



INTRODUCCIÓN A LA VIRTUALIZACIÓN

Virtualización - Concepto y definición

- Es la creación de abstracciones de dispositivos o recursos físicos con el fin de ofrecer las mismas a uno o más entornos de ejecución. Más precisamente, es la *metodología* de dividir los recursos de una computadora en múltiples ambientes de ejecución más o menos aislados entre si por medio de la aplicación de conceptos o tecnologías tales como partición de hardware y software, simulación total o parcial de un equipo, etc.
- Concretamente, es una técnica que permite ocultar funcionalidades o capacidades de un dispositivo (disco rígido, placa de red, memoria) o recurso (server, red, OS), sumar las de varios con el fin de presentarlos como otra entidad con capacidades diferentes o bien crear un equivalente virtual de los mismos.
- la virtualización **separa y hace independientes** a los recursos virtualizados del sustrato físico subyacente, presentándolos de manera **transparente** a usuarios y aplicaciones que los utilizan sin distinguirlos de los reales.

Un poco de historia

Porque se empezó a virtualizar...

En los 60s las computadoras eran voluminosas, escasas y caras.

Si un usuario no podía usar un equipo porque este estaba ocupado, se perdía tiempo y dinero.

Para subsanar esto, se desarrollaron OS que permitían a un equipo físico multiplexarse en copias por software de si mismo, dentro de las que se podía correr otro OS.

Así, distintos usuarios compartían un único mainframe sin usar un OS común.

Al mismo tiempo, para sobrellevar las limitaciones del hardware de la época (y ahorrar dinero, la "I" en "RAID" refería a "I"nexensive), se comenzó a virtualizar recursos tales como memoria y discos rígidos.

Un poco de historia

...y por que se perdió la necesidad de virtualizar equipos completos

Para finales de los 80s, las computadoras eran considerablemente más potentes, numerosas y baratas que en los 60s, y la necesidad de multiplexar un equipo para varios usuarios se había eliminado mediante el uso de OS multitarea y multiusuario.

En otras palabras: la razón original para virtualizar equipos completos había desaparecido.

Esto **no** significaría el fin de la virtualización (la primera especificación de la JavaVM data de 1996), pero si hubo un impasse en la investigación hasta fines de los 90s.

Por qué se virtualiza hoy?

- Para consolidar las tareas de diferentes servers con hardware subutilizado en un menor número de equipos (**consolidación de servidores**)
Beneficios: permite ahorros en hardware, consumo de electricidad, administración, infraestructura y espacio físico.
- Para ejecutar múltiples OS de manera simultanea en un mismo equipo, incluso en aquellos en los que no podrían correr de manera nativa.
- Para generar ambientes de testing y debugging aislados y seguros donde correr aplicaciones nuevas y/o no confiables.
El fallo de un equipo o proceso virtualizado no afecta a otros corriendo en paralelo y tampoco lo hace un ataque ya que no hay una superficie común susceptible al mismo.

Por qué se virtualiza hoy?

- Para asignar a OS y/o entornos de ejecución virtualizados los recursos que necesitan y no más, permitiendo hacer un uso eficiente de los mismos.
- Para emular recursos y/o dispositivos de hardware vía software o presentar aquellos disponibles como algo diferente.
- Para simplificar la administración de sistemas, reduciendo los tiempos de clonado, migración y recovery de los mismos, minimizando el downtime e independizándolos (hasta cierto punto) del hardware.
- Para correr aplicaciones legacy no compatibles con OS o hardware moderno o bien para permitir la portabilidad de aplicaciones (el "*write-once-run-everywhere*" de Java).

Formas de virtualización - El concepto de VMM o Hypervisor

Un **VMM (Virtual Machine Monitor) o Hypervisor** es un software que permite que múltiples OS se ejecuten simultáneamente en un mismo equipo.

El término VMM apareció en 1972 y se refería a la interface a través de la cual un OS huésped podía *acceder a servicios virtualizados por el programa de control* (el **CP** de los sistemas operativos CP/CMS de IBM), de manera *análoga a una llamada al sistema* por parte de una aplicación corriendo dentro de un OS normal.

A continuación, veremos los dos tipos de VMM existentes, clasificados según como interactúan con el hardware subyacente y las VMs que corren por sobre ellos.



