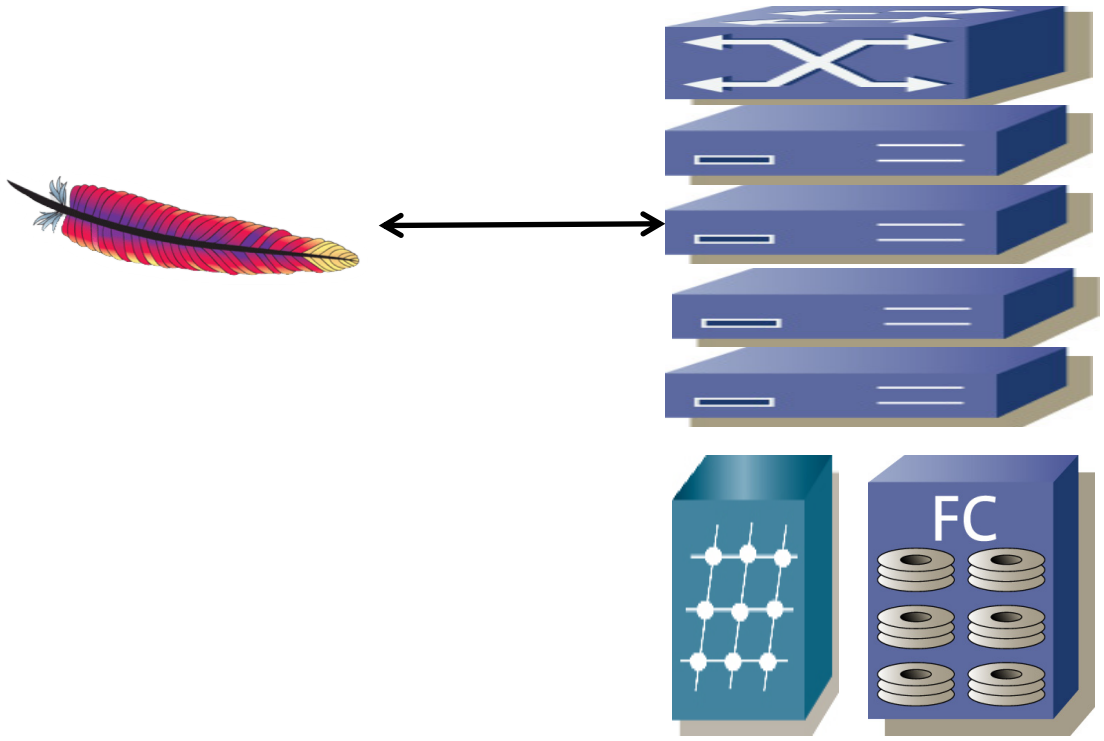
Virtualización

Application Silos

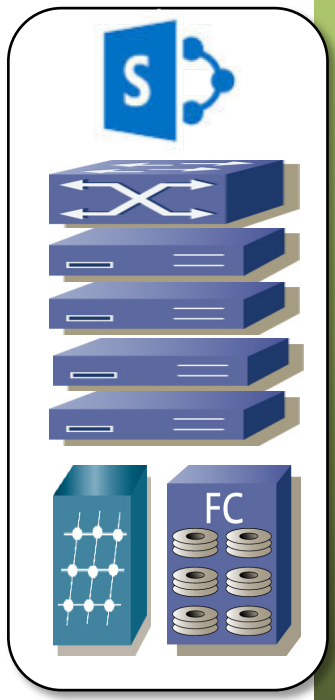
Application Silos

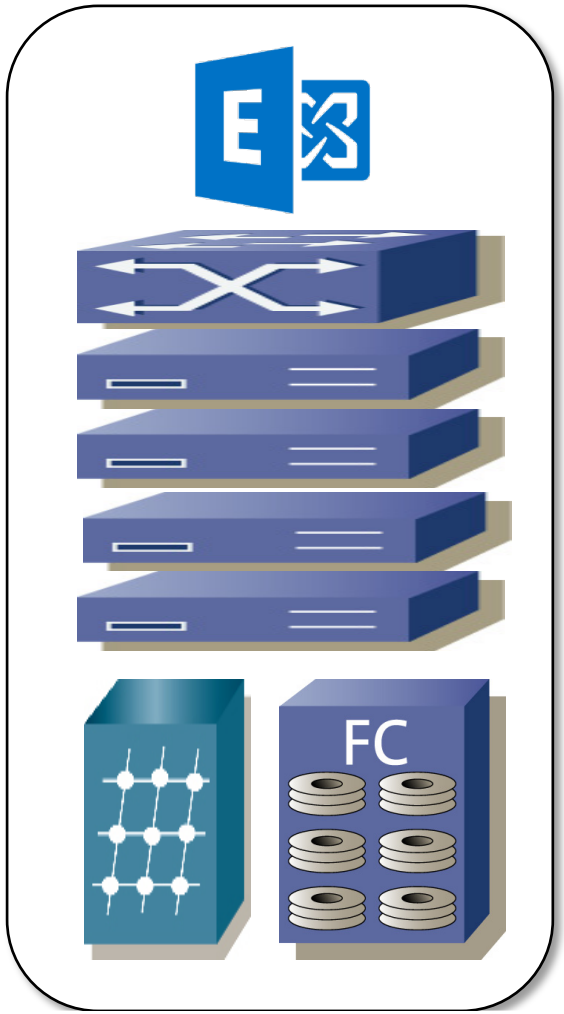
- En el entorno distribuido una nueva aplicación (software servidor) se desplegaba sobre un hardware independiente
- Una relación 1:1 entre la aplicación y el hardware servidor
- O como mucho 1:N porque tengamos múltiples servidores
- A esto habría que añadirle el almacenamiento
- Y la electrónica de red

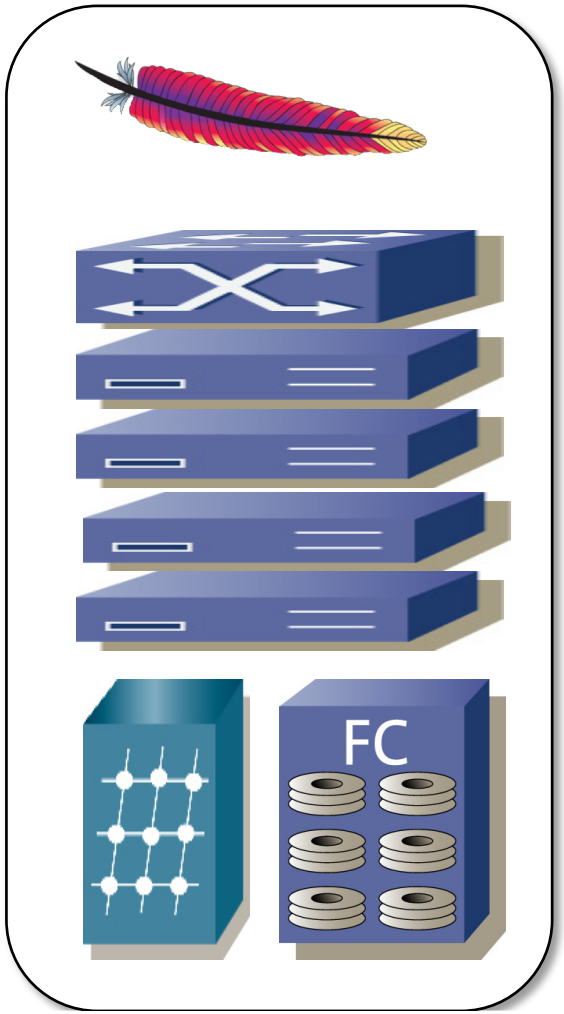


Application Silos

- ¿Y si tenemos otra aplicación?
- Cada una sus servidores y almacenamiento
- El hardware no es reutilizable por otras aplicaciones

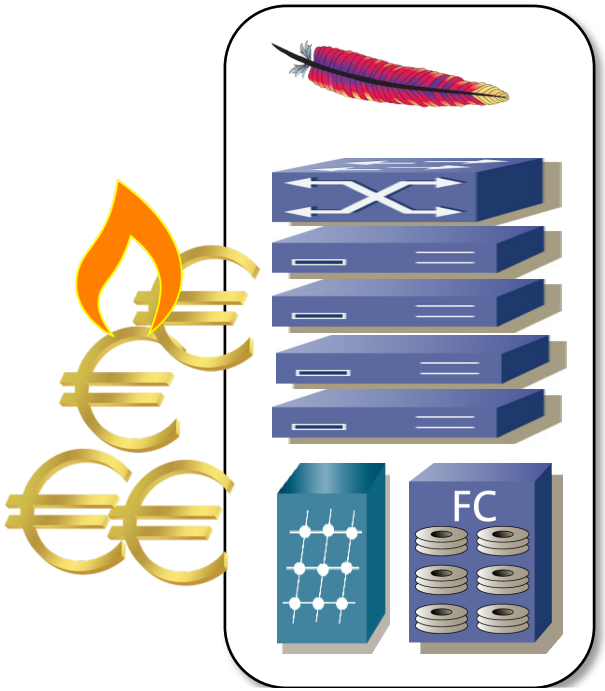






Application Silos

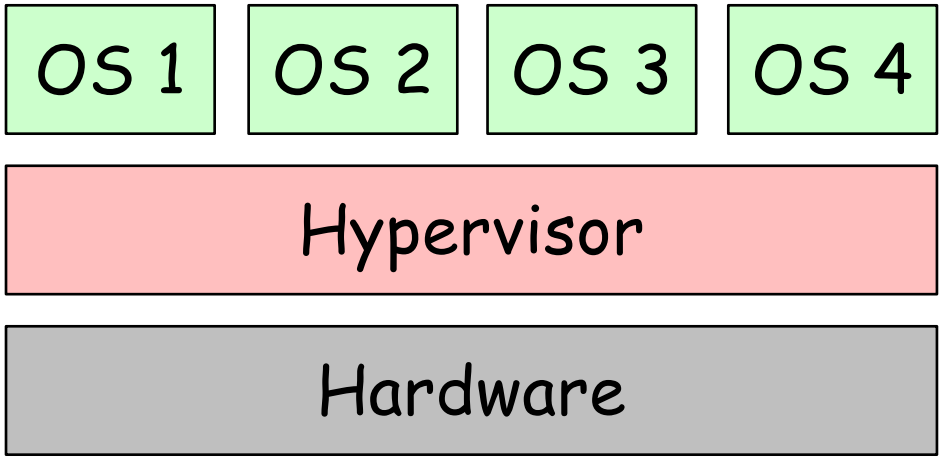
- La utilización (CPU) de los servidores es muy baja
- Esto es así para soportar incrementos de carga
- Si no es baja entonces ante un incremento de carga no es rápido provisionar nuevo hardware
- Lo mismo sucede con la utilización de los discos
- Esto se multiplica por el número de aplicaciones
- Pero ocupan todo el tiempo el espacio
- Y están encendidos, consumiendo potencia
- Y necesitando refrigeración
- Esto ha cambiado con la virtualización
- Consolidación



Virtualización: Ejemplos

¿Virtualización?

- La idea básica de virtualización del host es bastante conocida
- Una capa software intermedia hace creer a un sistema operativo que tiene hardware dedicado
- En realidad esto lo hemos visto antes (…)

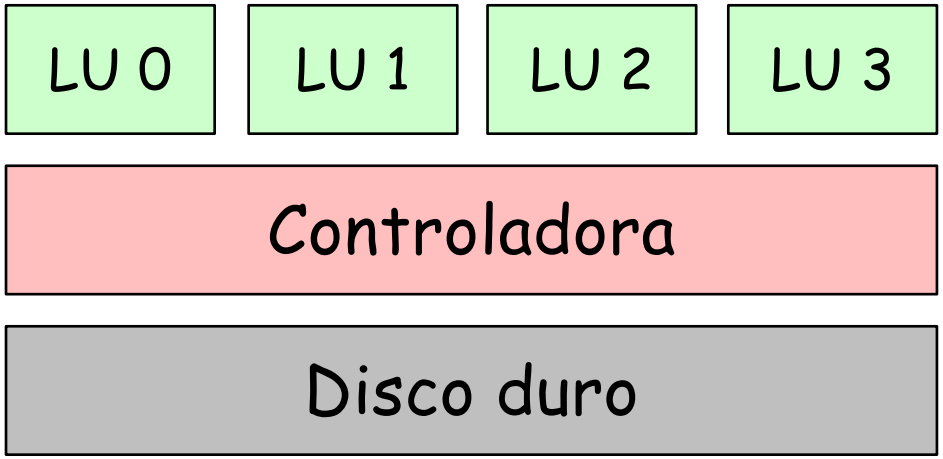


¿Virtualización?

- Es la misma idea detrás de las VLANs
- Los hosts de cada VLAN la ven como si estuvieran ellos solos en la LAN

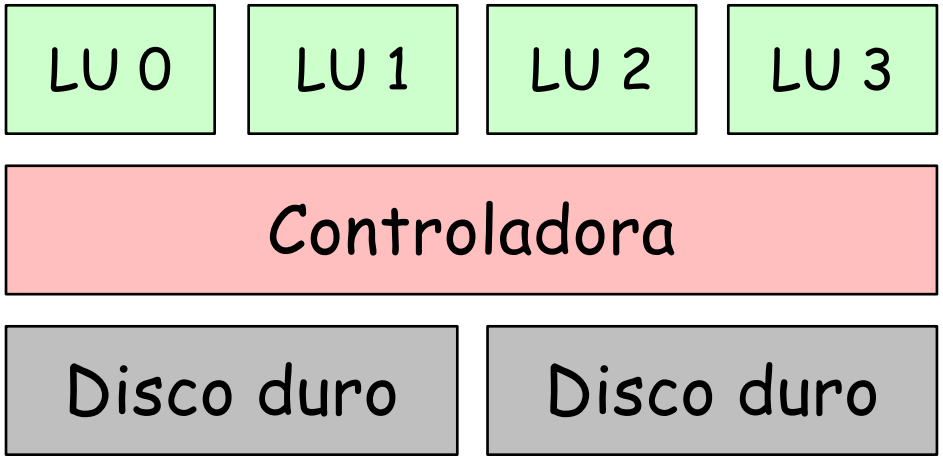
¿Virtualización?

