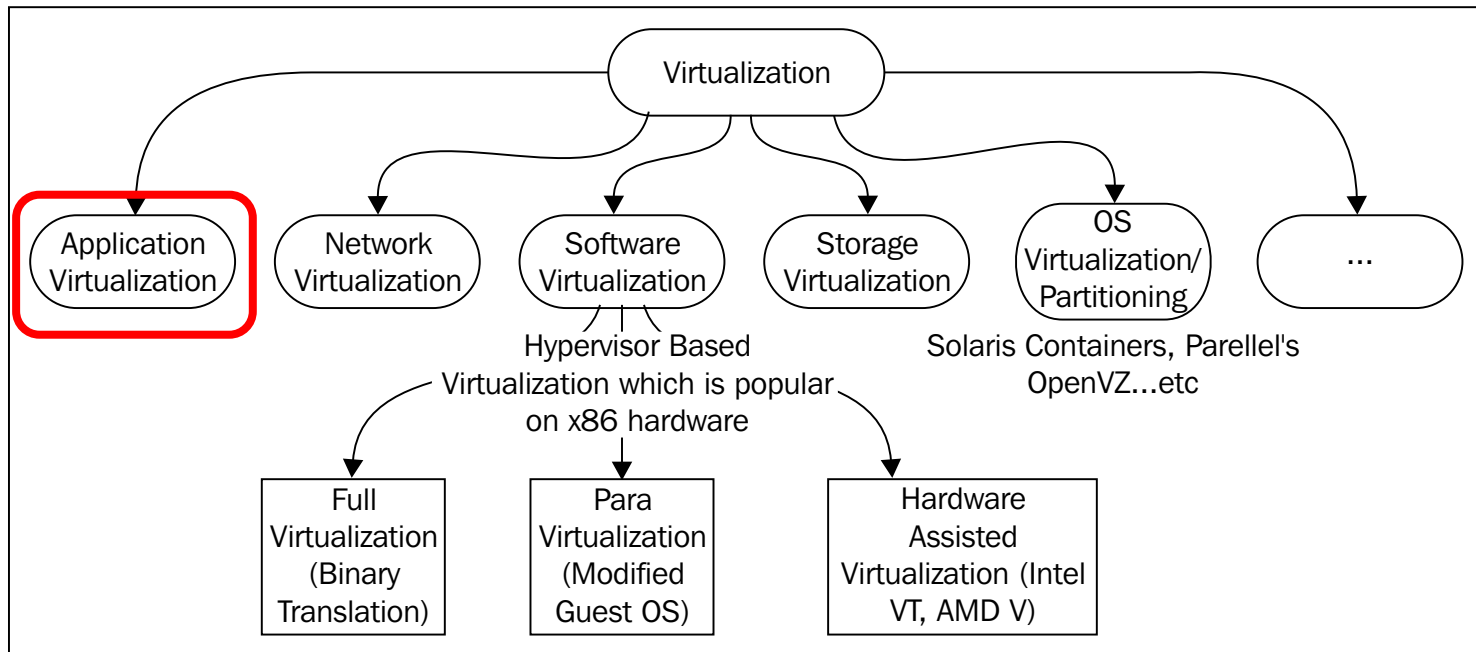
- Rápido aprovisionamiento de servidores
 - Creación de maquinas virtuales desde imágenes preconfiguradas
- Confiabilidad y seguridad del sistema mejorada
 - Se agrega un nivel de abstracción entre la maquina virtual y el hardware
 - En el caso de datos corruptos se presentaran fallas en el disco duro virtual y no en el disco duro del host
 - Configuración de seguridad particularizada para cada VM

Tipos de Virtualización



Fuente: Mastering KVM Virtualization

Tipos de Virtualización

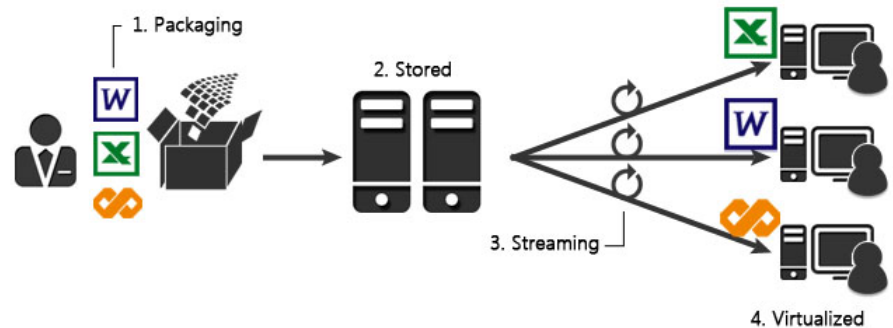


Fuente: Mastering KVM Virtualization

Virtualización de Aplicaciones

Ejemplos:

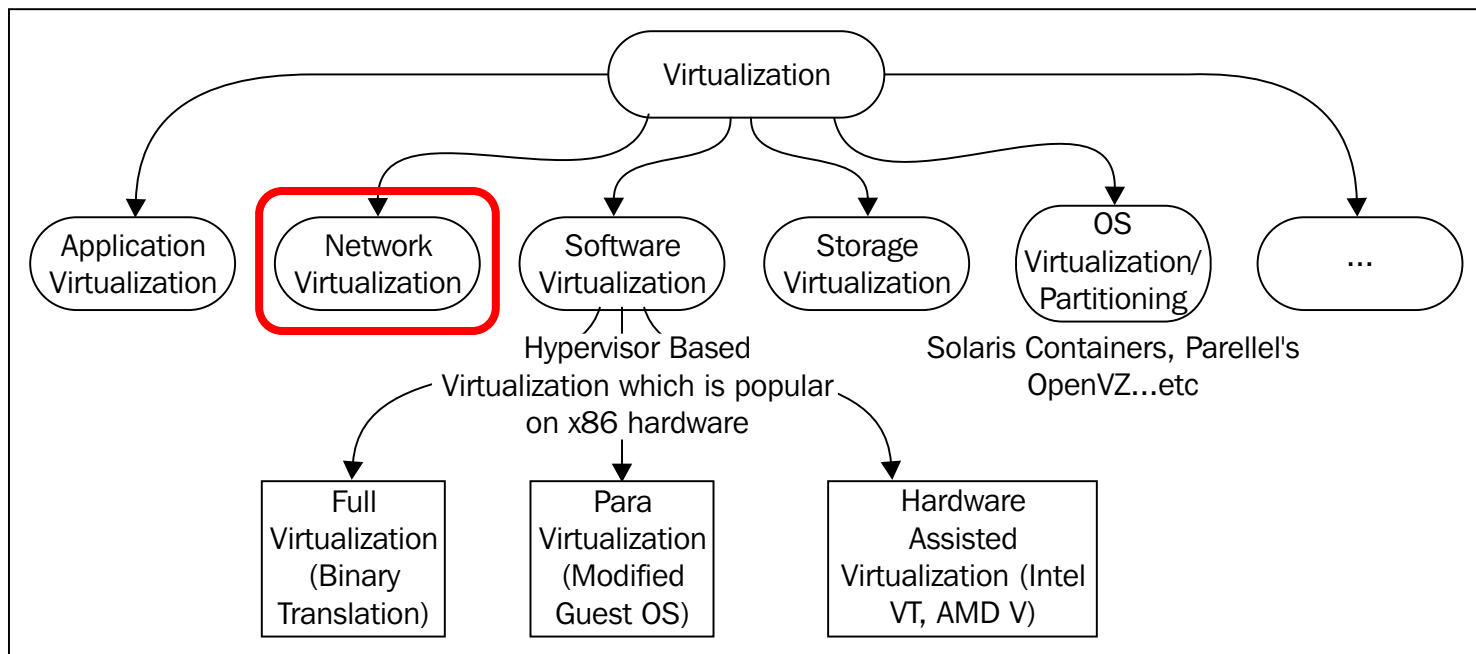
- Streaming de Aplicaciones
(Ej. Microsoft 360 Click to Run)
- Escritorio Remoto



Fuente: <http://www.softonnet.com/eng/technologies/application-virtualization>

- Cuando los usuarios inician la aplicación, el servidor la transmite a los usuarios en tiempo real.
- A medida que comienza el usuario necesita funciones de la aplicación estas son enviadas en streaming.