- **VMM de tipo 1 o nativo**

También llamado *unhosted* o *baremetal*.

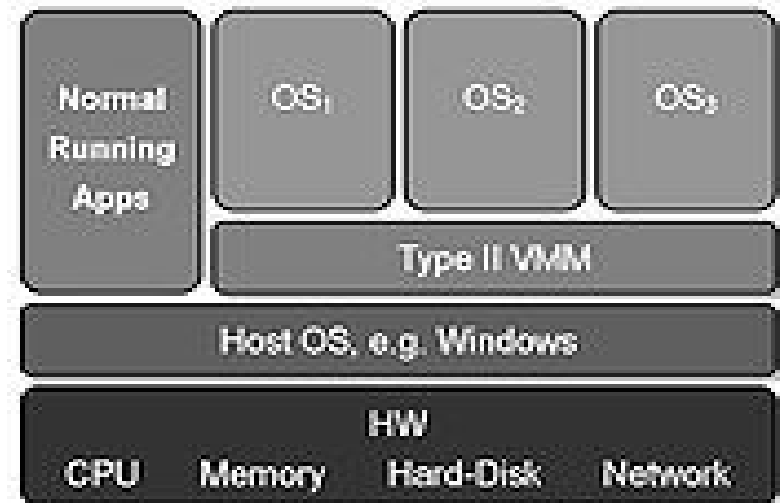
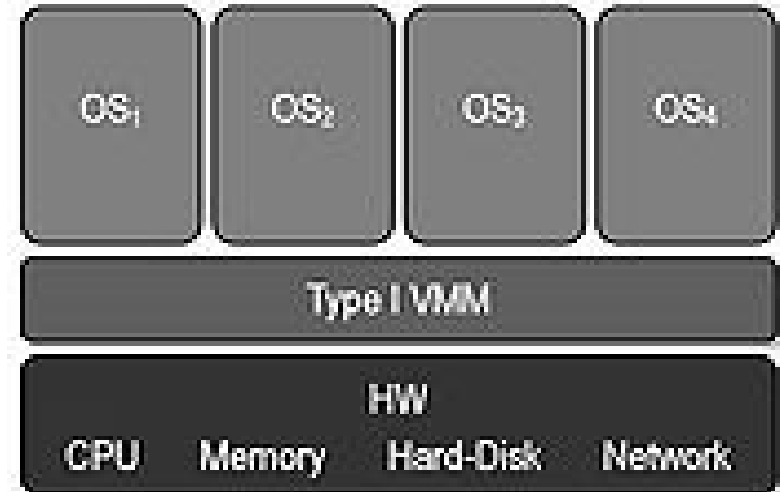
En este caso, el VMM corre directamente sobre el hardware físico, para ofrecer toda o parte de las funcionalidades del mismo a las VMs que sostiene.

Este es el esquema usado por las primeras VMs.

- **VMM de tipo 2 o hosteado**

En este caso el VMM se ejecuta sobre un OS que corre sobre el hardware físico.

El VMM captura las llamadas del hardware de la VM y las pasa al OS que a su vez las pasa al hardware físico subyacente.



Formas de virtualización

- *Emulación de un sistema no nativo*

Clave: software emulando hardware no nativo.

Cuando se emula hardware distinto al nativo para permitir que aplicaciones y/o OS concebidos para otras arquitecturas de procesador se puedan ejecutar sobre el mismo, que no los soporta.

No es necesario que el hardware emulado exista físicamente: en algunos casos bastará con que exista la especificación del mismo. Esta forma de virtualización permite la portabilidad de aplicaciones, ya que solo el emulador debe ser portado de una arquitectura a otra.

Ejemplos: **JavaVM**, **CLI** de .Net.



Formas de virtualización

- *Emulación de un entorno de ejecución*

Clave: combatir obsolescencia de software, portabilidad de aplicaciones.

Cuando se emula *unicamente* el entorno de ejecución de una o más aplicaciones.

Usado sobre todo para hacer frente a la obsolescencia de software desarrollado para plataformas/OS antiguos o bien para portabilidad entre distintos OS.

Ejemplos: [DOSBox](#), [Wine](#), [Cygwin](#).



Formas de virtualización

- *Emulación del hardware subyacente (ejecución nativa)*

Clave: interface que emula y multiplexa el hardware físico subyacente.