- O al crear unidades lógicas en un disco
- Al crear particiones



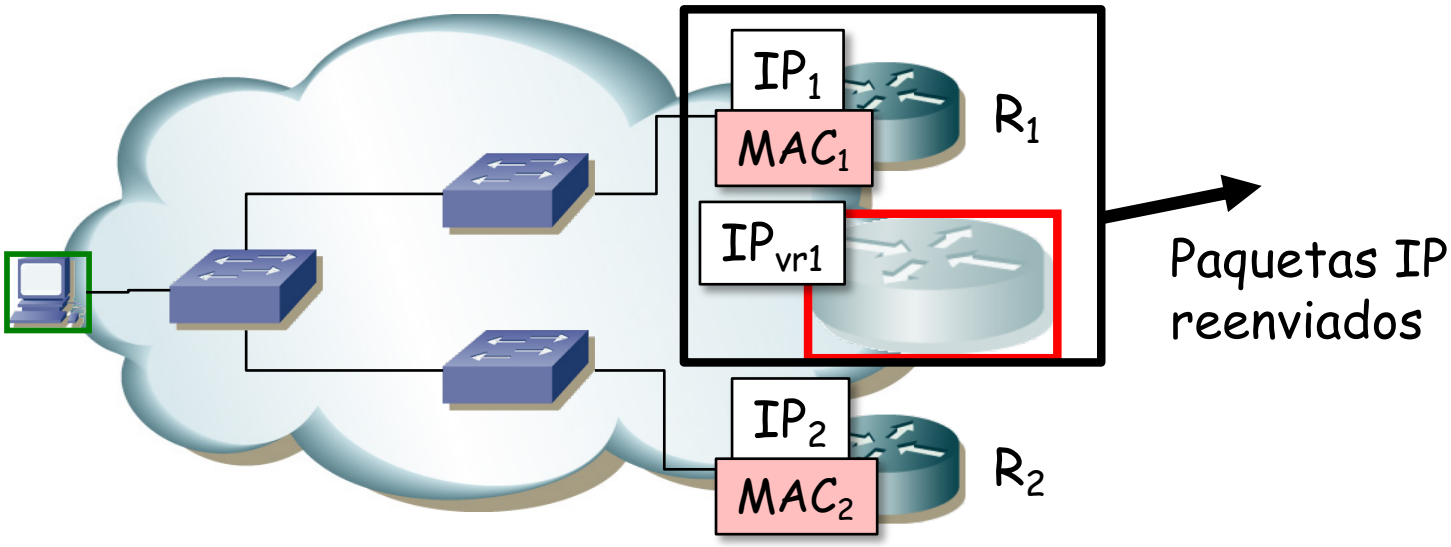
¿Virtualización?

- Y hemos visto que no tiene que ser necesariamente un 1:N (1 hardware compartido entre N)
- Con un RAID creamos una visión virtual de un conjunto de discos como uno solo
- O de nuevo como otro conjunto



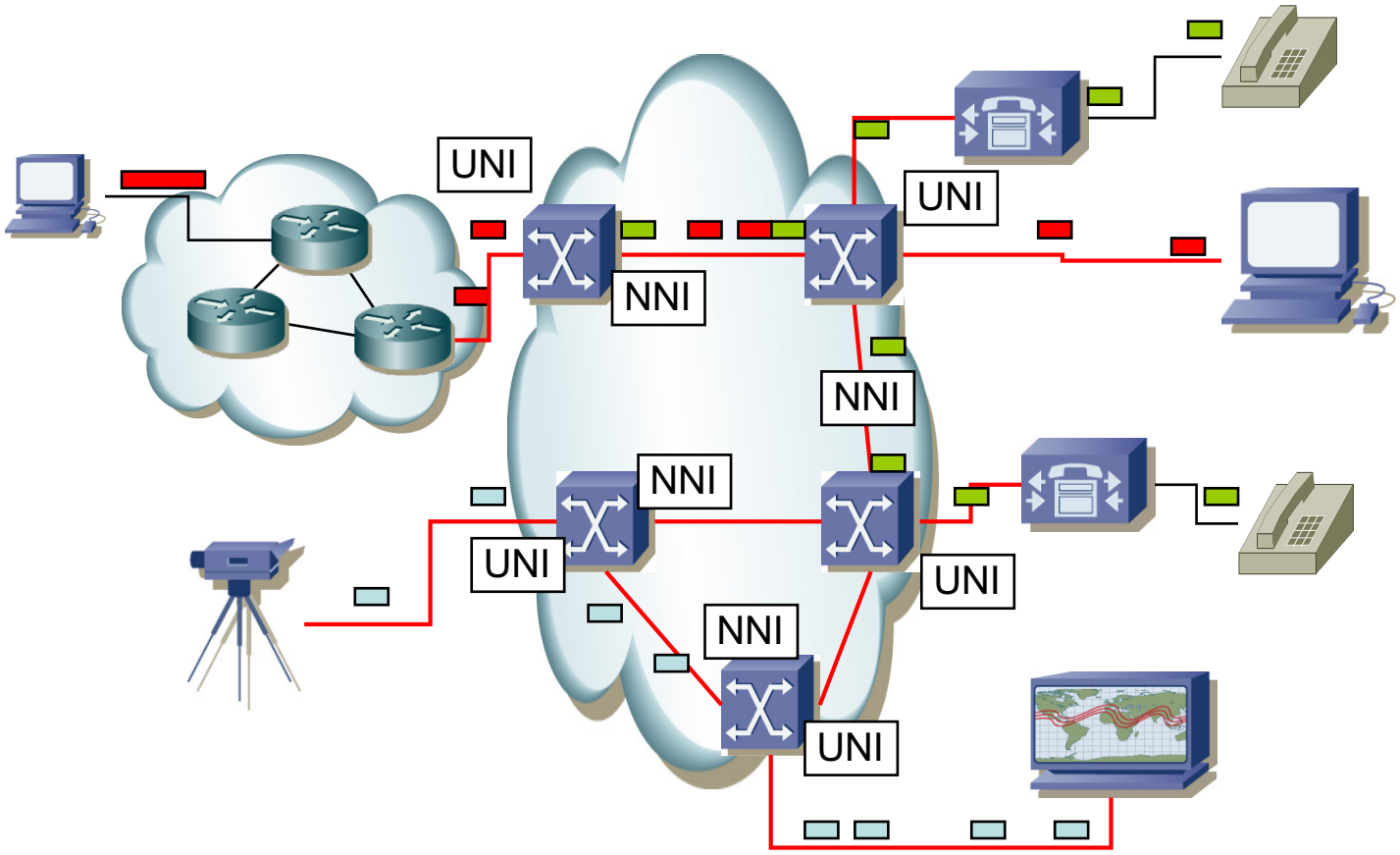
¿Virtualización?

- Cuando hemos descrito un FHRP como por ejemplo VRRP
- Hablábamos de un “router virtual”
- Los hosts ven 2+ routers como uno solo



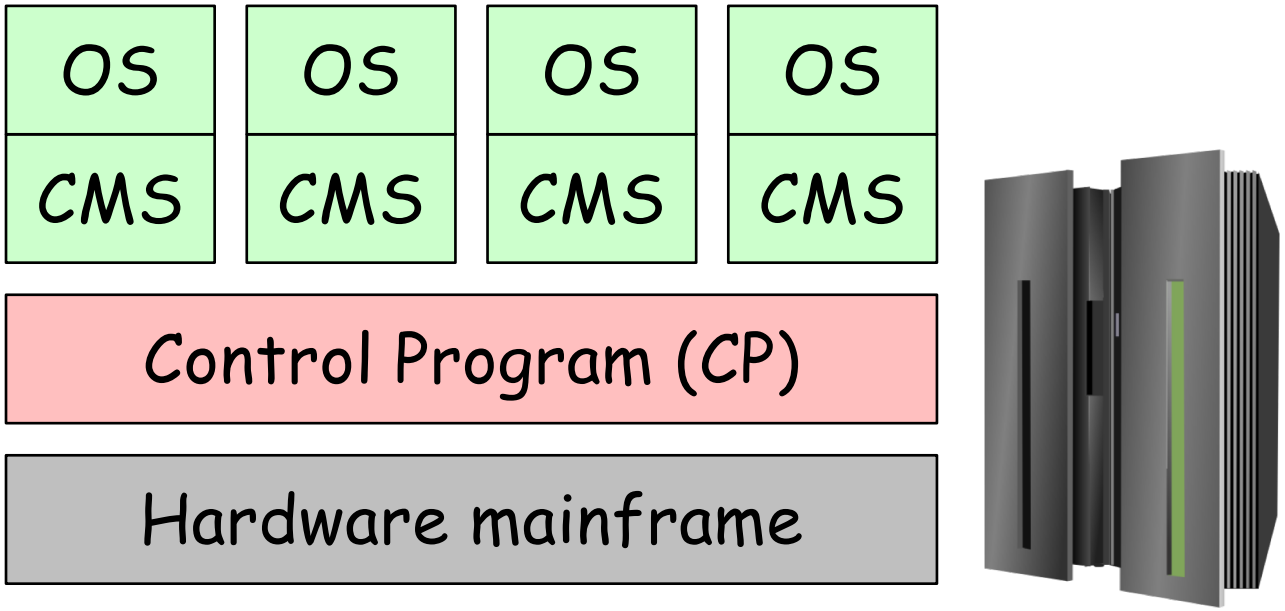
¿Virtualización?

- Hablábamos por ejemplo en ATM de “Circuitos Virtuales”



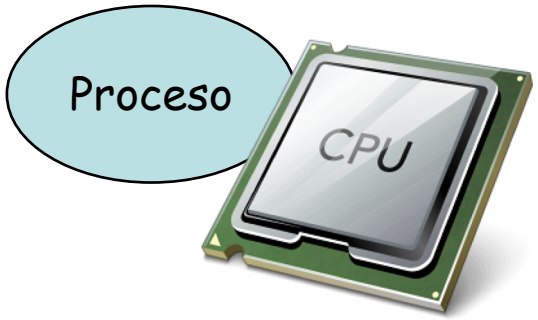
¿Virtualización?

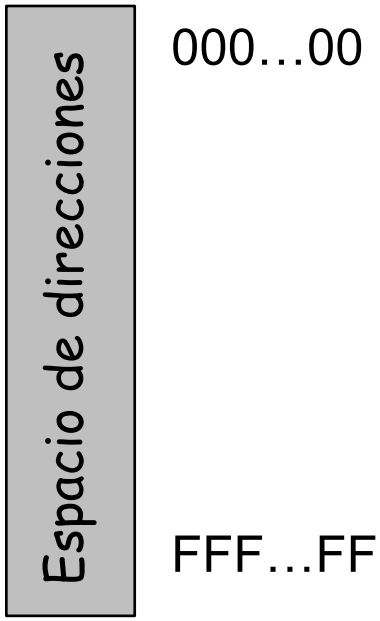
- En el entorno informático es un concepto muy antiguo
- A nivel de virtualización de sistemas operativos ya lo soportaban los mainframes en los 70s
- Comercialmente llega al entorno PC a principios de los 00s (VMware)



¿Virtualización?

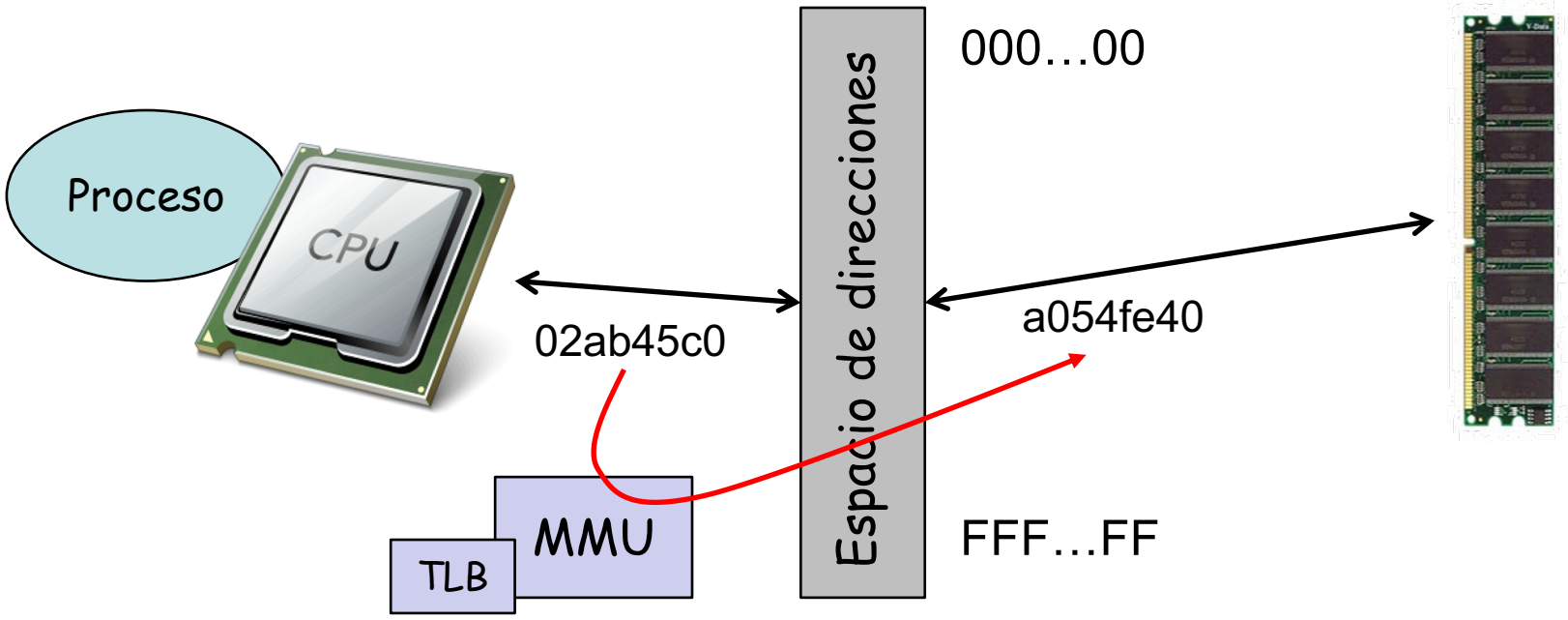
- La memoria virtual es anterior a la virtualización completa del hardware
- Se puede conseguir que un proceso (programa en ejecución) crea que dispone de toda la memoria
- De hecho podría ver más memoria que la existente
- Ve un espacio continuo de direcciones
- (…)





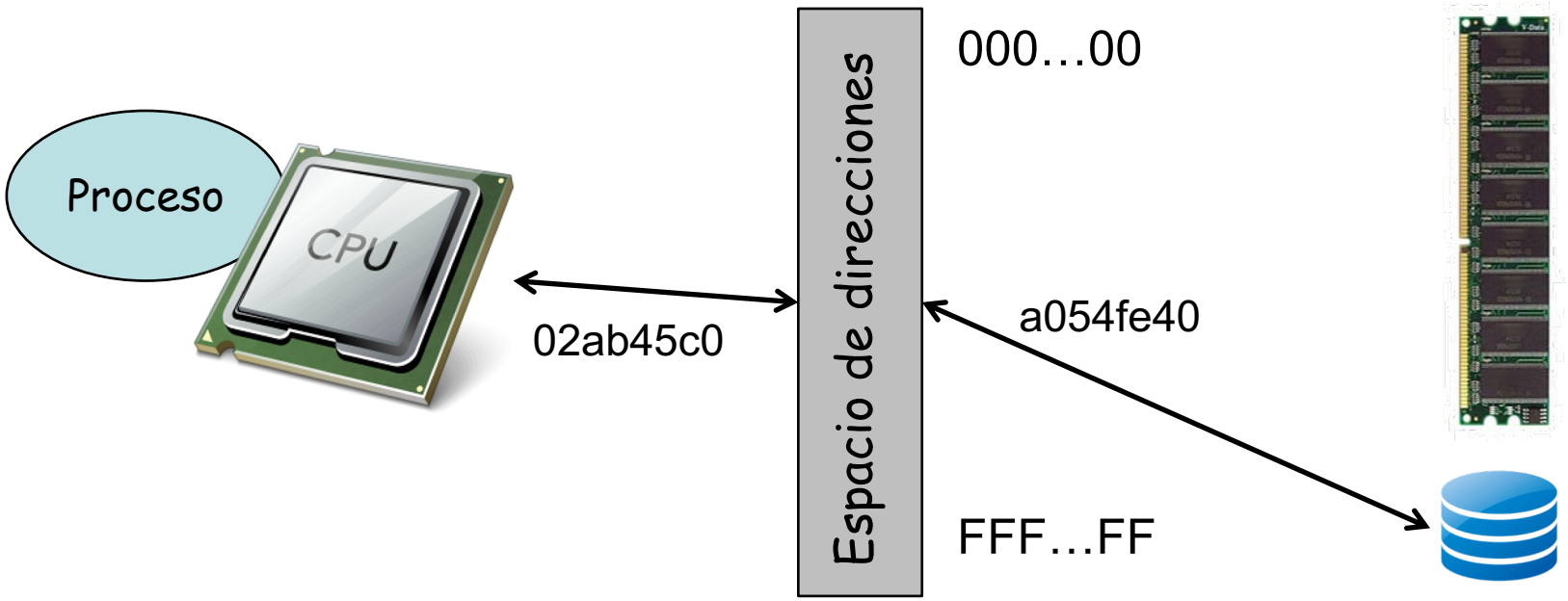
¿Virtualización?