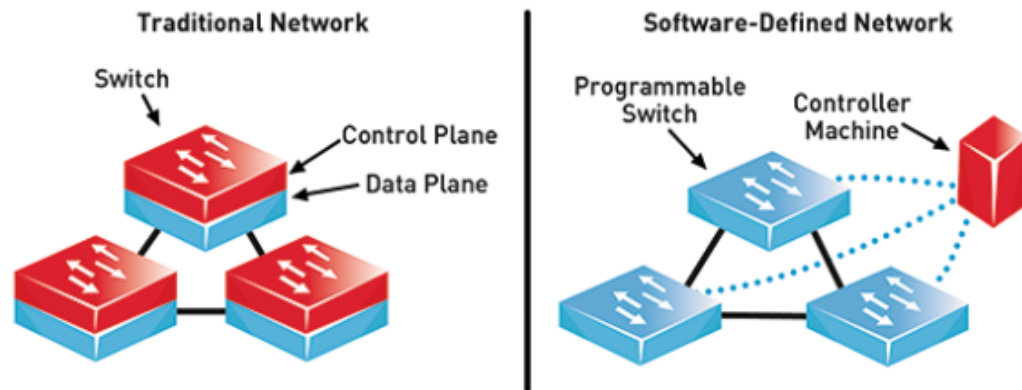
Tipos de Virtualización



Fuente: Mastering KVM Virtualization

Virtualización de Red

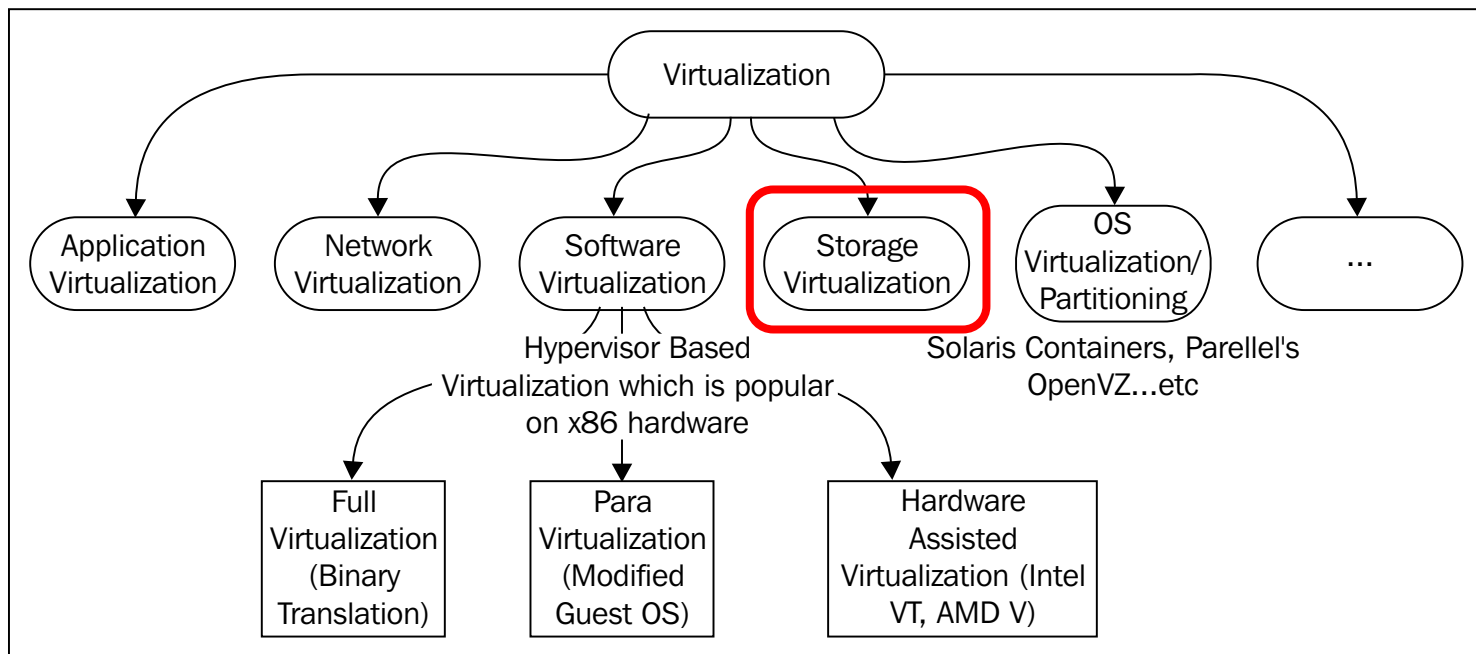
Ejemplo: Software Defined Networking



Fuente: <https://www.commsbusiness.co.uk/features/software-defined-networking-sdn-explained/>

- Se separa el plano de control del plano de datos
- **Plano de Datos:** dispositivos actuando como switches, routers o access points (dependiendo como sean programados)
- **Plano de Control:** Parte inteligente: control de tráfico, balanceo de carga
- Planos de control y datos se comunican usando una interfaz de programación abierta (ej: openflow)