Varias VMs independientes corren sobre un VMM de tipo 1 o 2. Cada una de ellas corre su propio OS, que debe ser capaz de correr sobre el hardware físico subyacente.

El VMM multiplexa el hardware, particionándolo para repartirlo y presentarlo a cada VM.

Ejemplos: VirtualBox, VMware ESX Server, VMware Player, VMware Server.

Formas de virtualización

- *Virtualización a nivel de OS*

Clave: división del OS en userlands más o menos aisladas entre si.

Cuando un mismo OS se divide en varios *entornos virtuales* independientes que comparten un kernel común, o al menos similar.

Esto último la convierte en una técnica de virtualización muy eficiente (no requiere tantos recursos para cambios de contexto ya que las *syscalls* no son emuladas) aunque limita la variedad de OS emulados a los soportados por un kernel común.

Ejemplos: Solaris containers, FreeBSD jail, entorno chroot.

Formas de virtualización

Paravirtualización

Clave: virtualización parcial que excluye operaciones costosas del VMM.

Cuando se utiliza una *para-API* presentada a una o más VMs, siendo la misma similar, pero no idéntica al hardware físico subyacente. Permite reducir la porción de tiempo de ejecución que el huésped virtualizado(guest) invierte realizando operaciones más costosas (en ciclos de CPU) dentro del entorno virtualizado que fuera del mismo, mejorando el rendimiento del primero y simplifica al VMM.

Hay que señalar que la paravirtualización requiere que el OS a virtualizar sea portado a la *para-API*; una distribución común del mismo no correrá en un VMM paravirtualizado.

Ejemplos: Xen, L4

Formas de virtualización

Virtualización de recursos de hardware

Clave: literalmente, presentar partes del hardware a un OS cualquiera como si otra cosa.

Repetimos: la virtualización *es la creación de abstracciones de dispositivos o recursos físicos con el fin de ofrecer las mismas a uno o más entornos de ejecución*. Dichos recursos pueden ser, memoria, redes, discos duros, etc.

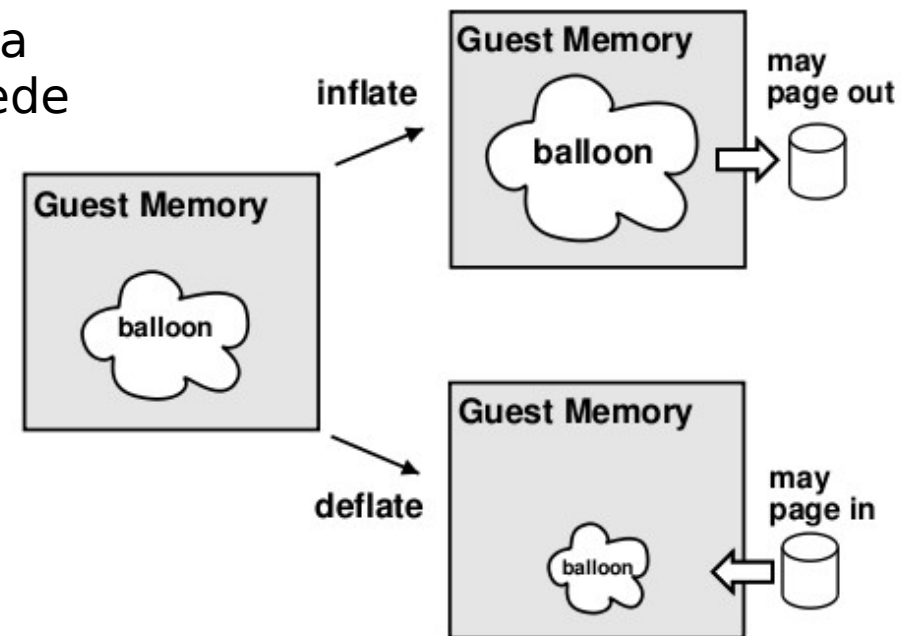
Ejemplos:

- Niveles de RAID. Presenta varios discos físicos como uno solo o viceversa
- VPN en la que se establecen circuitos virtuales sobre los nodos de la red física subyacente