- Cuando la CPU intenta acceder a una dirección de memoria se debe convertir la dirección *virtual* en la dirección física
- Con esa dirección física se puede acceder a la RAM (ignorando las posibles caches)
- Esta conversión la hace la MMU (*Memory Management Unit*)
- Hoy en día es parte de la CPU
- Es decir, necesitamos (o al menos mejora el rendimiento) apoyo del hardware



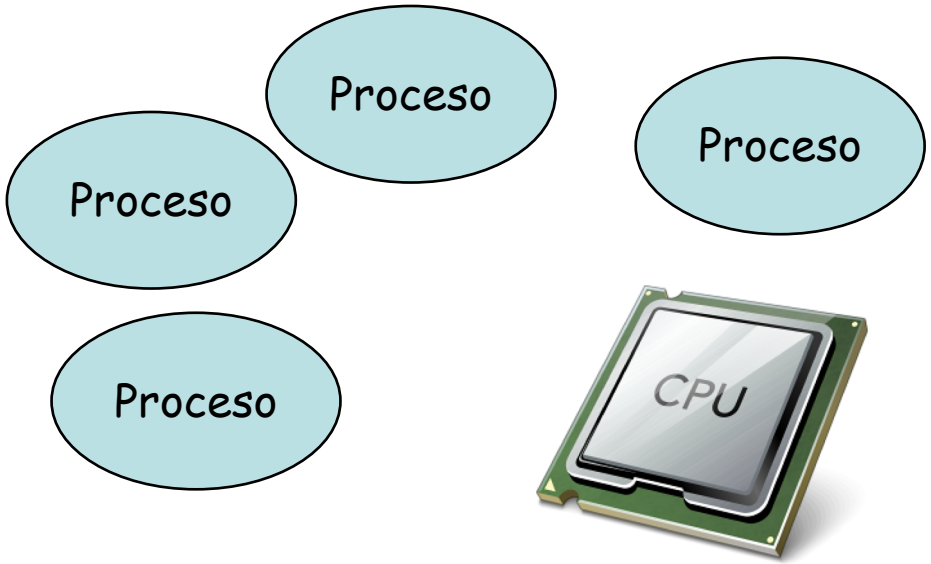
¿Virtualización?

- El mapeo podría no llevar a memoria RAM sino a datos guardados en disco
- El disco es un dispositivo mucho más lento así que lo normal es mover los datos frecuentemente utilizados a RAM y los poco utilizados a disco



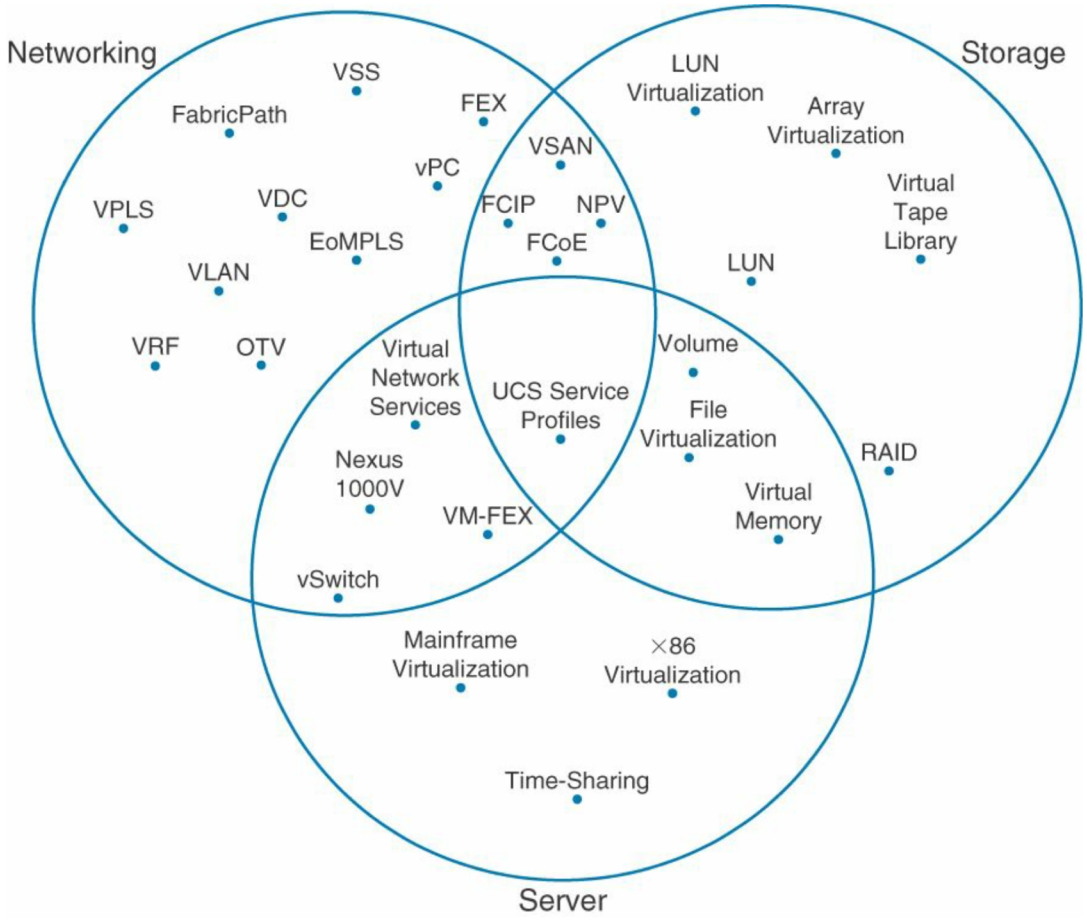
¿Virtualización?

- Cuando varios procesos se ejecutan pero no disponemos de varias CPUs
- Cada proceso cree que dispone de la CPU pero se va alternando la ejecución entre procesos
- De nuevo se le está haciendo creer a alguien que dispone de ciertos recursos de forma exclusiva cuando no es así



Virtualización, ¿dónde?

- Servidor
- Red
- Almacenamiento

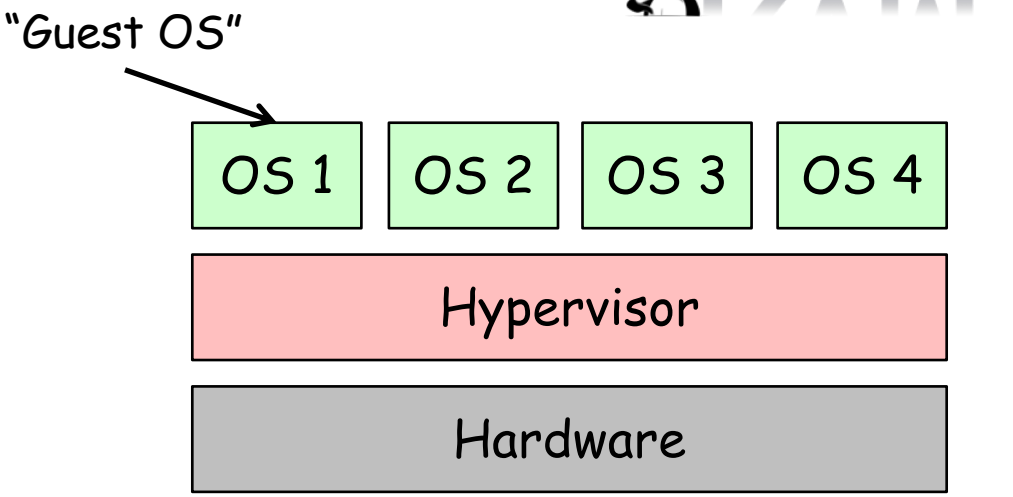


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Virtualización de servidor

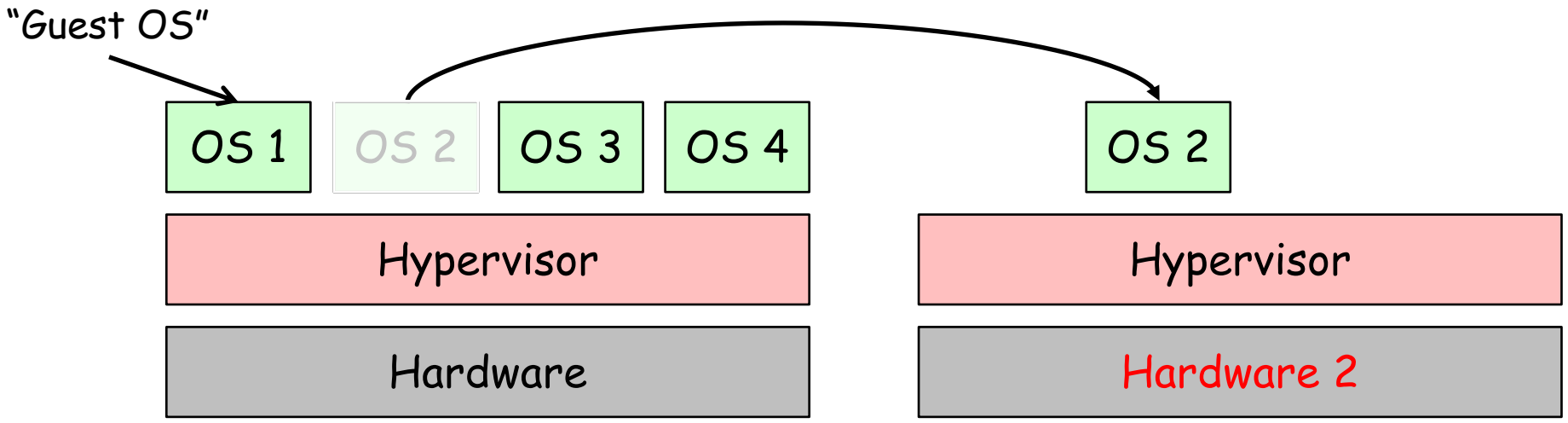
Hypervisor

- Es una capa software entre el hardware y el sistema operativo "guest"
- También llamado "*Virtual Machine Monitor*" (VMM)
- Oculta el hardware real y puede presentar diferente hardware a cada máquina virtual
- Esas máquinas virtuales no necesitan cambios para funcionar en otro hypervisor aunque emplee un hardware diferente siempre que les presente el mismo hardware virtual (…)



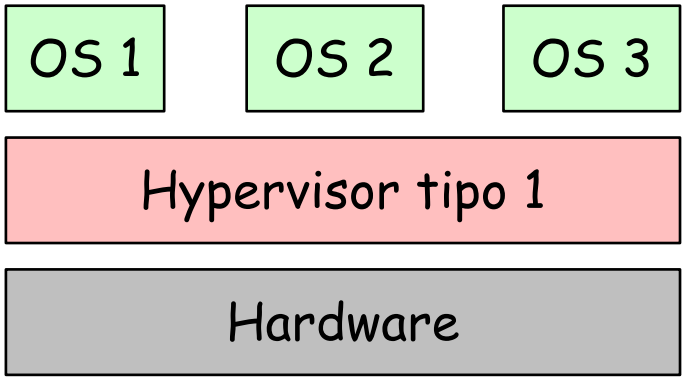
Hypervisor

- Es una capa software entre el hardware y el sistema operativo “guest”
- También llamado “*Virtual Machine Monitor*” (VMM)
- Oculta el hardware real y puede presentar diferente hardware a cada máquina virtual
- Esas máquinas virtuales no necesitan cambios para funcionar en otro hypervisor aunque emplee un hardware diferente siempre que les presente le mismo hardware virtual
- La máquina virtual, todo su sistema operativo instalado y las aplicaciones, puede ser un solo fichero, sencillo de copiar a otra máquina



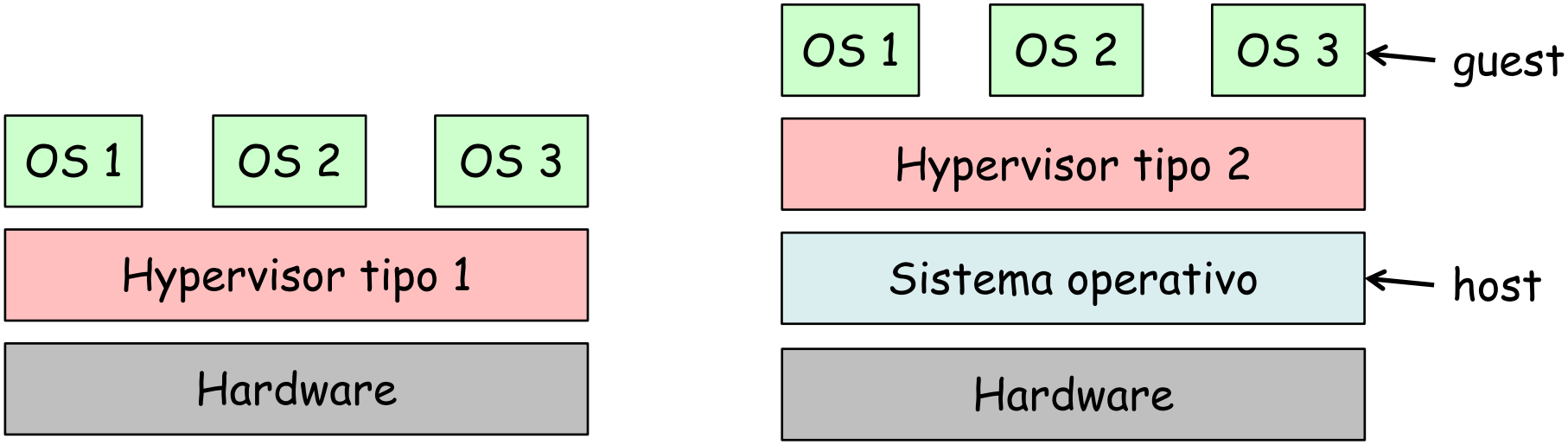
Tipos de Hypervisores

- Tipo 1, nativo o "*bare-metal*"
  - Se ejecuta directamente sobre el hardware
  - Controla dicho hardware
  - Consume poco espacio y memoria
  - El mejor rendimiento potencial
  - El hypervisor debe contar con drivers para el hardware
  - Ejemplos: Citrix XenServer, Vmware ESXi, Microsoft Hyper-V, Linux KVM
- (…)