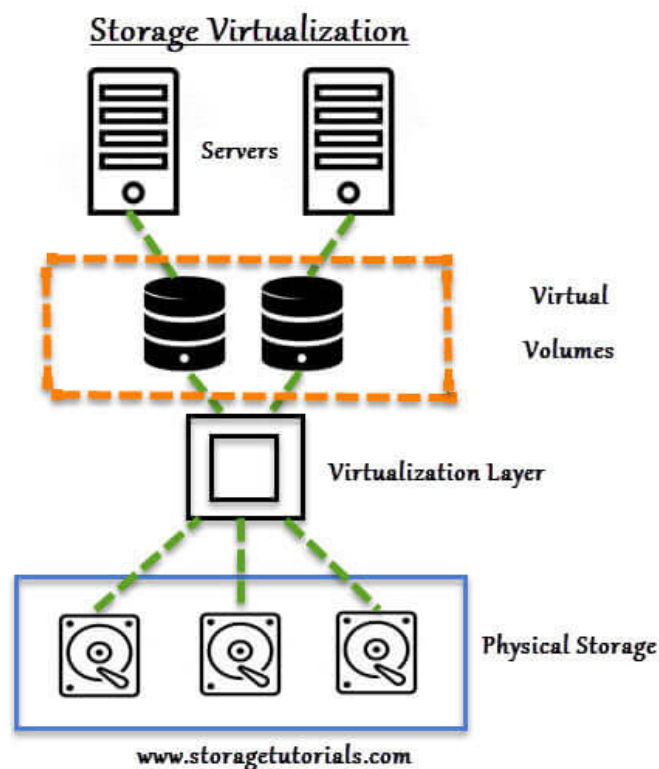
Tipos de Virtualización



Fuente: Mastering KVM Virtualization

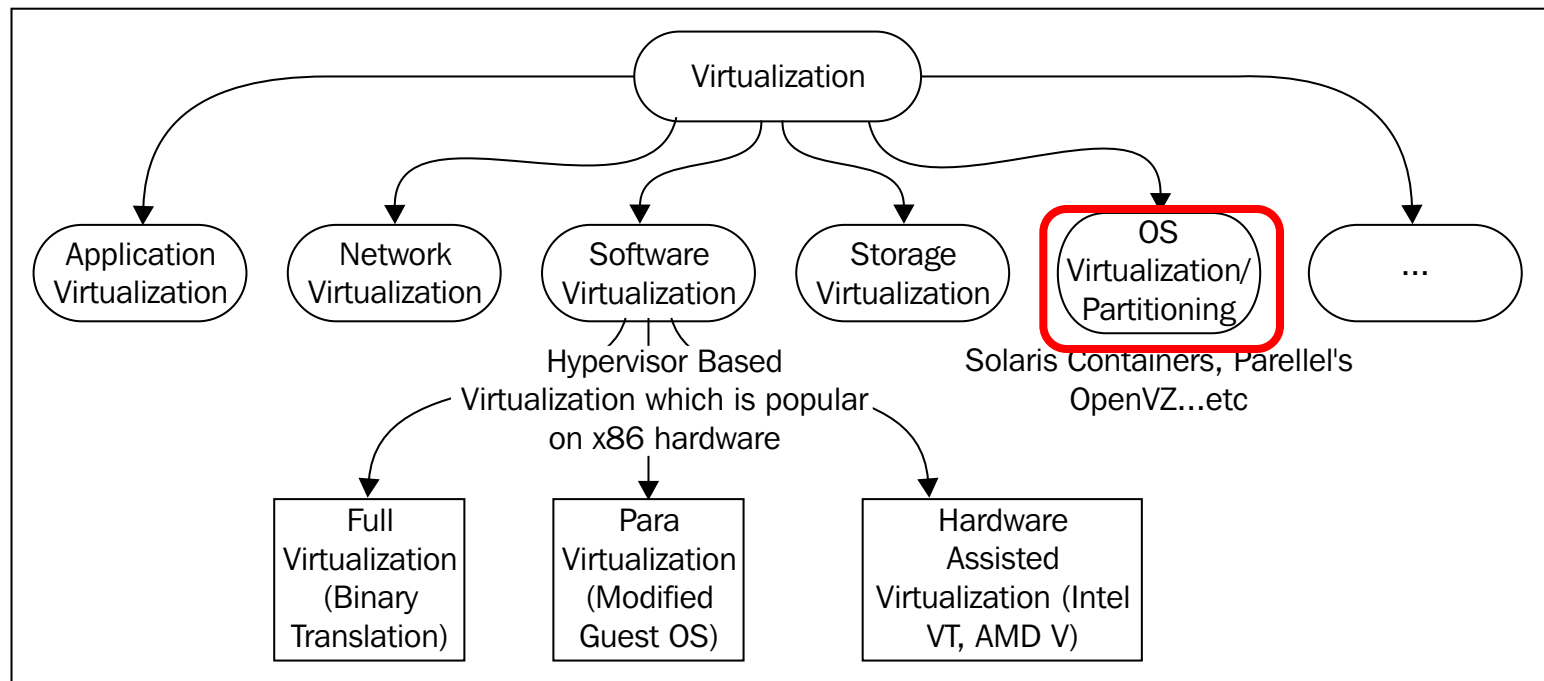
Virtualización de Almacenamiento

En virtualización del almacenamiento, varios discos físicos se combinan en un grupo y de ese grupo de discos físicos, el almacenamiento virtual o los bloques de almacenamiento lógico se asignan a un servidor para su uso.



Virtualización de OS / Particionamiento

Virtualización de OS / Particionamiento



Fuente: Mastering KVM Virtualization

El Host Corre Contenedores Aislados

- La virtualización del sistema operativo permite que el mismo host físico sirva diferentes “workloads” de forma aislada
- Los workloads operan independientemente en el mismo OS
- El host corre múltiples instancias aisladas del sistema operativo llamadas **contenedores**

Ejemplos

- Solaris Containers
- FreeBSD jails
- Parallels
- OpenVZ
- Linux Containers
- Docker

Características

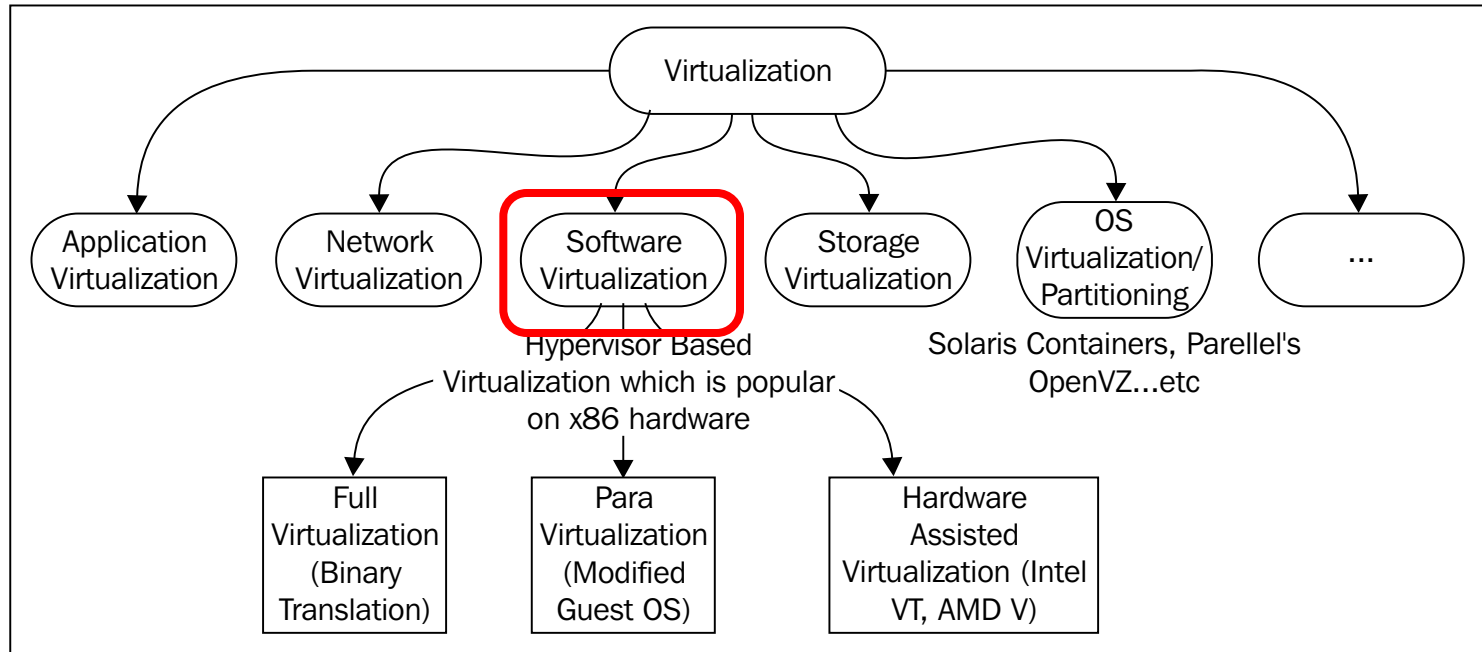
- El aislamiento de procesos y administración de recursos es suministrada por el kernel
- Los containers tienen su propio sistema de archivos, procesos, memoria y dispositivos

Beneficios

- El sistema operativo no necesita emular llamadas de sistema (system calls) de otros sistemas operativos
- El hecho de que un solo sistema operativo soporte a todos los ambientes virtuales (contenedores) se traduce en mejor desempeño y eficiencia
- Moverse de una partición a otra es muy rápido

Virtualización de Software

Virtualización de Software



Fuente: Mastering KVM Virtualization

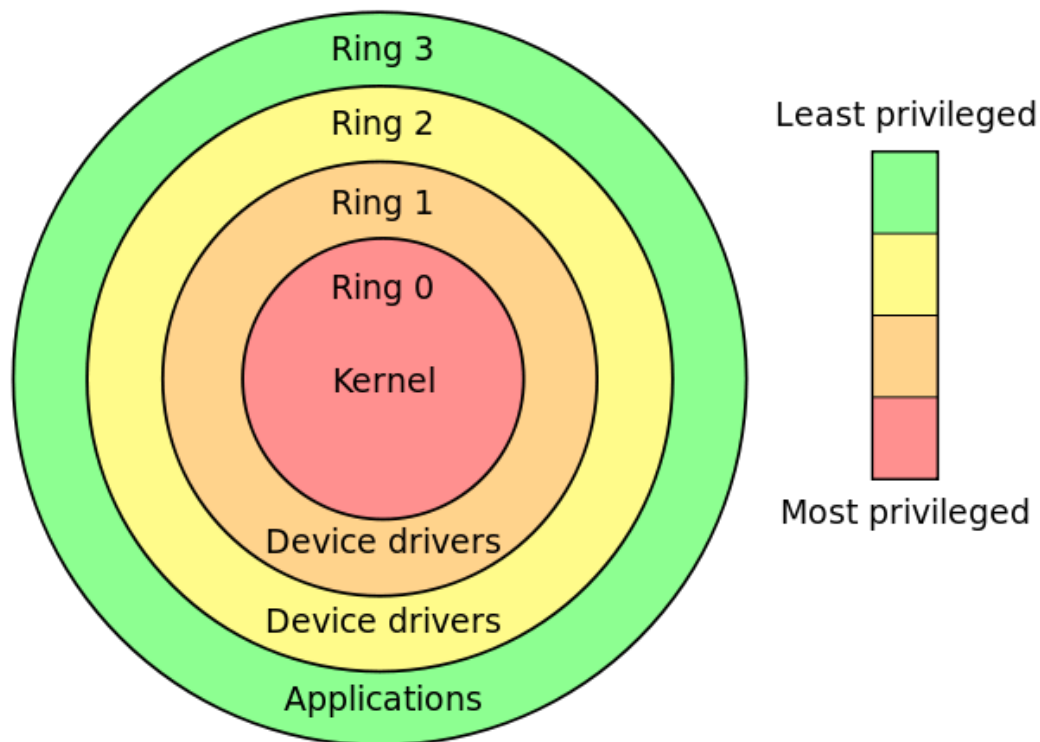
El kernel de un Sistema Operativo

- El kernel (núcleo) es la parte fundamental del sistema operativo y es el encargado de administrar los recursos y permitir que los programas hagan uso de los mismos
- Entre los recursos administrados por el kernel se encuentran: CPU, Memoria, Dispositivos de Entrada/salida
- Adicionalmente el kernel se encarga de:
 - Protección mediante diferentes niveles de acceso
 - Acceso compartido (multiplexado) a los recursos

Niveles de Seguridad

- Las CPUs incluyen diferentes niveles de acceso, que se conocen como anillos (rings)
- Los diferentes kernel suelen utilizar al menos dos niveles para acceder tanto a la CPU como a la memoria:
 - Kernel mode (Ring 0)
 - User mode (Ring 3)

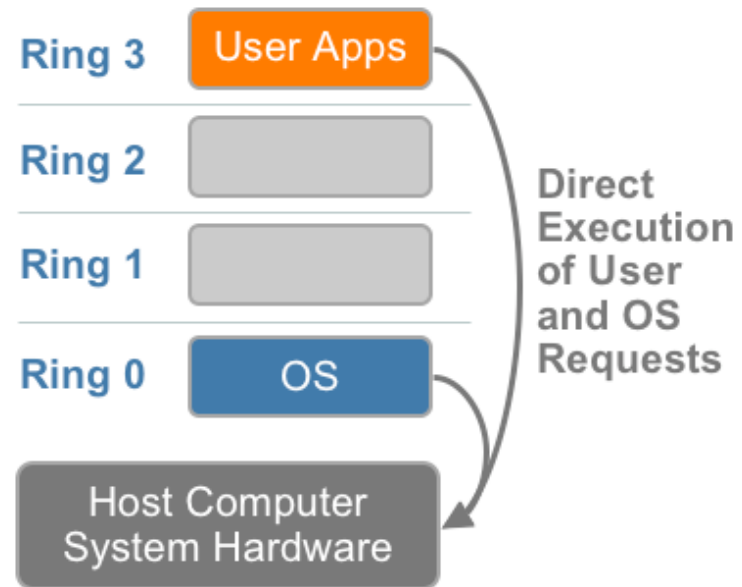
Niveles de Seguridad



Fuente: Mastering KVM Virtualization

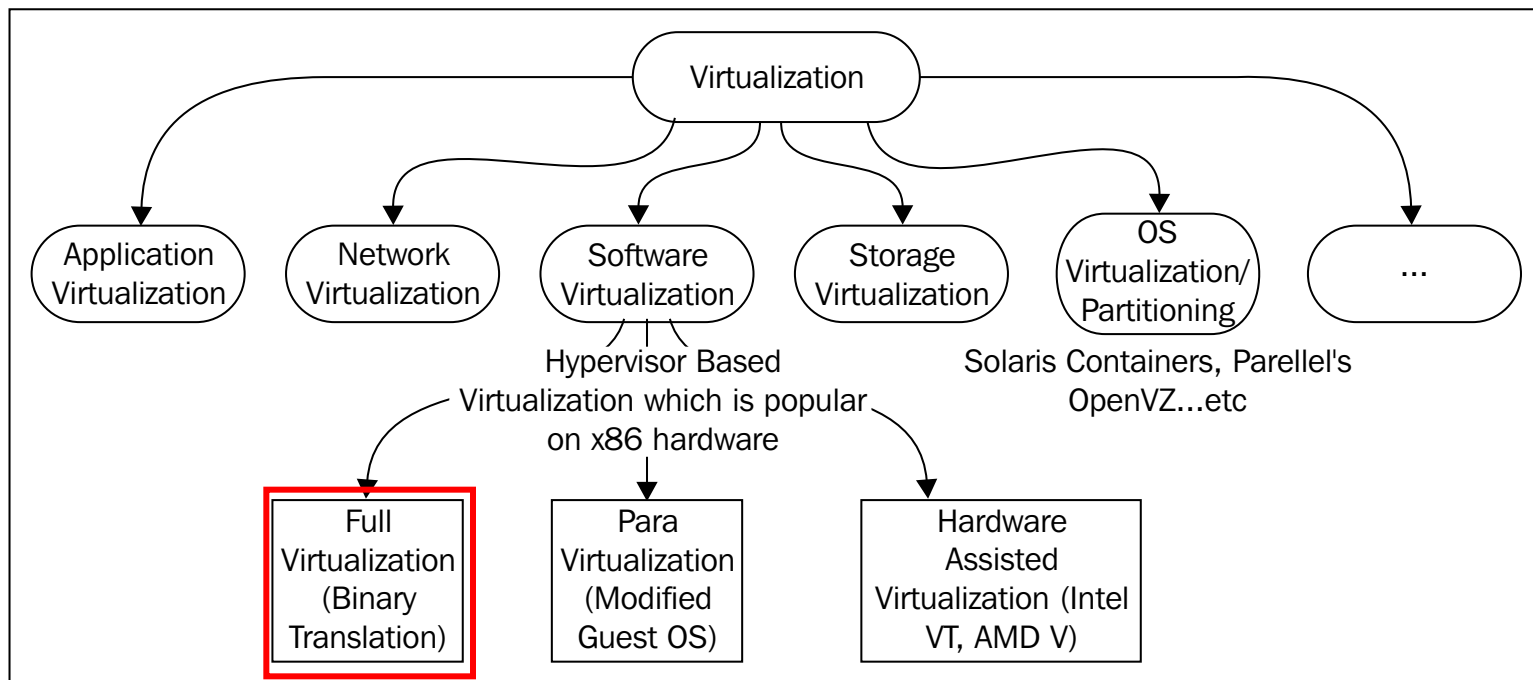
Desafíos de la virtualización basada en Hypervisor

- Los sistemas operativos
 - Están diseñados para correr directamente en “metal”
 - Asumen que controlan el hardware completamente
- Niveles de seguridad
 - Las aplicaciones corren en Ring 3
 - El sistema operativo (kernel) corre en Ring 0
 - Al virtualizar se debe poner un nivel de virtualización por debajo del sistema operativo → **Complicado!**
 - **Algunas instrucciones no pueden virtualizarse completamente!**



Fuente: <http://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html>

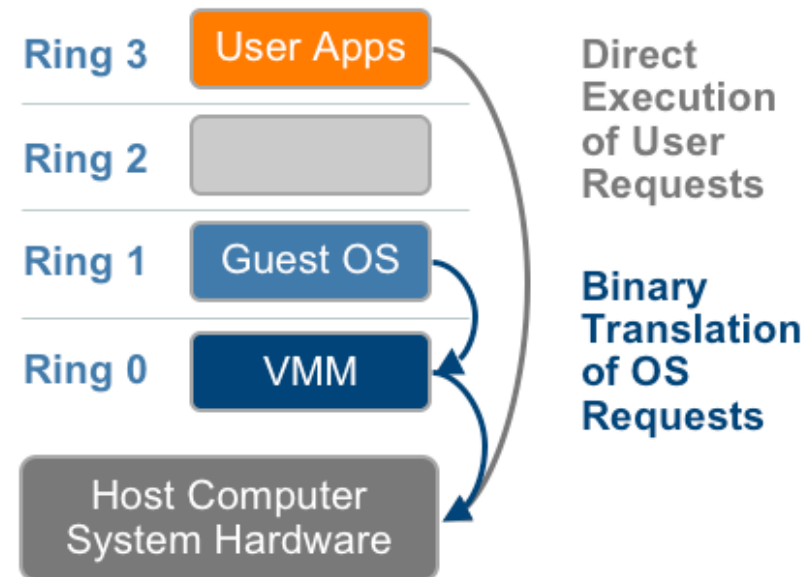
Virtualización de Software



Fuente: Mastering KVM Virtualization

Virtualización Completa (Full Virtualization)

- Usa una combinación de técnicas de **traducción binaria y atrapar y emular**
- El VMM **Traduce instrucciones privilegiadas del kernel** a nuevas instrucciones que tienen el mismo efecto en el hardware virtual
- Estas instrucciones son identificadas y reemplazadas para ser emuladas → **Bajo desempeño** en comparación con otros tipos de virtualización
- Recurso Youtube: [Full Virtualization](#)

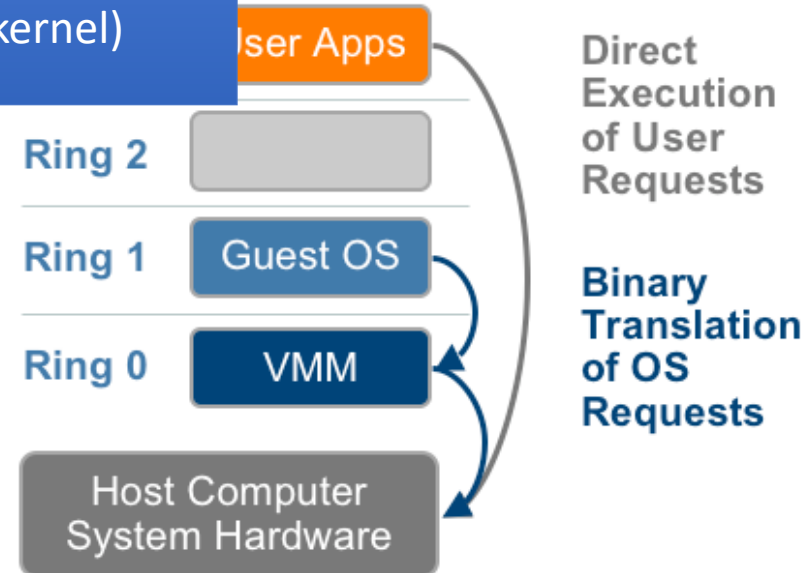


Fuente: <http://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html>

Virtualización Completa (Full Virtualization)

- Usa una combinación de **traducción binaria** y **emular**
- El VMM **Traduce instrucciones privilegiadas del kernel** a nuevas instrucciones que tienen el mismo efecto en el hardware virtual
- Estas instrucciones son identificadas y reemplazadas para ser emuladas → **Bajo desempeño** en comparación con otros tipos de virtualización
- Recurso Youtube: [Full Virtualization](#)

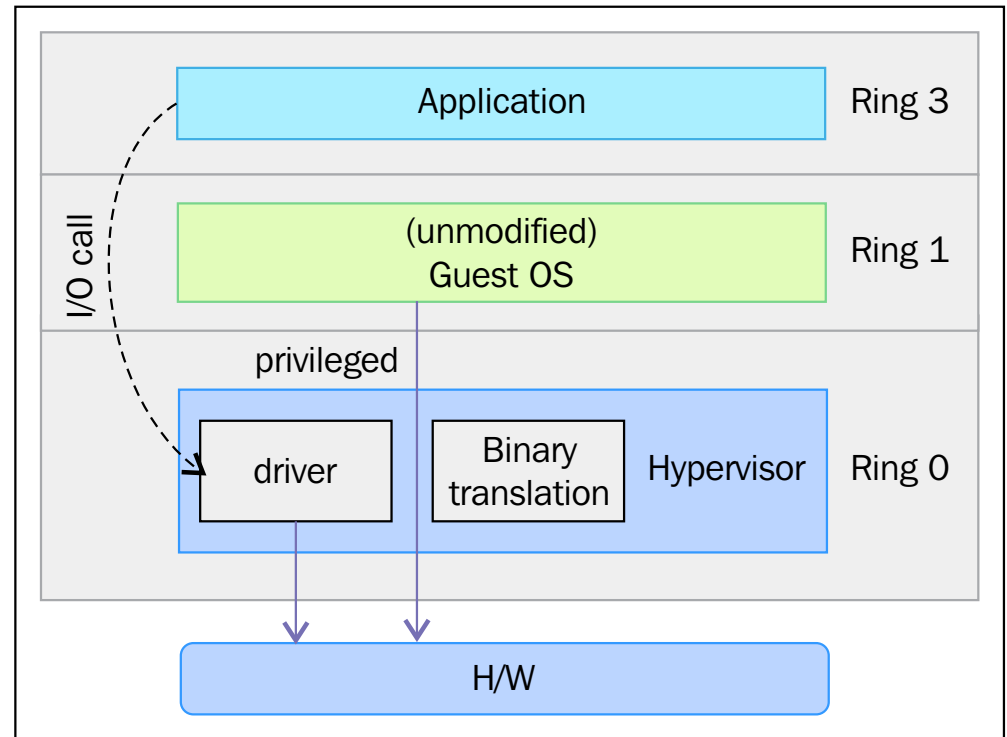
Instrucciones que incluyen operaciones como I/O, manejo de memoria, o acceso a recursos (Se ejecutan en modo kernel)



Fuente: <http://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html>

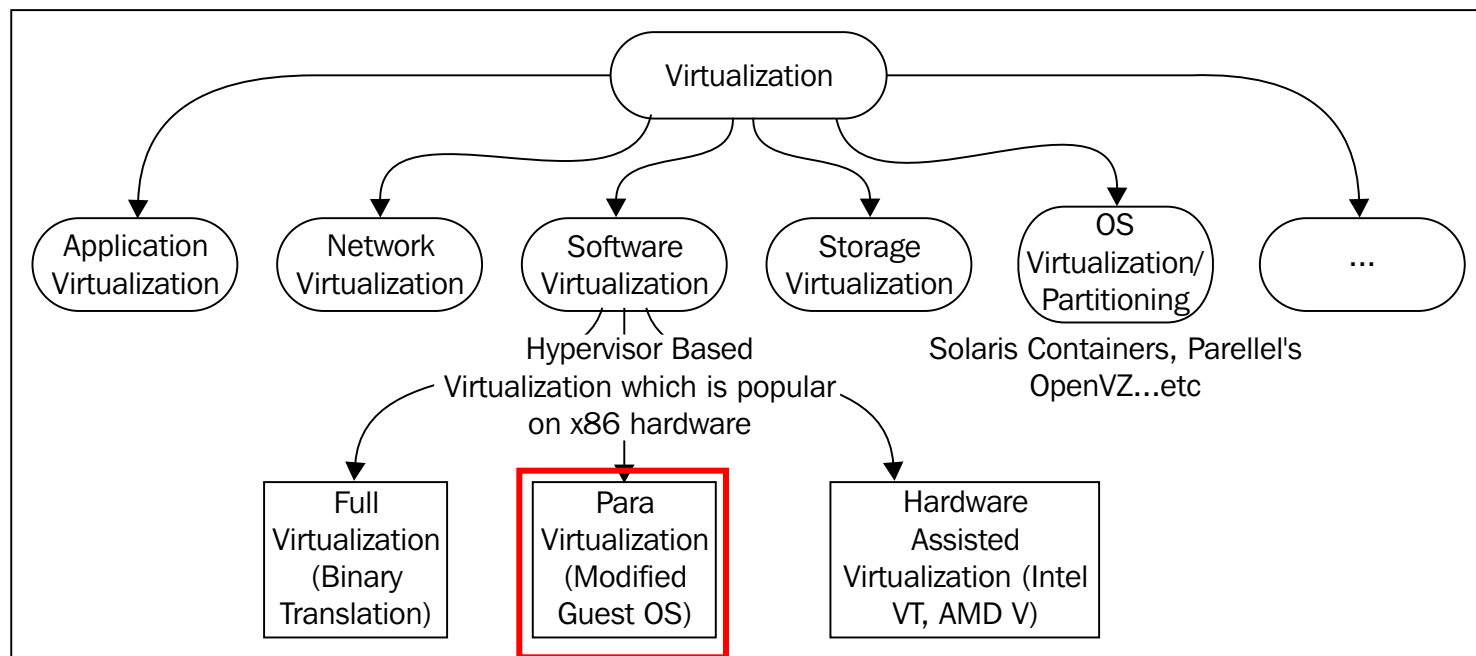
El Guest OS no sabe que esta siendo virtualizado!

En “Full virtualization” se usa el sistema operativo invitado (Guest OS) sin modificación → El sistema operativo invitado no se da cuenta que esta siendo virtualizado



Fuente: Mastering KVM Virtualization

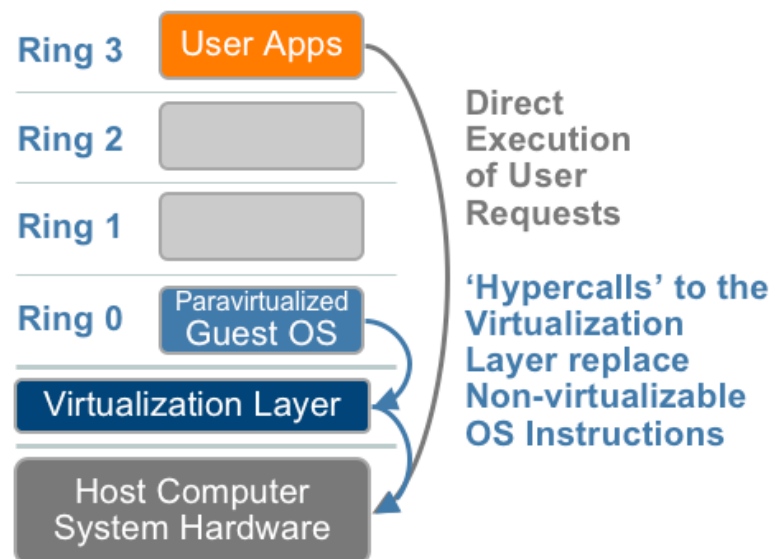
Virtualización de Software



Fuente: Mastering KVM Virtualization

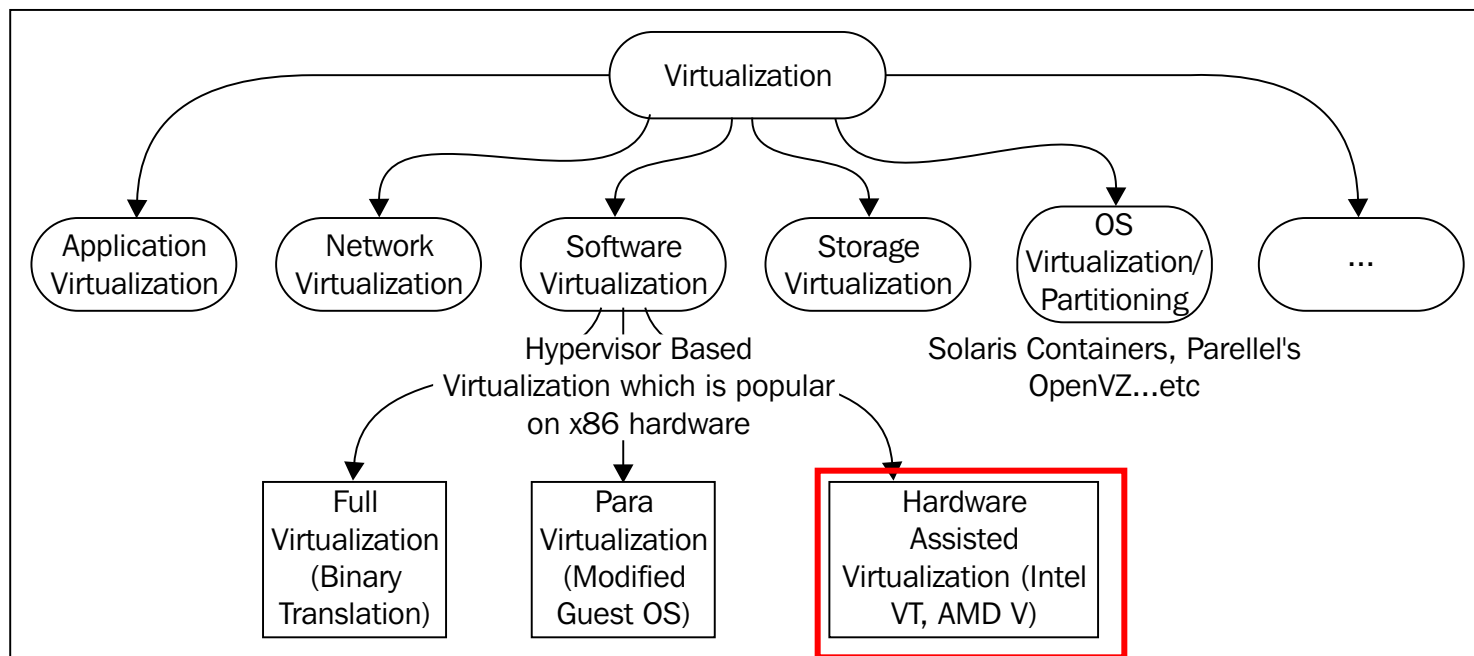
Paravirtualización

- El Guest OS necesita ser modificado para comunicarse con el VMM a través de hiper llamadas (**hypercalls**)
- El hipervisor (VMM) provee un API, el cual es usado por el sistema operativo guest
- **No es necesario usar traducción binaria**
- Ventaja
 - Mejor desempeño que Full virtualization
- Desventaja
 - Portabilidad es pobre
- Recurso YouTube: [Paravirtualization](#)