Conceptos importantes

Memory Ballooning

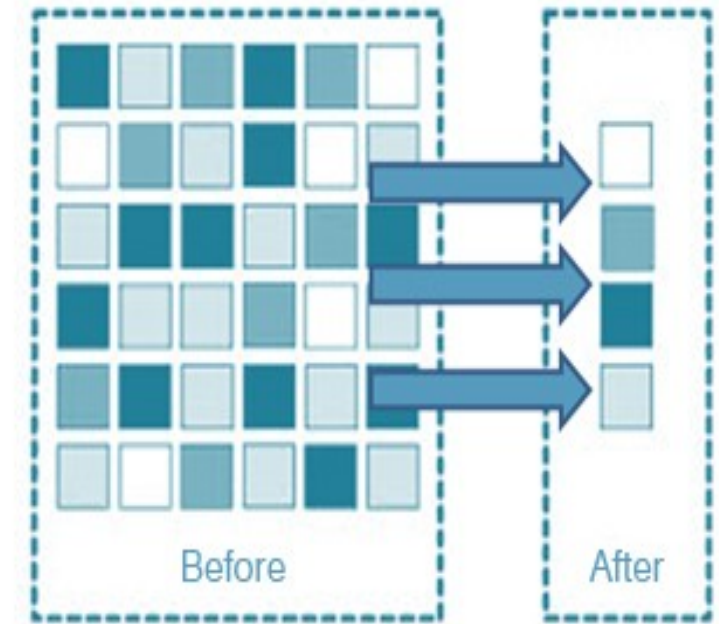
- Originalmente, para cambiar la memoria asignada a una VM era necesario reiniciarla.
- memory ballooning = hot-swapping memory para VMs
- Cuando “el globo se infla” se reserva memoria de una VM, que el SO huésped (guest) no puede usar ni asignar. Dicha memoria puede ser redistribuida por el hypervisor a otras VMs.
- Cuando se “desinfla” el hypervisor devuelve la memoria al SO huésped para su uso.



Conceptos importantes

Deduplicación

- Utiliza una función que calcula y compara los hash de secciones de bits de tamaño fijo.
- Si se encuentra un patrón duplicado, se lo reemplaza con una referencia al original.
- Según como se lo mire, existen cuatro formas de deduplicar datos:



Deduplicación post-proceso

Los datos nuevos son guardados y luego procesados en busca de duplicados.

Deduplicación en tiempo real

Los hash son calculados y comparados en tiempo real, a medida que los datos nuevos llegan al dispositivo.

Deduplicación del lado fuente

Los datos son deduplicados en su filesystem original.

Deduplicación del lado destino

Se deduplican los datos bajados a destino, por ejemplo, en un backup.

Conceptos importantes

Open Virtualization Format (OVF)

Problema: como migrar una VM de un hypervisor a otro?

Solución: un standard para VMs independiente de hypervisors

En Septiembre de 2007 la DMTF (Distributed Management Task Force) recibió una propuesta de Dell, IBM, Microsoft, Vmware, HP y XenSource para la creación del "Open Virtual Machine Format", más tarde conocido como OVF.

La primera especificación, v1.0.0, fue publicada en Septiembre de 2008 y la última a la fecha (Octubre de 2011), v1.1.0, en Enero de 2010.



Conceptos importantes

Open Virtualization Format (OVF)