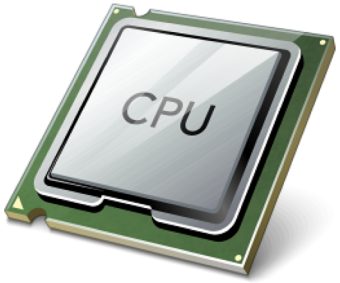
Tipos de Hypervisores

- Tipo 2 o “hosted”
  - El hypervisor corre como una aplicación sobre un sistema operativo convencional
  - El sistema operativo guest sobre el hypervisor
  - El sistema operativo host tiene un impacto en el rendimiento
  - Es más frecuente la existencia de drivers para el hardware
  - Ejemplos: VMware Workstation, VMware Server, Microsoft Virtual PC, Parallels Workstation, VirtualBox, QEMU



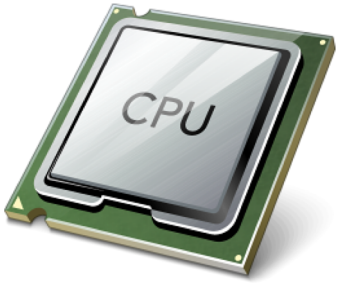
Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
  - Hace traducción (*on-the-fly*) de instrucciones (*binary translation*)
  - Se sustituyen las instrucciones no virtualizables por otras equivalentes
  - No requiere modificar el OS instalado
  - Ejemplos: VMware, Microsoft Virtual Server, Linux KVM, Parallels, VirtualBox, QEMU
- (…)



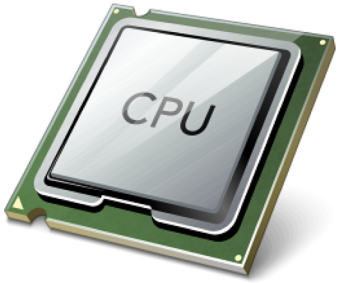
Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
- *Paravirtualization (OS assisted virtualization)*
  - Se modifica el sistema operativo guest sustituyendo las instrucciones no virtualizables
  - Requiere menos sobrecarga en ejecución pero hay que poder modificar el código de ese sistema operativo guest
  - Ejemplos: Xen, VMware (VMTools), Virtualbox (additions), UML
- *(…)*



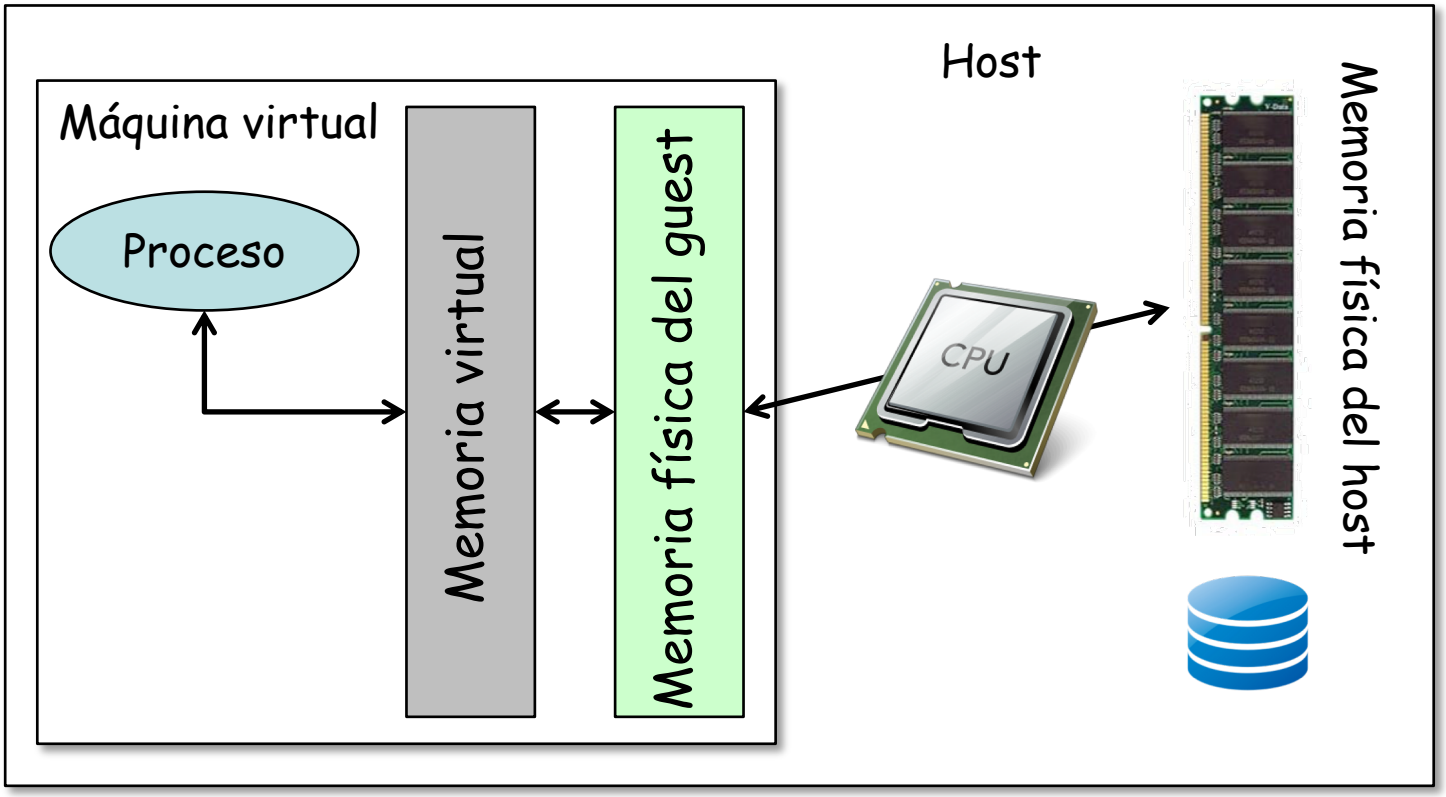
Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
- *Paravirtualization (OS assisted virtualization)*
- *Hardware-assisted virtualization*
  - El hardware se encarga de la traducción de instrucciones privilegiadas
  - Requiere soporte por el hardware (Intel VT-x, AMD-V)
  - Ejemplos: VMware, Microsoft, Parallels, Xen, Virtualbox



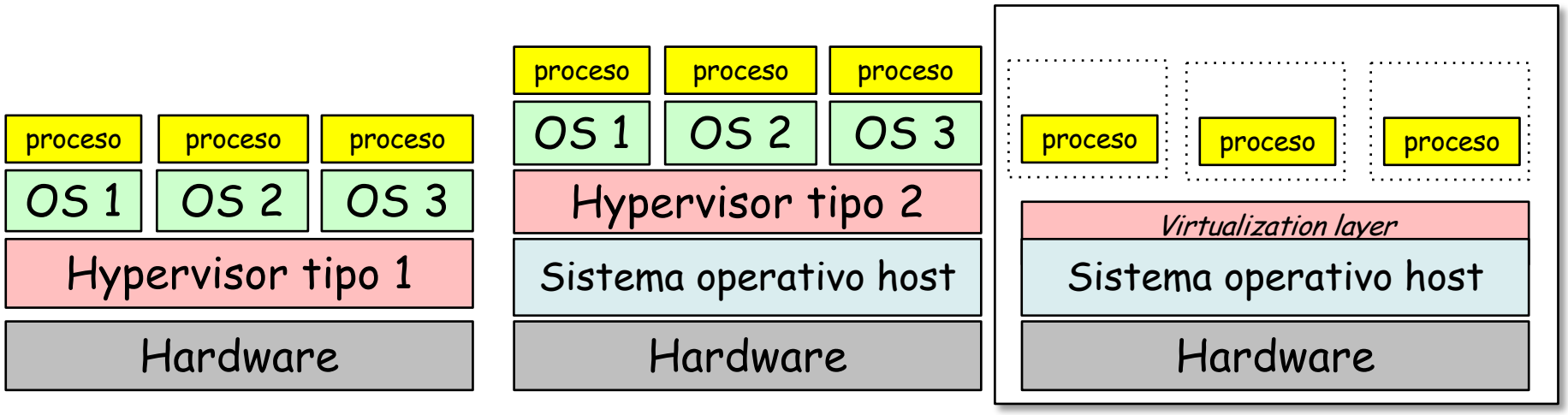
Virtualización de RAM

- El sistema operativo guest emplea memoria virtual y la mapea a lo que él cree que es memoria física
- Eso no puede ser la auténtica memoria física, así que debe ser de nuevo mapeada
- *Shadow page tables* para hacerlo por soft o *nested paging* (Second Level Address Translation) por hardware si lo soporta la CPU
- Hay que virtualizar la MMU



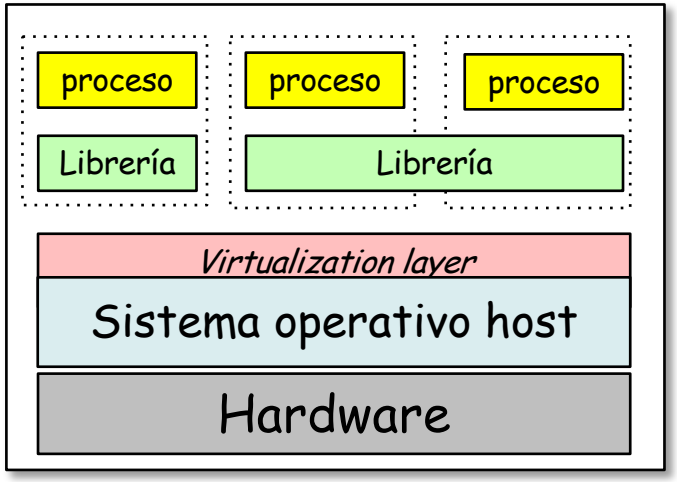
Operating-system-level virt.

- *Containers, Virtualization Engines, Virtual Private Servers, Jails*
- Ejemplos: Docker, rkt, LXC, OpenVZ, Parallels Virtuozzo Containers, Linux-VServer, Solaris Zones, FreeBSD Jails, etc
- Principalmente en entornos UNIX
- El kernel del sistema operativo aísla los procesos como si corrieran en máquinas independientes
- Su implementación más básica en Unix es el comando *chroot* pero solo aísla el sistema de ficheros
- *chroot* permite que a partir de su ejecución un proceso crea que /path/que/quieras es directamente /



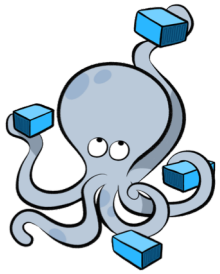
Operating-system-level virt.

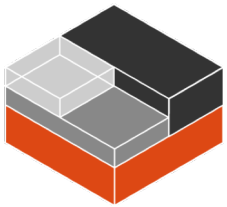
- *FreeBSD Jails* ofrece poco más que chroot (y de aquí viene el término *jailbreaking*)
- El kernel de Linux soporta ya namespaces
- Permite que un proceso o grupo de procesos vea un recursos como suyo enteramente cuando tal vez es compartido
- Control Groups (cgroups): permite controlar el uso de recursos como CPU, RAM, acceso a disco, a la red, etc
- Network namespaces: limitar la visibilidad de recursos de red
- El resultado es un solo kernel y procesos aislados
- Pueden compartir también librerías

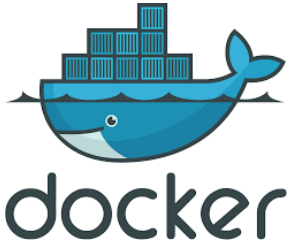


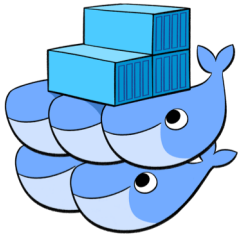
Operating-system-level virt.

- Suele haber utilidades para gestionar esos contenedores
- Un caso muy extendido es Docker (desde 2013)
  - Soporta Windows en Windows 10 Pro y Windows Server 2016 a través de Hyper-V
  - Soporta Mac OS X a partir de El Capitan (macOS 10.11) a través del Hypervisor.famework
  - Open Source su runtime (runC) y formato de contenedor a OCI
- Desde finales de 2015 OCI (Open Container Initiative), proyecto de la Linux Foundation
- Otra alternativa es rkt (CoreOS) y LXC/LXD
- Gran cantidad de proyecto software relacionados con la gestión de contenedores



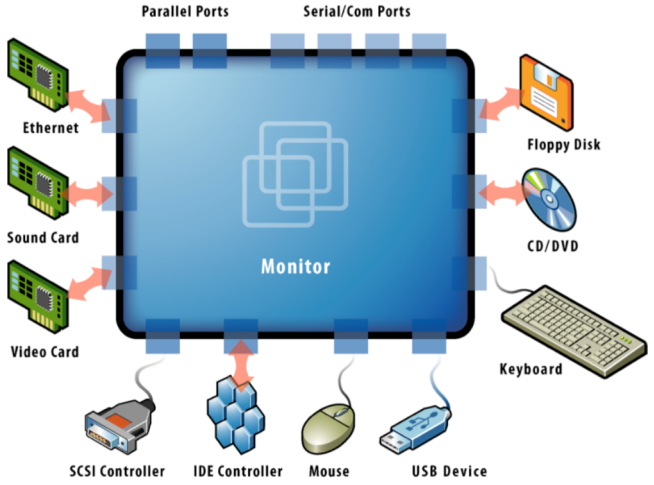






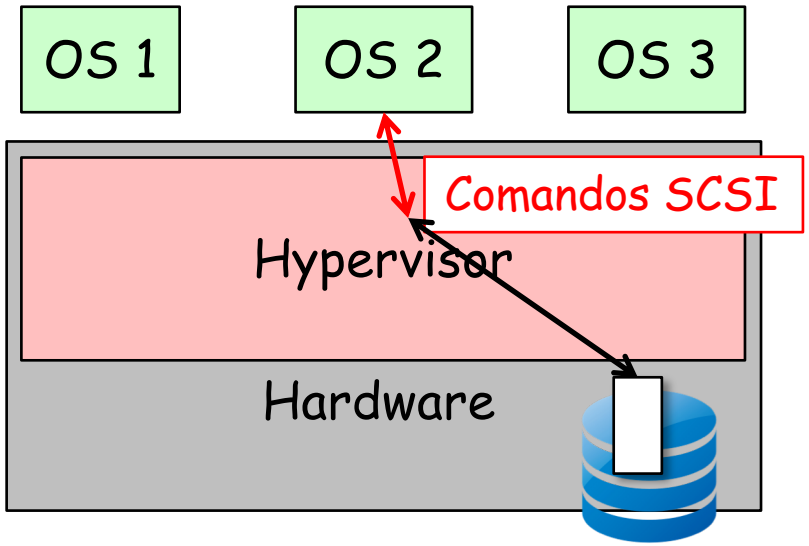
Virtualización de dispositivos

- El VMM presenta a la VM unos dispositivos comunes, de forma que sean fácilmente soportados
- Puede tener varias opciones, por ejemplo ofrecerle al guest diferentes modelos de tarjeta de red
- El hardware puede tener soporte para ser virtualizado



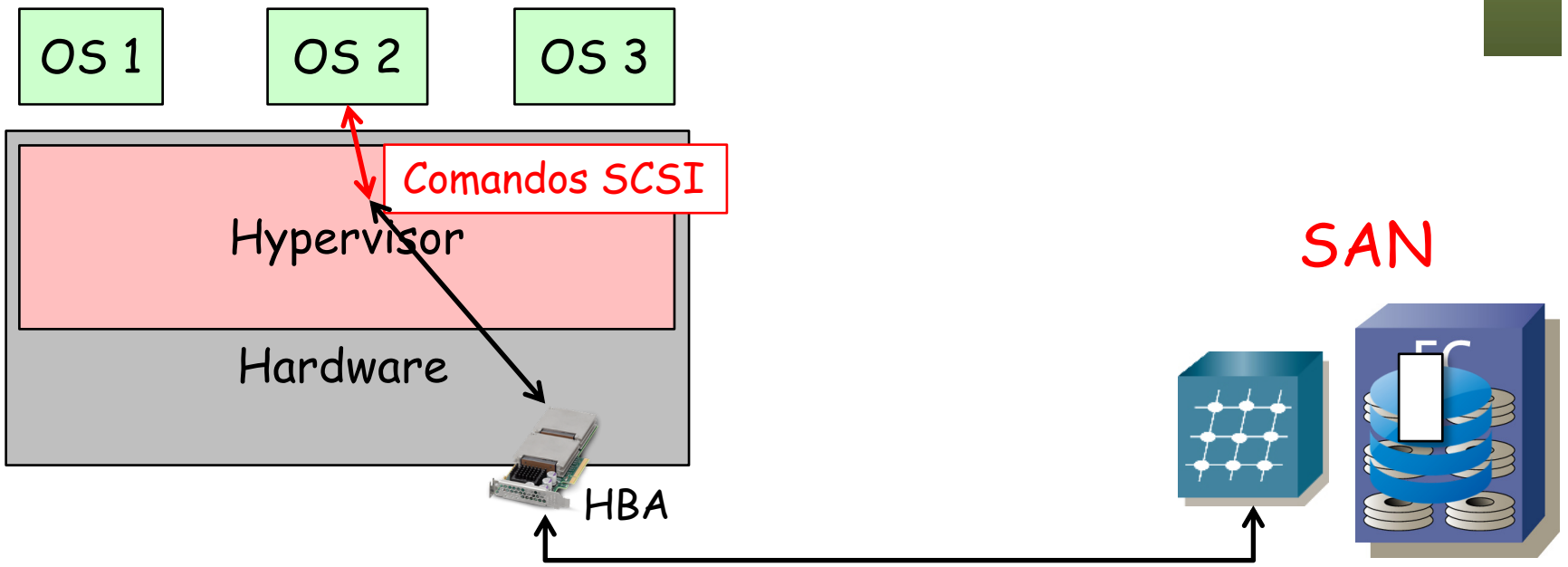
Acceso a disco desde la VM

- En el caso del almacenamiento lo más común es presentarle al guest dispositivos virtuales que responden a comandos SCSI
- De la máquina virtual se reciben comandos SCSI, que se responden obteniendo los datos del sistema de ficheros virtual
- El sistema de ficheros virtual puede almacenarse en un fichero
- Ese fichero puede estar en un disco local (SCSI o no)
- (…)



Acceso a disco desde la VM

- O puede estar en una SAN, por ejemplo mediante un HBA Fibre Channel
- Es decir, el sistema operativo entero (todo su sistema de ficheros) estaría en la SAN
- El HBA también se puede virtualizar y ofrecer un HBA virtual a la VM
- También podría estar en un NAS

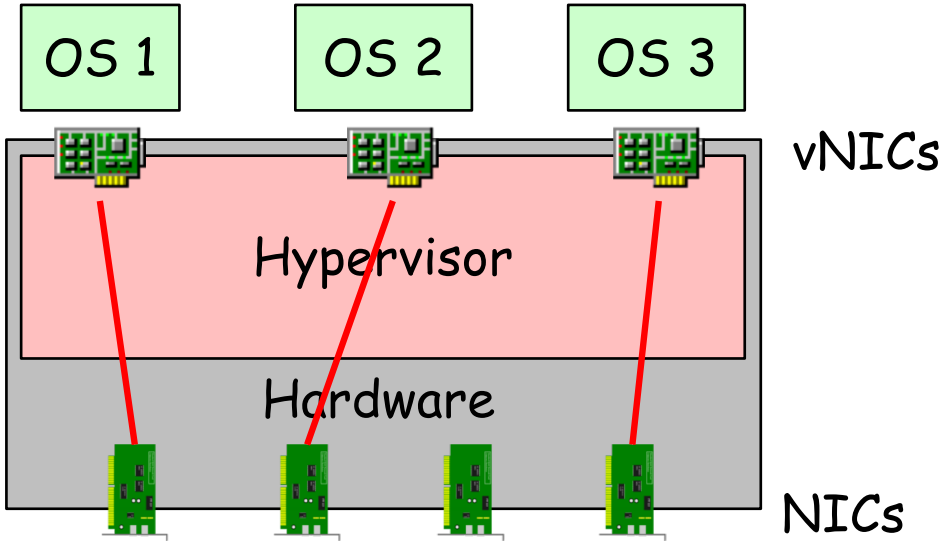


upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Virtualización de red en el servidor

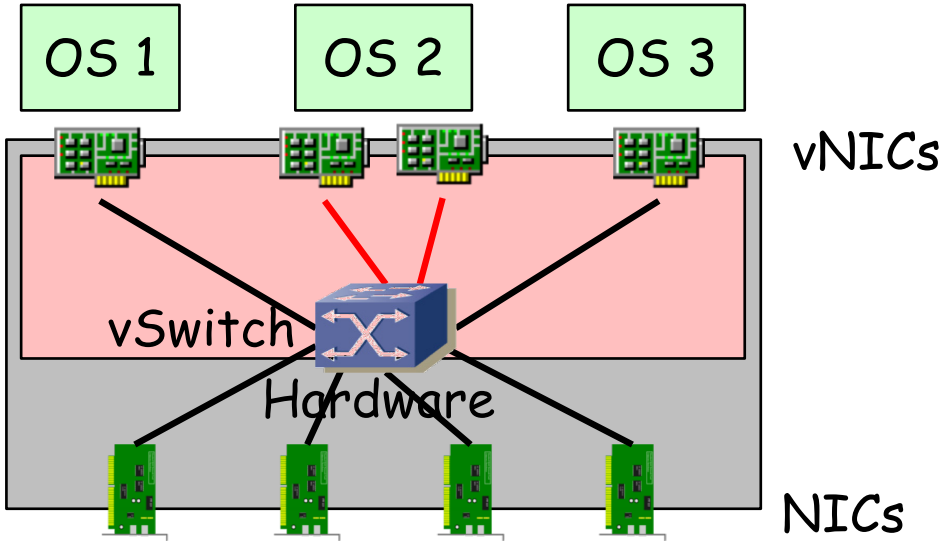
Virtual NIC

- Las NICs reales pueden ser de diferentes modelos que las virtuales
- Puede haber una relación 1:1 entre NIC y vNIC
- (…)



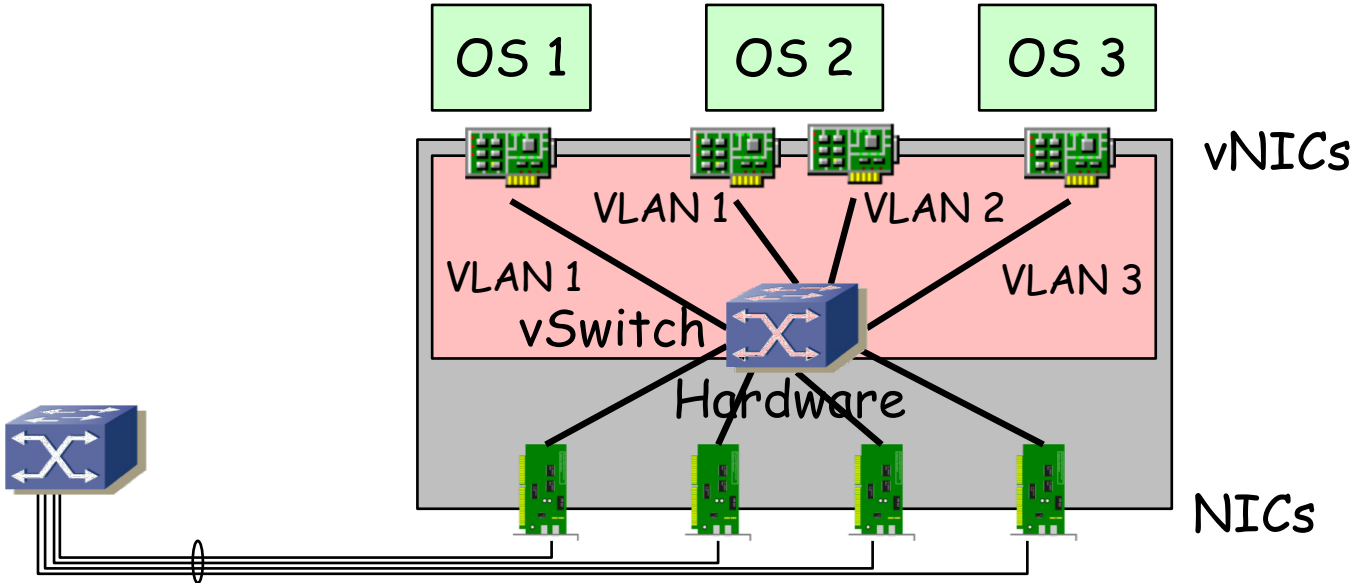
Virtual NIC

- Las NICs reales pueden ser de diferentes modelos que las virtuales
- Puede haber una relación 1:1 entre NIC y vNIC
- Puede implementarse un conmutador Ethernet en software
- Se suele llamar un vSwitch o VEB (Virtual Ethernet/Embedded Bridge)
- La dirección MAC de la vNIC suele ser diferente de la MAC de la NIC
- OUI reservado para la empresa desarrolladora del hypervisor
- Puede haber varias vNICs en la misma VM

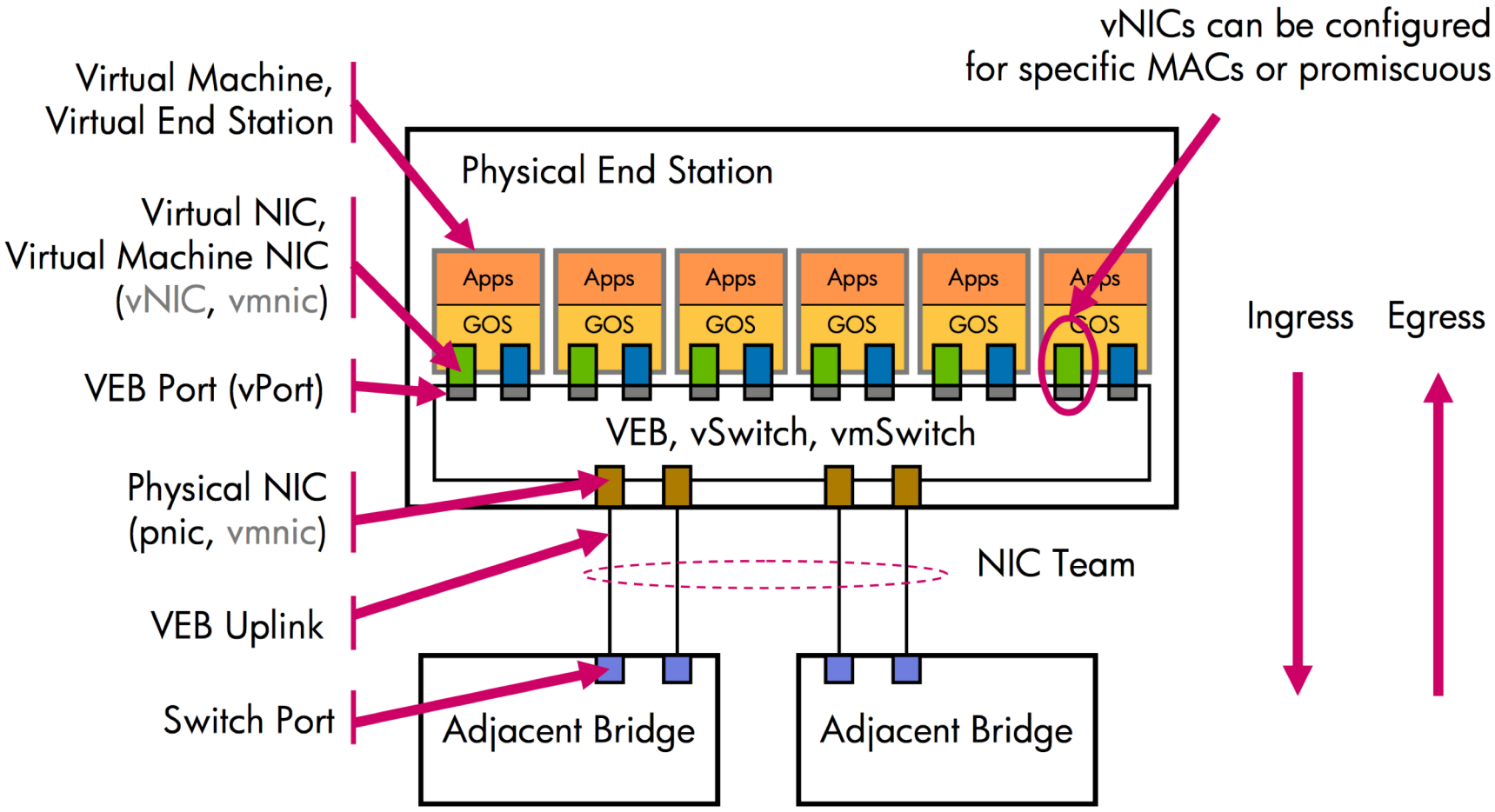


Virtual Switch

- Se pueden asignar los puertos a VLANs diferentes
- Las NICs soportan 802.1Q
- Y agregación (802.3ad) o *NIC teaming*
- El vSwitch tiene más información sobre los hosts que la que puede tener un puente hardware (sabe sus MACs sin usar aprendizaje)
- Puede estar implementado enteramente en software o parte en hardware (normalmente funcionalidades en la NIC)
- Puede estar desarrollado junto con el hypervisor o por otra empresa y así gestionarse como parte del entorno de virtualización o de red



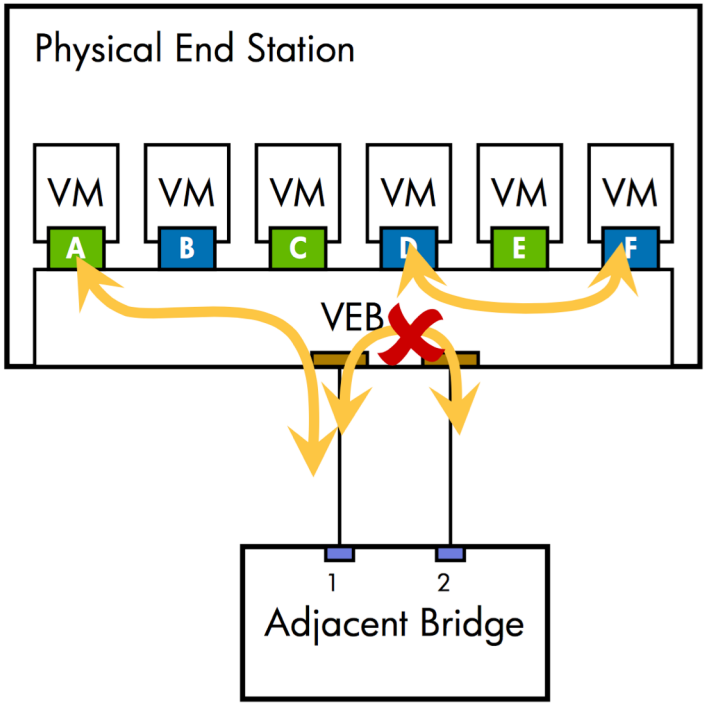
Virtual Switch



http://www.ieee802.org/1/files/public/docs2009/new-hudson-vepa_seminar-20090514d.pdf

Virtual Switch

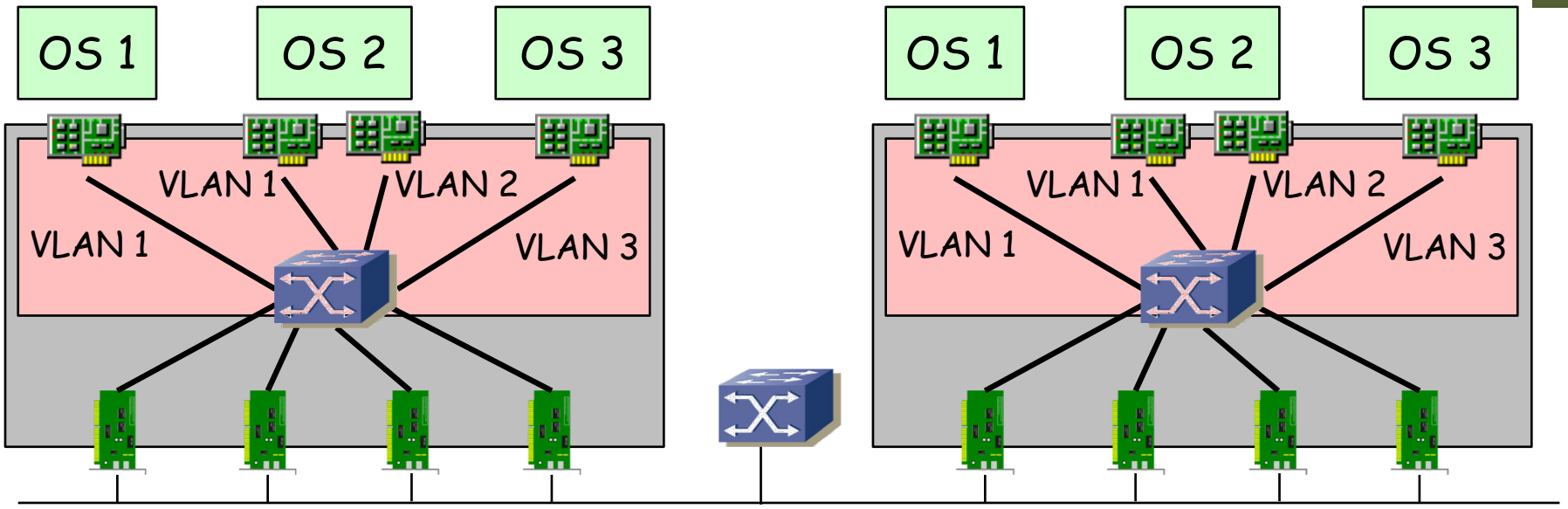
- No reenvía entre los puertos hacia la infraestructura de red
- No participa en el STP
- No necesita hacer aprendizaje, solo tiene las MACs de las VMs estáticamente y el resto debe estar en el exterior
- Pero hay que configurar políticas en sus puertos lógicos
- Probablemente no tenga las funcionalidades de un switch físico (QoS, ACLs, etc)
- ¿De quién es la gestión?



http://www.ieee802.org/1/files/public/docs2009/new-hudson-vepa_seminar-20090514d.pdf

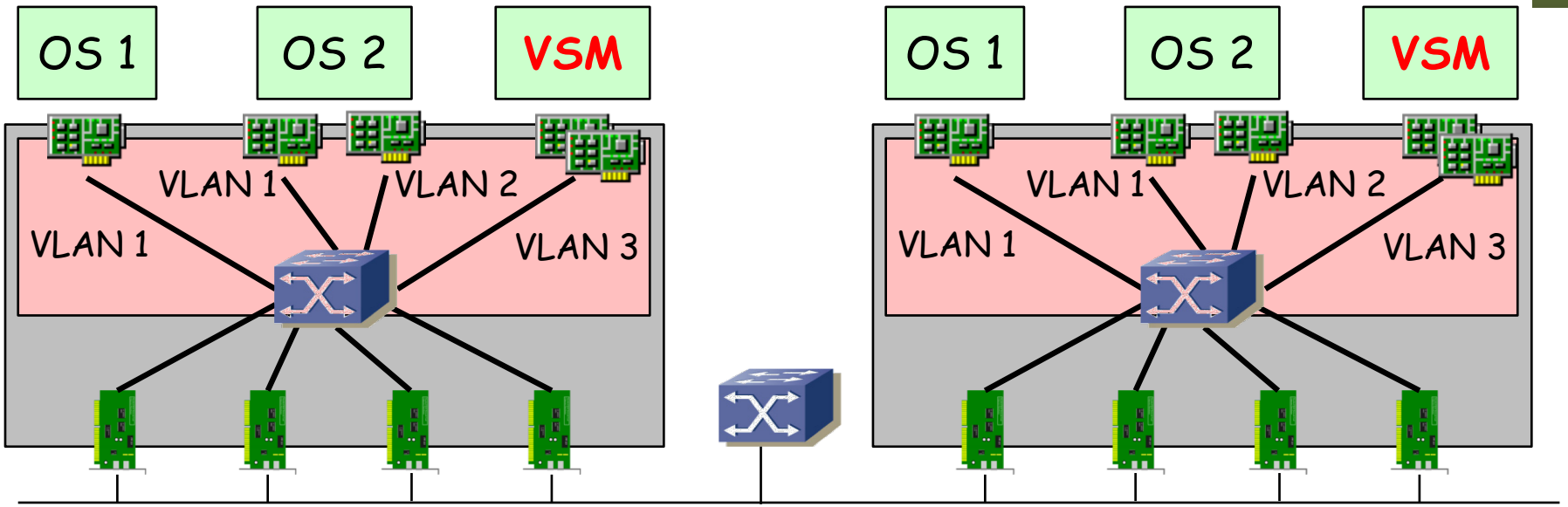
Virtual Switch

- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- ¿Y el supervisor? (…)



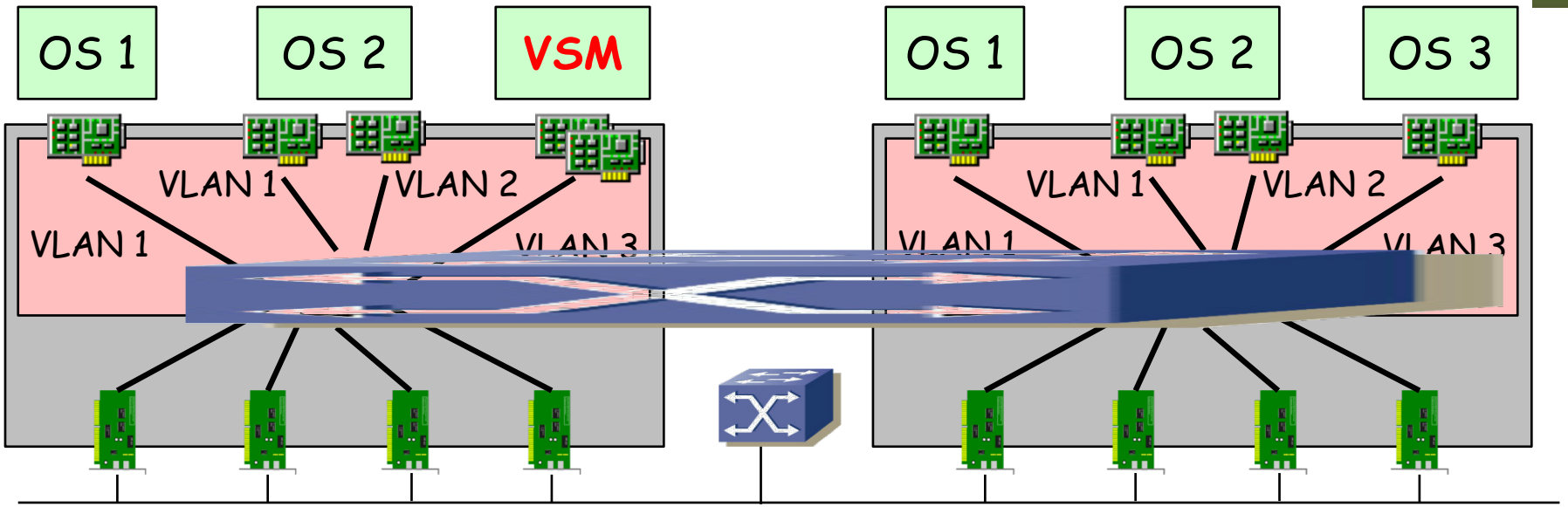
Virtual Switch

- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- El supervisor corre como una máquina virtual (Ej: Cisco 1000v)
- (…)



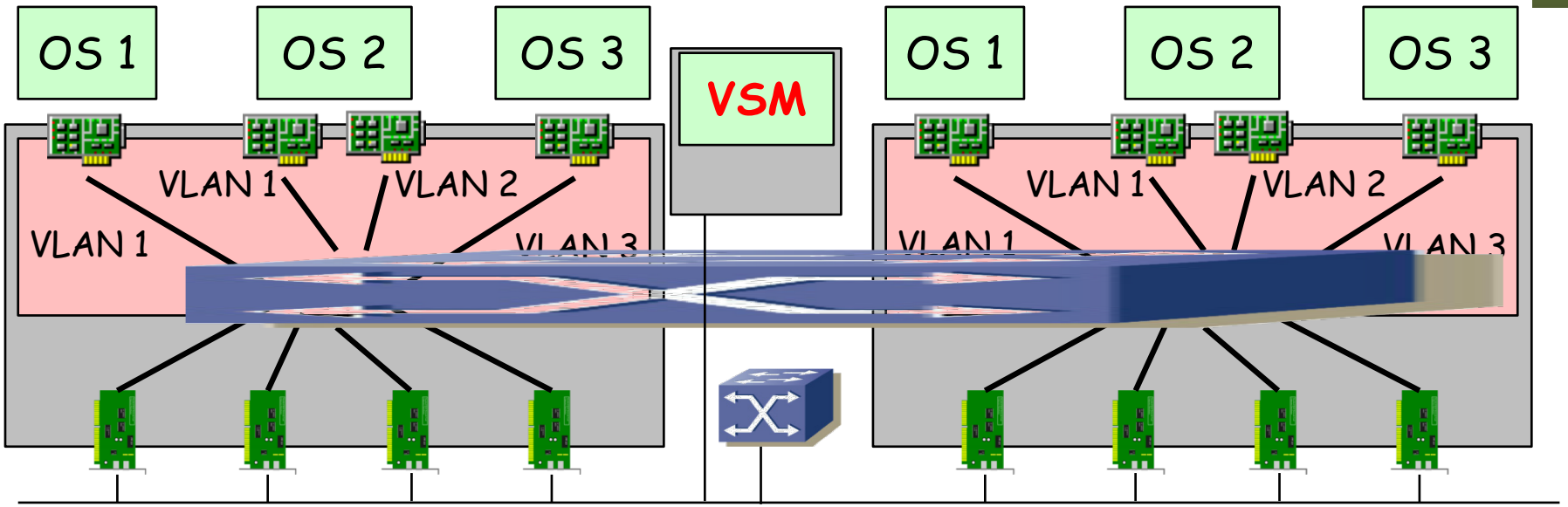
Virtual Switch

- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- El supervisor corre como una máquina virtual (Ej: Cisco 1000v)
- Vale con un supervisor para controlar varios hosts y entonces es como si todos formaran un switch virtual
- (…)



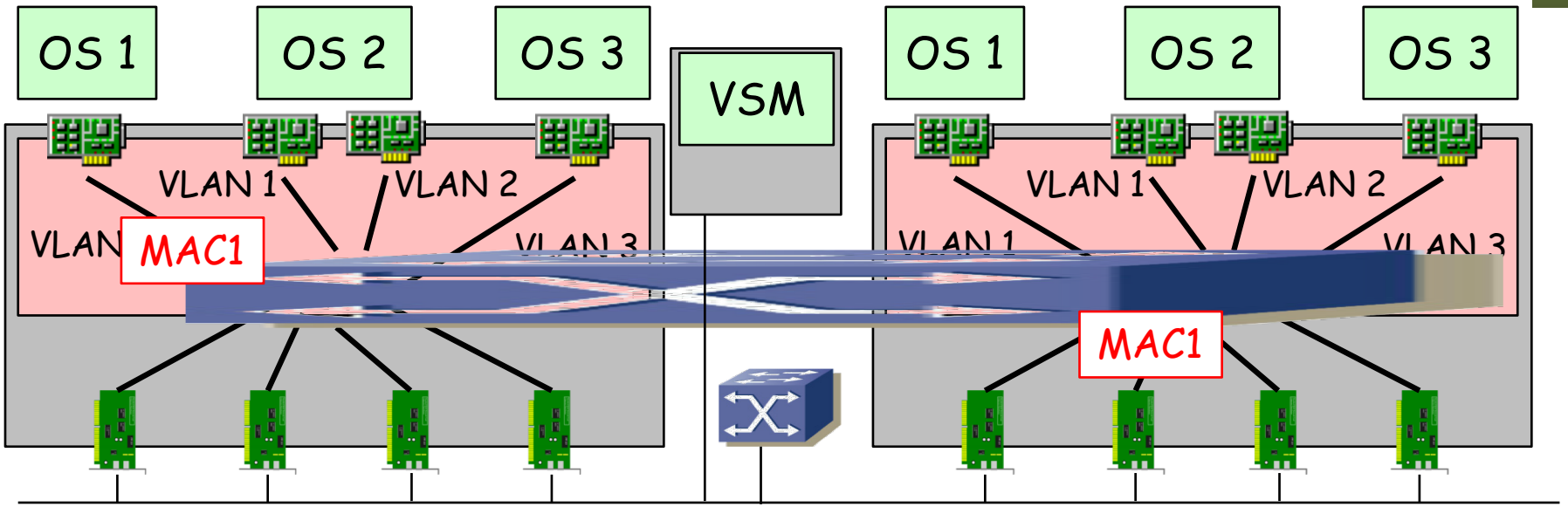
Virtual Switch

- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- El supervisor corre como una máquina virtual (Ej: Cisco 1000v)
- Vale con un supervisor para controlar varios hosts y entonces es como si todos formaran un switch virtual
- Ese supervisor podría correr en su propio hardware (Ej: Cisco 1100)



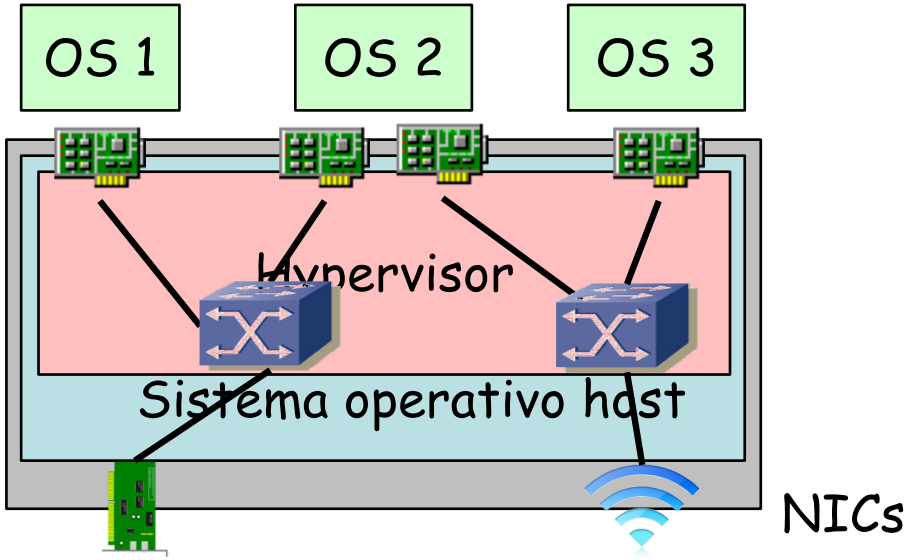
Virtual Switch

- Cada host mantiene su propia tabla de reenvío
- El switch de un host no tiene conocimiento de las MACs aprendidas en otro, ni aunque formen parte del mismo switch distribuido
- Es decir, aunque hablemos de un switch distribuido NO hay una base de datos de filtrado única
- Eso quiere decir que una dirección MAC puede aparecer más de una vez, dado que puede aparecer en todas las tablas de host



Escenario *hosted*

- Es utilizado principalmente en soluciones de escritorio
- En estos casos suele haber 1 ó 2 NICs (Ethernet + WiFi)
- La NIC WiFi se virtualiza haciéndola parecer una Ethernet



upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

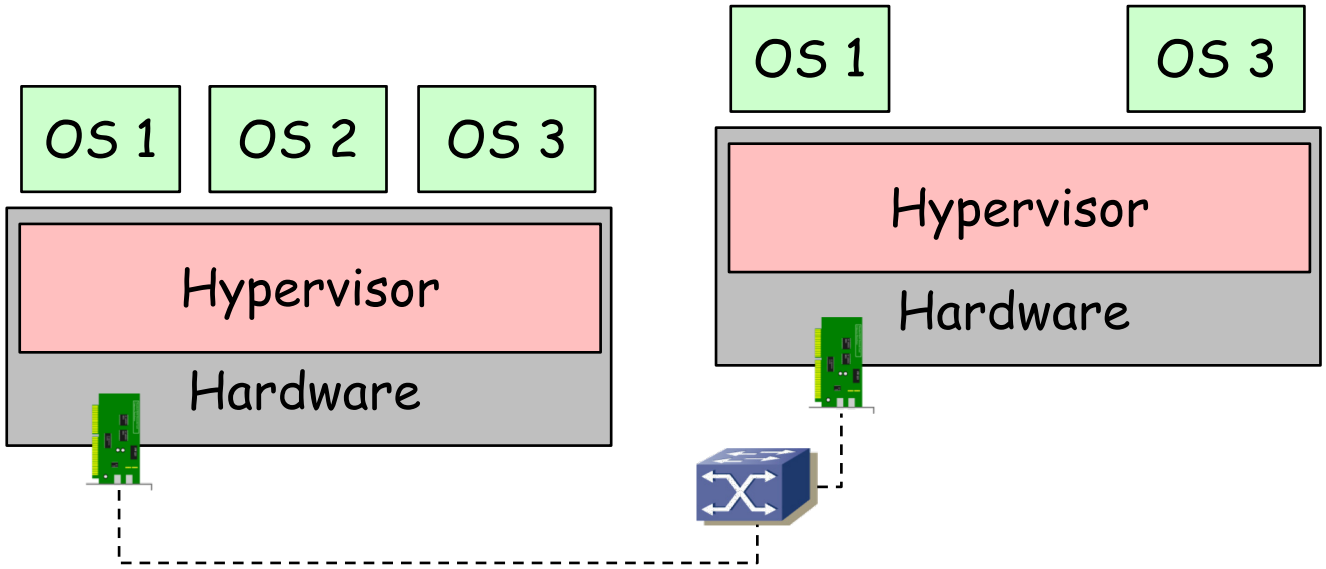
Gestión y *provisioning* de máquinas virtuales

Gestión

- El hypervisor y sus VMs se pueden gestionar remotamente
- Tareas como crear una VM, arrancarla, detenerla, clonarla, hacer un backup, migrarla, etc
- Virtualization Infrastructure Management (VIM)
- Software que corre en un controlador, normalmente un ordenador independiente
- Puede que el host tenga alguna NIC dedicada a la gestión
- Se pueden crear VMs a partir de *templates*

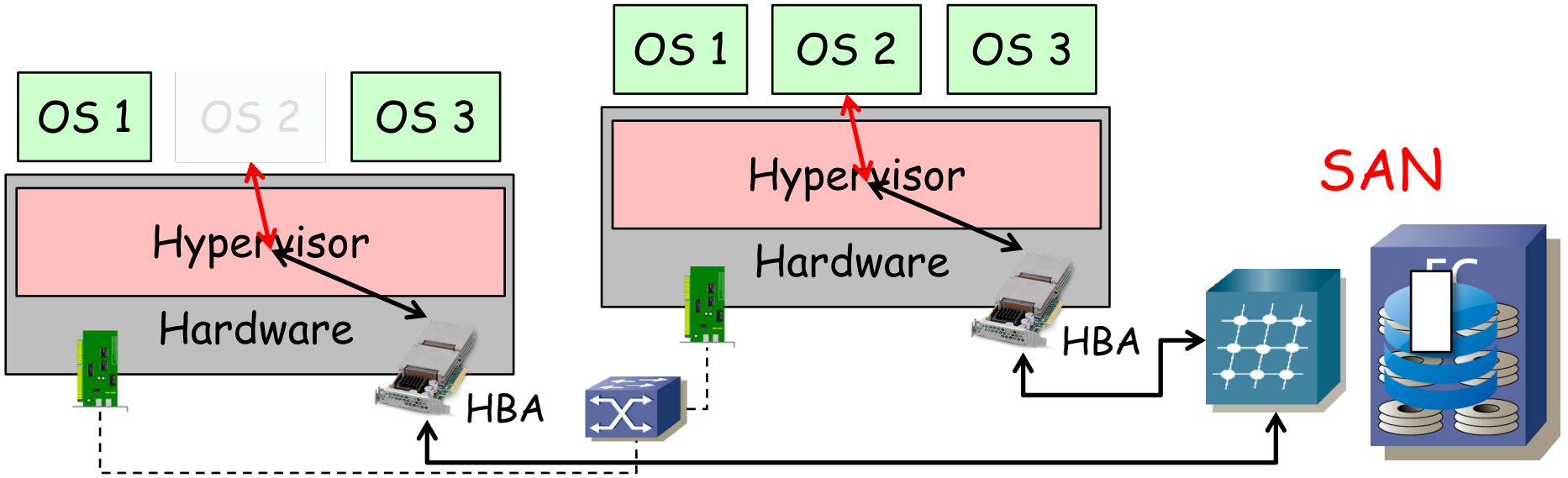
Virtual Machine Mobility

- Una VM se puede trasladar (en funcionamiento) a otro host (…)
- Por ejemplo porque las VMs del host supongan en ese momento una alta carga y se pretende distribuirla
- No cambia su identidad ni detiene sus conexiones de red (no modifica la dirección MAC de la vNIC)
- Se mantiene su estado completo
- Ambos hosts deben estar en la misma LAN (VLAN)
- Hypervisor manda un ARP gratuito al reanudarla en el otro host
- (…)



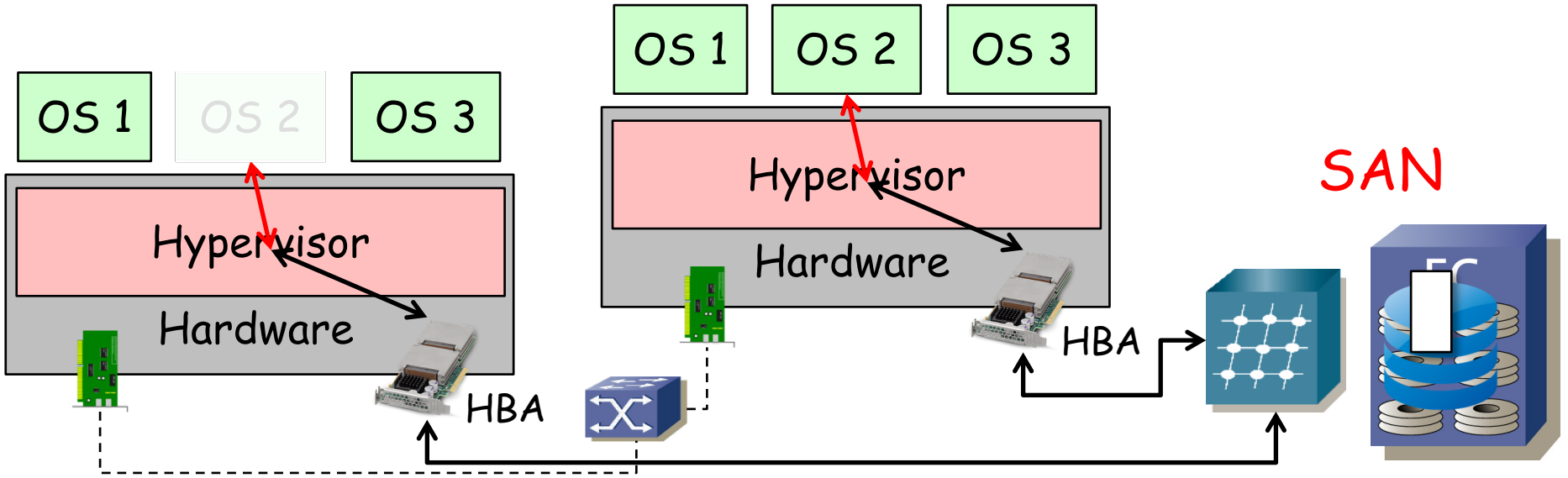
Virtual Machine Mobility

- La imagen de la máquina virtual está en una SAN accesible por ambos hosts
- Esto puede requerir un sistema de ficheros que permita acceso concurrente desde los dos hosts (*clustered*)
- Consolidar las VMs puede permitir apagar servidores, incluso su refrigeración
- Ejemplo: VMware vMotion
  - Requiere RTT entre los hosts de menos de 10ms
  - Hay que mover la RAM (por partes)
  - Se puede hacer en segundos, pero según la RAM a mover y el BW disponible



VM Mobility y VLANs

- Las VMs en un host pueden pertenecer a diferentes VLANs
- Eso hace que el host deba recibir el tráfico de múltiples VLANs
- Además, si se pueden mover las VMs, le pueden venir VMs de cualquier VLAN
- El vSwitch no tiene forma de informar al switch físico de las VLANs que necesita
- Así que se acaba configurando para que reciba el tráfico de todas
- Eso implica que debe procesar el broadcast de todas ellas



upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Gestión y *provisioning* de máquinas virtuales

Beneficios de la virtualización

- Independencia del hardware
- Consolidación
  - Ahorro en hardware para correr los servicios (y espacio)
  - Ahorro en consumo eléctrico
  - Ahorro en refrigeración
- Sencilla separación de entornos de desarrollo, pruebas y producción
- Sencilla creación, backup y replicación
  - Facilita la migración a otro hardware
  - Creación de instantáneas y retorno a ellas
- Instalaciones menos atadas al hardware pues requieren drivers para el hardware virtualizado
- Permite mantener software (sistemas operativos) antiguos sobre hardware moderno (aunque no tengan drivers)

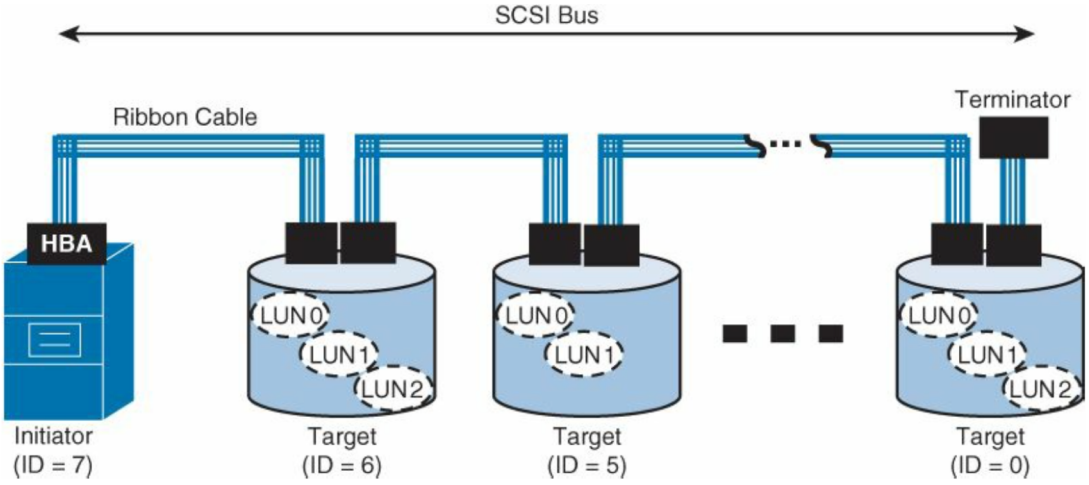
Desventajas de la virtualización

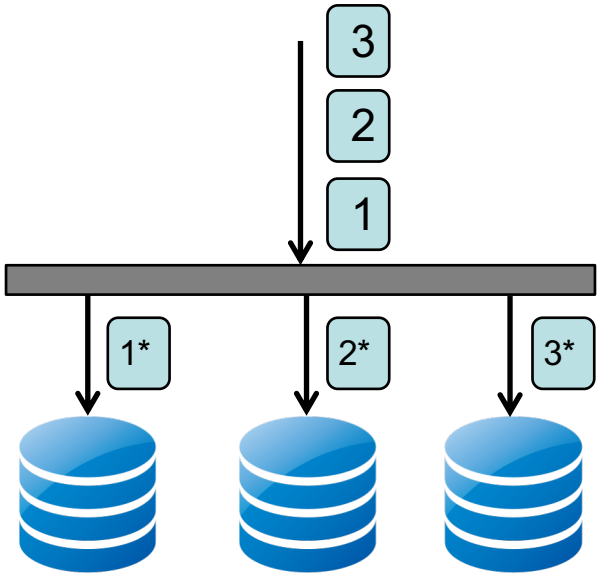
- Pérdida de rendimiento
  - Con aplicaciones con alta carga, que hacen un uso intensivo del hardware, puede no ser rentable
  - Hay que dimensionar la capacidad para la combinación de carga de VMs
- Compatibilidad con el hardware
  - Podemos contar con hardware especializado para el que no exista drivers en el hypervisor
- Un fallo hardware tiene efecto en múltiples VMs
- Depuración del sistema global más compleja, mayor acomplamiento
- Nuevas herramientas de gestión, nuevas habilidades requeridas al personal de IT

Virtualización del almacenamiento

Storage Virtualization

- Ya hemos visto varios casos:
  - Logical Units: Podemos segmentar un disco
  - RAIDs: Varios discos físicos se ven como una sola unidad

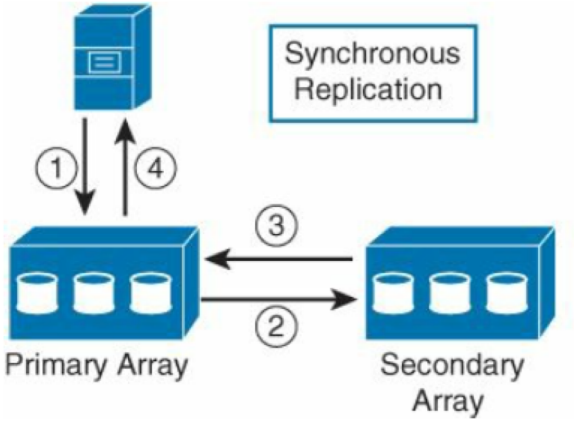


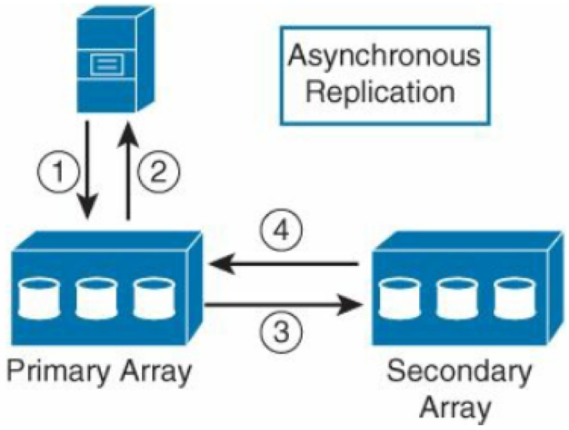


G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

Disk Array Virtualization

- *Partitioning*
  - Una cabina puede soportar subdividirse en dispositivos lógicos
  - Cada uno tendría asignados recursos de: discos, cache, memoria, puertos
  - Cada partición puede crear sus propias LUNs
- *Array-based data replication*
  - Múltiples cabinas pueden trabajar juntas en replicación
  - La *replicación síncrona* se basa en devolver confirmación de haber almacenado el dato cuando se ha escrito en las dos
  - La *replicación asíncrona* se basa en copiar después o periódicamente los datos (no bloquea la respuesta al usuario)

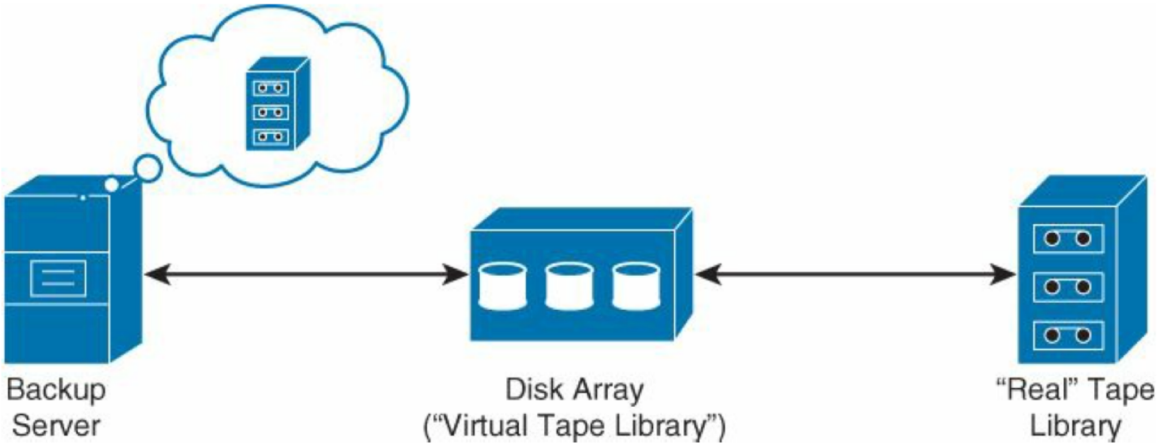




G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

Virtual Tape Library

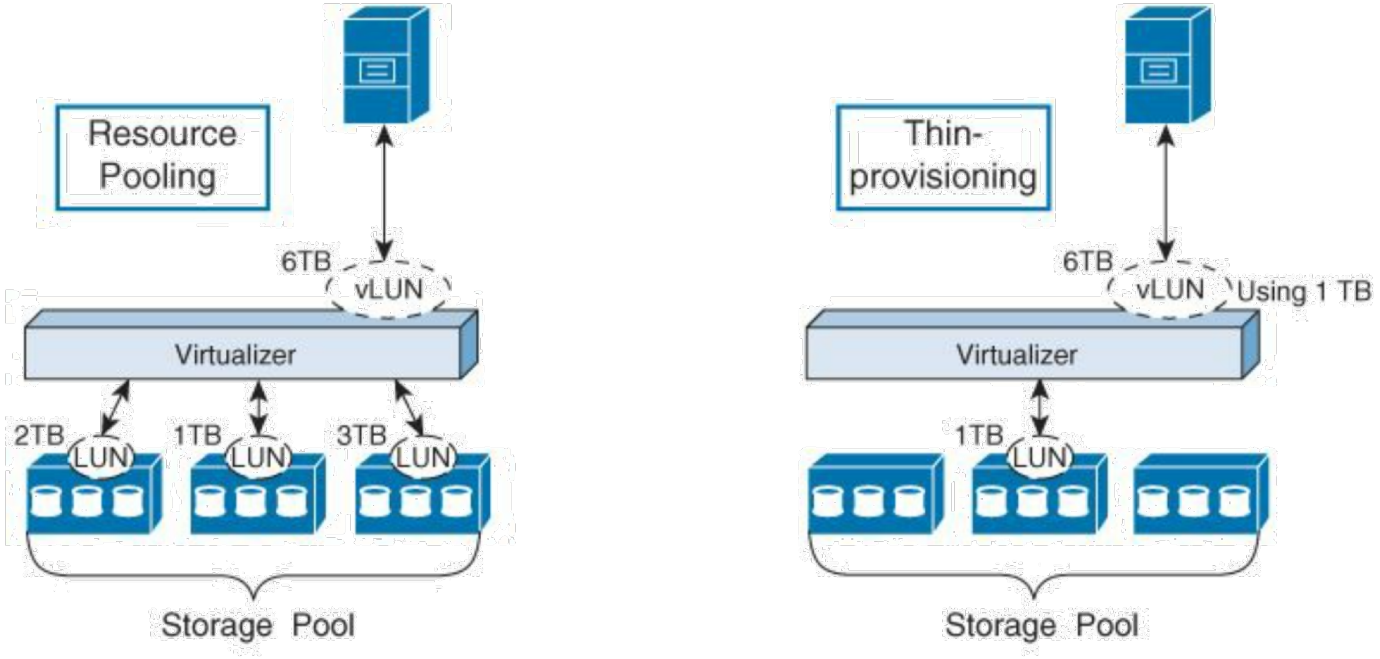
- El servidor accede a la cabina como si fuera la biblioteca de cintas
- La cabina actúa como una cache
- *Deduplication*
  - No manda al almacenamiento una segunda vez algo que ya existe
  - Apunta simplemente una referencia
  - Si luego uno de los dos se modifica puede guardar solo las modificaciones
  - También en el escenario de cabina de discos independiente
  - Ahorra por ejemplo bastante con imágenes de OS
  - También puede hacer compresión



G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

Virtualización de LUNs

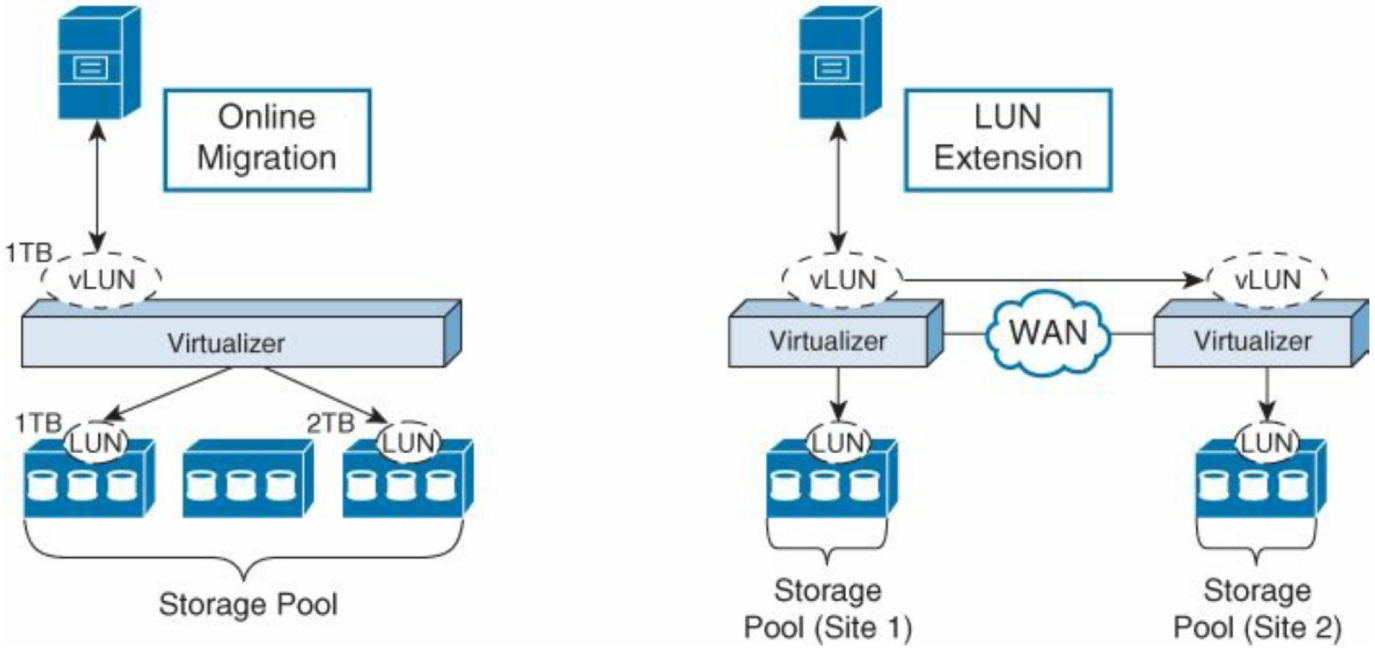
- Un virtualizador se interpone entre el servidor y la LUN
- Ofrece una vLUN al servidor
- Eso le permite modificar cómo la implementa sin alterar al servidor
- Puede agregar varias LUNs en una (*storage resource pooling*)
- Puede ofrecer una vLUN de mayor capacidad que la que realmente está empleando (*thin-provisioning*)
- Esto puede llevar a *over-subscription* y como tal funciona mientras todas las vLUNs no quieran usar toda la capacidad que anuncian



G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

Virtualización de LUNs

- Permite la migración de la LUN de una cabina a otra de forma transparente (*online migration*)
- Por ejemplo para cambiar a discos o un RAID más rápido
- El virtualizador puede dar la funcionalidad para la replicación entre dos cabinas, por ejemplo en DCs alejados
- Un virtualizador en cada DC puede estar ofreciendo la vLUN a los servidores de ese DC (*LUN extension*)



G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Virtualización del escritorio

VDI

- VDI = *Virtual Desktop Infrastructure*
- El PC del usuario pasa a ser un *thin client*
- El escritorio que muestra es el de una máquina virtual en un servidor
- Ejemplos: VNC, X Windows, RDP (*Remote Desktop Protocol*), Citrix XenDesktop (ICA protocol = *Independent Computing Architecture*), VMware Horizon (with View), PCoIP (Amazon Workspaces), TeamViewer, Oracle Secure Global Desktop
- La experiencia del usuario depende del servidor y de la red
- *Virtual Desktop Clouds*

Application Virtualization

- En este caso el usuario ve solo la aplicación en cuestión
- Puede ejecutarse en el servidor o puede enviarse el binario al cliente
- Si se ejecuta en el servidor es más sencillo que funcione en cualquier plataforma del usuario
- Si se envía al PC del usuario debe ser capaz de ejecutarlo (nativamente o virtualizado) y le puede permitir modo offline
- Ejemplo: Citrix XenApp, Oracle Secure Global Desktop