Fuente: <http://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html>

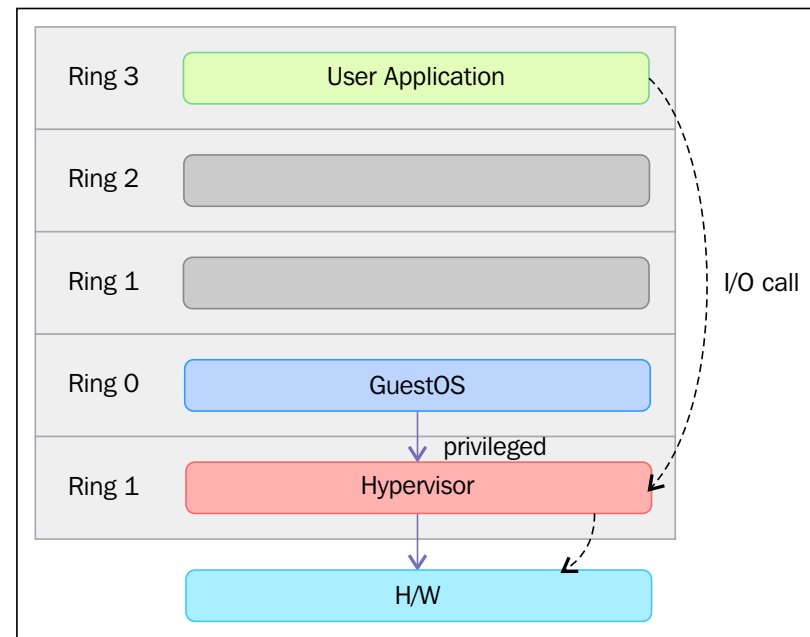
Virtualización de Software



Fuente: Mastering KVM Virtualization

Virtualización Asistida por Hardware (virtualización nativa)

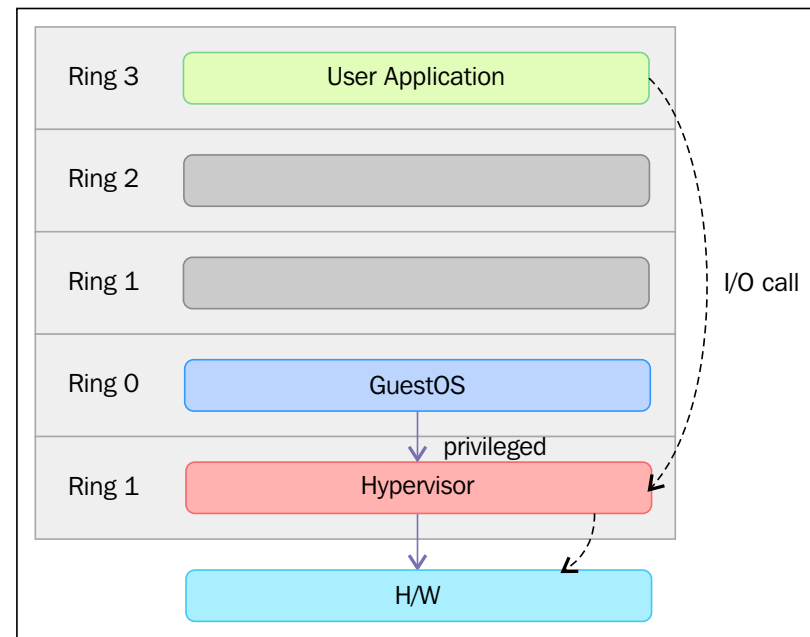
- Método de virtualización diseñado para proveer virtualización eficientemente en hardware
- Aparece debido a los problemas de desempeño y la complejidad de las anteriores tipos de virtualización
- Intel
 - Virtualization Technology (VT)
 - Extensiones de Virtualización: VT-x
- AMD
 - Secure Virtual Machine (SVM)
 - Extensiones de Virtualización: AMD-V
- Agrega un nuevo nivel de seguridad: ring - 1: El hipervisor corre en ring -1 y el guest OS en ring 0



Fuente: Mastering KVM Virtualization

Virtualización Asistida por Hardware (virtualización nativa)

- VENTAJA
 - Hipervisor mas liviano → necesita realizar menos trabajo en comparación con otros tipos de virtualización, por lo tanto el desempeño se ve menos afectado
 - Solución menos compleja
- Usado por soluciones de virtualización modernas, ejemplo: VMWare, Xen



Fuente: Mastering KVM Virtualization

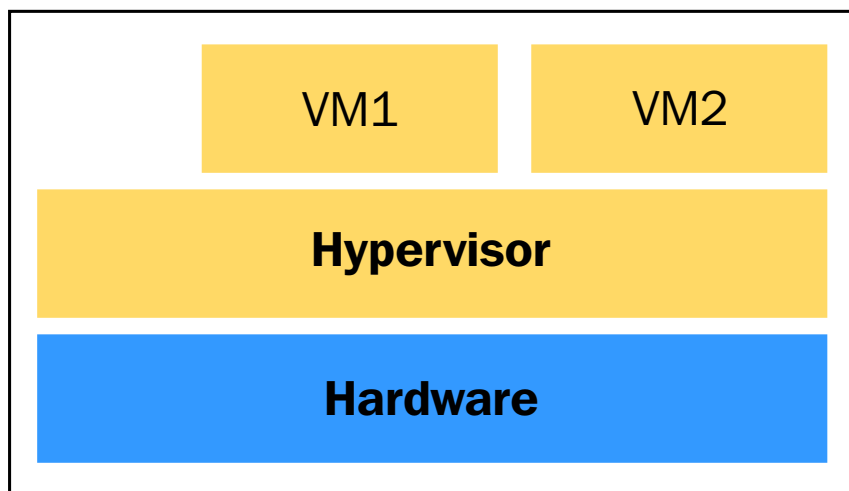
El Monitor de Maquinas Virtuales (VMM/Hypervisor)

VMM / Hypervisor

- Pieza de software responsable del monitoreo y control de maquinas virtuales
- Responsable de:
 - Manejo del ciclo de vida de las VMs
 - Migración de VMs
 - Asignación de recursos en tiempo real
- Soporta múltiples VMs con diferentes sistemas operativos

Hipervisores Tipo 1 (Nativos)

- Corre directamente sobre el hardware
- No necesita un sistema operativo host
- Ejemplo: Virt-node



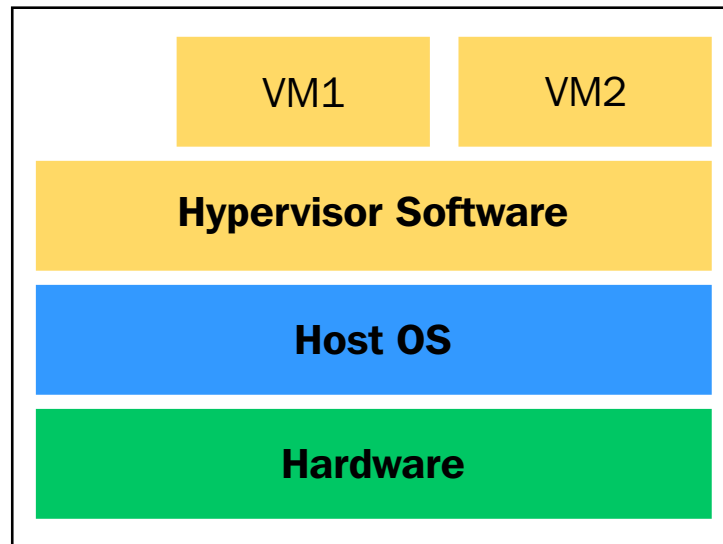
Fuente: Mastering KVM Virtualization

Hipervisores Tipo 1 (Nativos)

- Ventajas
 - Fácil de instalar y configurar
 - Optimizados para sacar máximo provecho al hardware
 - Menos pesados → vienen solo con las aplicaciones necesarias para correr maquinas virtuales
- Desventajas
 - No permiten personalización → No permiten instalar ninguna aplicación de terceros o drivers

Hipervisores Tipo 2 (Hosted)

- Residen sobre un sistema operativo (host)
- Dependen del sistema operativo host para su operación



Fuente: Mastering KVM Virtualization

Hipervisores Tipo 2 (Hosted)

- Ventaja
 - Soporte amplio de hardware → El sistema operativo host controla el acceso a hardware

Ejemplos de hipervisores

TYPE 1 HYPERVISOR

VERSUS

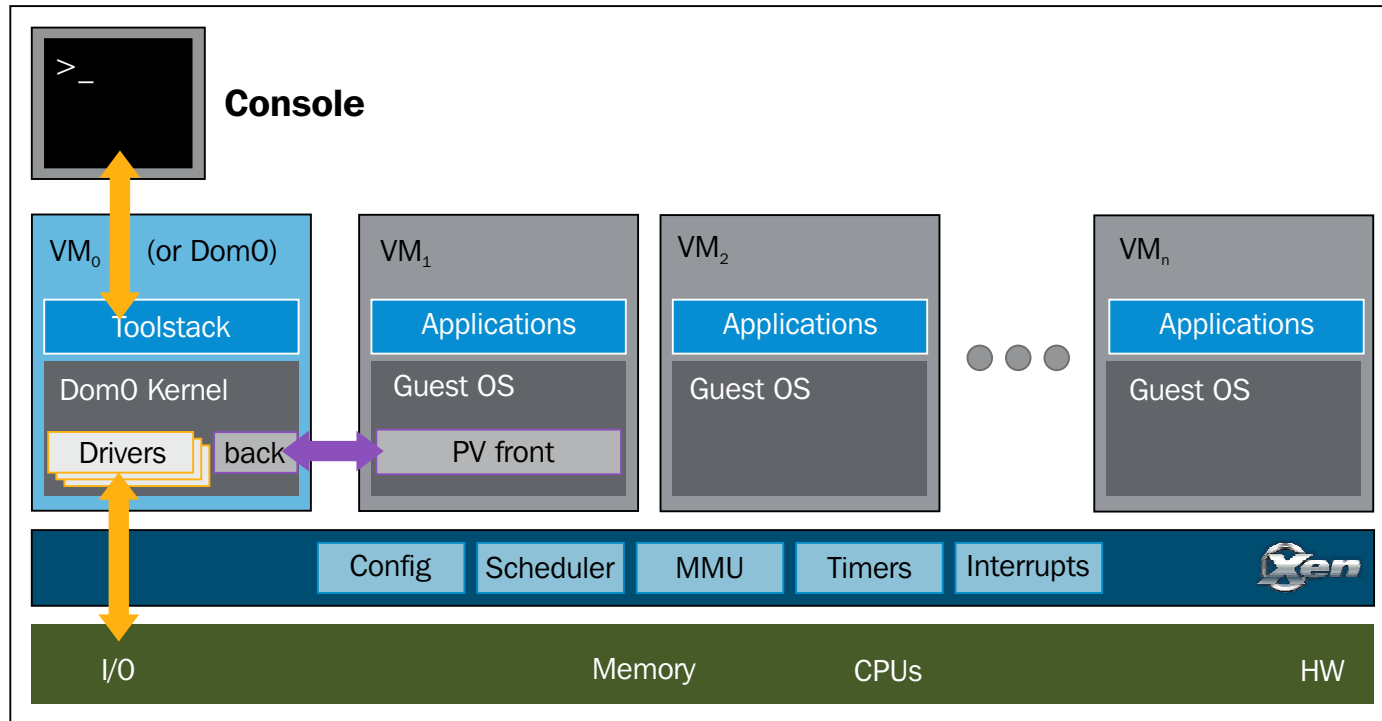
TYPE 2 HYPERVISOR

TYPE 1 HYPERVISOR	TYPE 2 HYPERVISOR
A hypervisor that runs directly on the host's hardware to control the hardware and to manage guest operating systems	A hypervisor that runs on a conventional operating system just as other computer programs do
Called a native or Bare Metal Hypervisor	Called a Host OS Hypervisor
Runs directly on the host's hardware	Runs on an operating system similar to other computer programs
Examples: AntsleOs, Xen, XCP-ng, Microsoft Hyper V, VMware ESX/ESXi, Oracle VM Server for x86	Examples: VMware Workstation, VMware Player, VirtualBox, Parallel Desktop for Mac Visit www.PEDIAA.com

Xen Hypervisor

- Originado en University of Cambridge (2003)
- Soporta virtualización asistida por hardware y Paravirtualization
- Corre sistemas operativos guests llamados **Domains**, estos pueden ser
 - **Dom 0**: dominio privilegiado, contiene los drivers para controlar todos los dispositivos hardware
 - **Dom U**: guest sin privilegios controlados por Dom 0 → toda la comunicación con hardware se hace a través de Dom 0

Xen Hypervisor

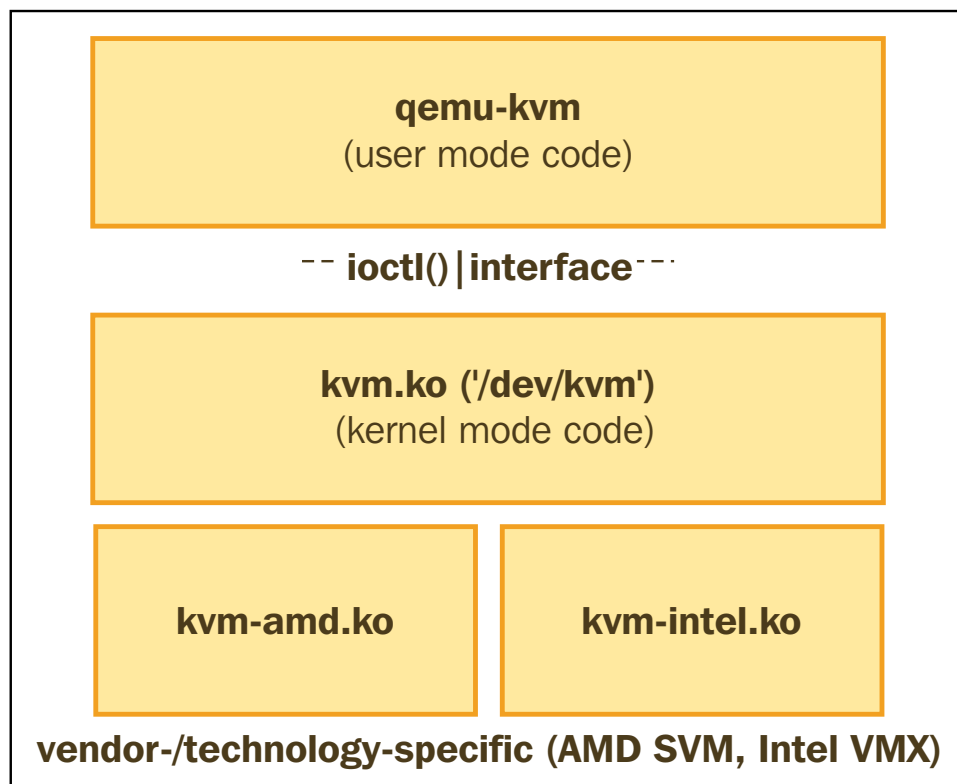


Source: <http://www.xenproject.org/>

Kernel-based Virtual Machine (KVM)

- Solución Full Virtualization
- Se apoya en hardware con soporte para virtualización VT-x, AMD-V
- Convierte el kernel de Linux en un hipervisor cuando se instala el módulo de kernel KVM
- Usa QEMU para emular el procesador, disco, red, VGA, puertos y demás hardware

Kernel-based Virtual Machine (KVM)



Fuente: Mastering KVM Virtualization

Resumen de Técnicas de Virtualización

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

Referencias

- Chirammal, Humble Devassy, Prasad Mukhedkar, and Anil Vettathu. Mastering KVM Virtualization. Packt Publishing Ltd, 2016.
- A Cure for Virtual Insanity: A Vendor-Neutral Introduction to Virtualization Without the Hype. <https://oracle-base.com/articles/vm/a-cure-for-virtual-insanity>
- VMWARE. Understanding Full Virtualization, Paravirtualization and Hardware Assist. <http://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualization-1008.html>
- VMWARE. Virtualization Overview. <https://www.vmware.com/pdf/virtualization.pdf>
- Linux Containers: <https://www.ubuntu.com/cloud/lxd>
 - <https://linuxcontainers.org/lxd/introduction/>