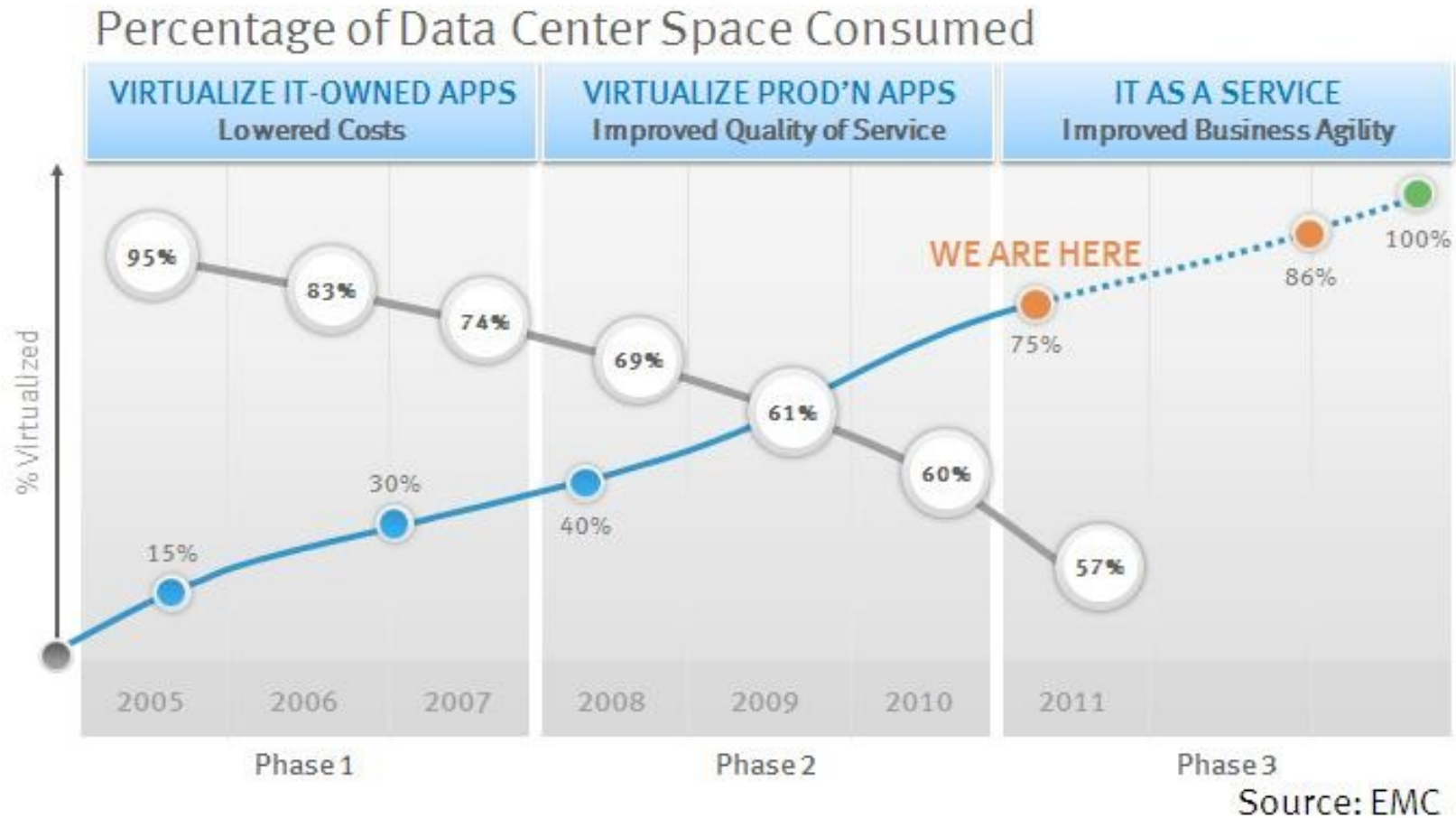
Que es OVF?

Desde el punto de vista del usuario:

Un formato de archivo que permití hacer packaging de software que una vez instalado agrega a la infraestructura preexistente una solución contenida y consistente por si misma usada para alcanzar un objetivo particular, ie: un sistema L/WAMP (Linux/Windows + Apache + MySQL + PHP).

Desde el punto de vista técnico:

Un OVF es un medio de transporte para plantillas de VMs. One archivo OVF puede contener una o varias VMs, que deben estar instaladas en una plataforma de virtualización para poder correr.



FUENTE: <http://fountnhead.blogspot.com/2011/09/documenting-it-transformation-since.html>



● Bibliografía (para profundizar un poco)

- www.virtualizacion.com - Bueno para empezar.
- http://en.wikipedia.org/wiki/Popek_and_Goldberg_virtualization_requirements
- http://es.wikipedia.org/wiki/Maquina_virtual - Ver también el artículo en inglés.
- <http://en.wikipedia.org/wiki/Hypervisor>
- www.virtualizationreview.com - Mucha info y mucha publicidad.
- www.kernelthread.com/publications/virtualization - Amit Singh.
- <http://queue.acm.org/detail.cfm?id=1017000> - Mendel Rosenbaum.
- http://www.vmware.com/pdf/ovf_whitepaper_specification.pdf
- http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf