

# Soluciones robustas de virtualización con software libre

Morfeo Formación

Miguel Vidal, José Castro

{mvidal,jfcastro}@libresoft.es  
GSyC/Libresoft – URJC

1 de junio de 2010

*we study libre software*



Universidad  
Rey Juan Carlos

GSyC

LibreSoft

© 2010 Miguel Vidal, José Castro.

Some rights reserved. This work is licensed under a Creative Commons Attribution-Share Alike 3.0 License, available at <http://creativecommons.org/licenses/by-sa/3.0/>

# Agenda

- Parte 1: Qué es la virtualización
- Parte 2: Tipos de virtualización





# Definición

Es la combinación de hardware y software que permite a una única máquina comportarse como si fueran varias máquinas.

Esto implica hacer que un recurso físico (como un servidor, un sistema operativo o un dispositivo de almacenamiento) funcione como múltiples recursos lógicos.

También se conoce a la virtualización como la **abstracción** de recursos de una máquina.

*we study libre software*

# Orígenes

- El término “virtualización” fue acuñado en los años 60 para referirse a una máquina virtual (también conocido como “pseudo máquina”).
- En los 70 se desarrollaron varios sistemas de máquinas virtuales basadas en IBM: el CP-40, el CP-67, el famoso VM/370...
- Un componente llamado *Virtual Machine Monitor (VMM)* corría directamente sobre el hardware “real”.
- Se podían crear varias máquinas virtuales y cada instancia ejecutaba su propio sistema operativo.

## Extensiones de virtualización para x86

- Desde 2005, Intel y AMD han añadido soporte hardware para la virtualización.
- Intel Virtualization Technology (VT) *codename* Vanderpool
- AMD Virtualization (AMD-V) *codename* Pacifica
- Añaden una funcionalidad específica para permitir a los hipervisores un rendimiento mayor en virtualización completa.
- La virtualización completa es más sencilla de implementar.

# Hipervisores (I)

- Los *Virtual Machine Monitors* (aka **hipervisores**) permiten que diferentes sistemas operativos, tareas y configuraciones de software coexistan en una misma máquina física.
- El hipervisor abstrae los recursos físicos de la máquina anfitriona para las distintas “máquinas virtuales”.
- Los hipervisores garantizan un nivel de aislamiento entre los invitados.
- Los hipervisores también proporcionan una interfaz única para el hardware.

# Hipervisores (y II)

Hay dos clases de hipervisores:

- **Tipo 1** (o “nativo”, “bare-metal”): el hipervisor es una capa entre el hardware y el sistema operativo. Al sistema operativo del huésped se le llama Dominio de Control y corre sobre el hipervisor.
- **Tipo 2** (o “hosted”): el hipervisor es una capa de software que corre sobre el sistema operativo del huésped.

*we study libre software*

# Razones para virtualizar

- Permite el aislamiento de aplicaciones y usuarios sobre la misma máquina para que no se interfieran entre sí.
- Entorno de servidor: fusiona varios servidores o servicios en una misma máquina (email, web, dns, etc.).
- Ejecutar diferentes sistemas operativos o software específico para un tipo de CPU.
- Entornos de test fáciles y seguros: entornos de depuración, desarrollo y aislamiento para estudiar virus, gusanos, etc.
- Fácil desarrollo de software utilizando aplicaciones de software virtuales. Desde la perspectiva del negocio, es posible reducir los costes totales de propiedad (TCO).
- Minimiza el consumo de energía y la infraestructura de refrigeración en los *datacenters*.

# Conceptos básicos

- Al sistema operativo que ejecuta el software de virtualización se le conoce como **anfitrión** (*host*).
  - El anfitrión controla el hardware real.
- Al sistema operativo virtualizado se le conoce como **invitado** o **huésped** (*guest*).
  - Hay varios invitados en un mismo anfitrión.
  - Los invitados no deben interferir entre sí ni con el anfitrión.

# Conceptos de virtualización

- Al software de virtualización se le llama:
  - **Hipervisor.**
  - **Virtual Machine Manager** or **VMM.**
- El VMM o hipervisor corre como parte del sistema operativo del anfitrión (o es el anfitrión)
- A una instancia del hardware virtualizado se la conoce como **Máquina Virtual** o **VM.**
- Los sistemas operativos huéspedes corren dentro de una VM.

# Virtualización y Cloud Computing

- El Cloud Computing **no** es lo mismo que la gestión de la virtualización.
- Pero es cierto que la mayoría de los entornos de Cloud Computing hacen uso de la virtualización.
- La gestión del hardware está totalmente abstraída.
- La capacidad de la infraestructura en Cloud Computing es elástica (tanto para crecer como para menguar).

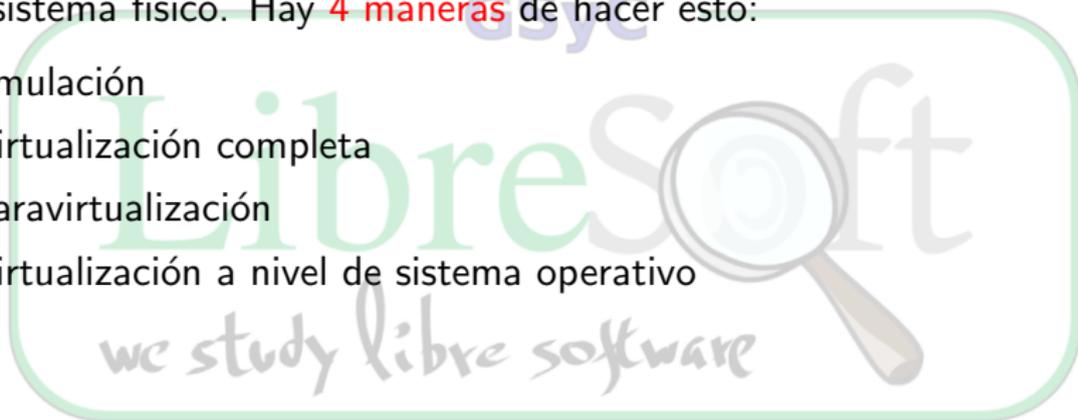
Las nubes son recursos hardware convertidos en un “*pool de recursos*”.



# Tipos de virtualización

El objetivo es crear la impresión de tener hardware separado en un único sistema físico. Hay **4 maneras** de hacer esto:

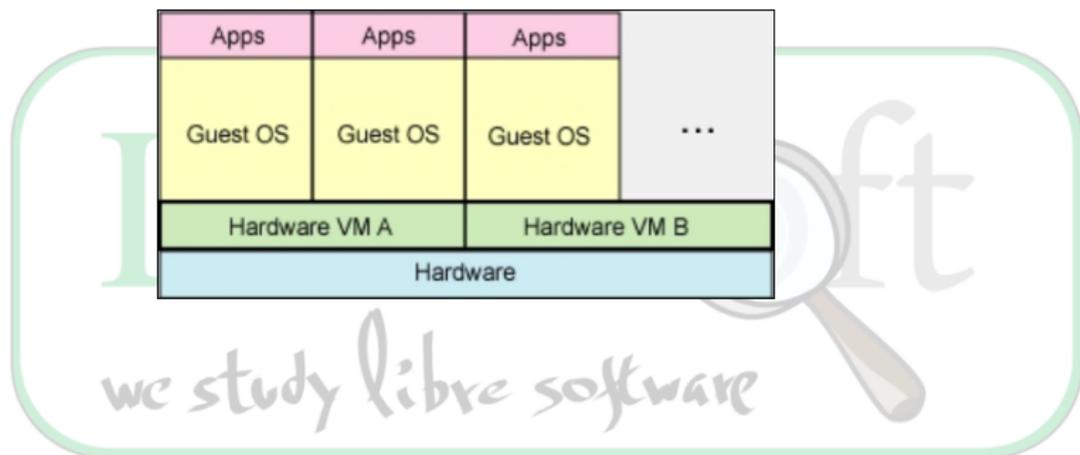
- 1 Emulación
- 2 Virtualización completa
- 3 Paravirtualización
- 4 Virtualización a nivel de sistema operativo



# Emulación

- La máquina virtual simula el hardware completo.
- Huéspedes sin modificar de diferentes arquitecturas hardware corren dentro de una VM.
- Utilizado para crear nuevos sistemas operativos o microcódigo para nuevos diseños de hardware antes de que el hardware esté disponible físicamente.
- **Ventajas:** Simular hardware que no está físicamente disponible.
- **Desventajas:** Bajo rendimiento y baja densidad (coste alto)
- *Ejemplo:* basilisk II, un emulador de Mac (m68k) para x86.

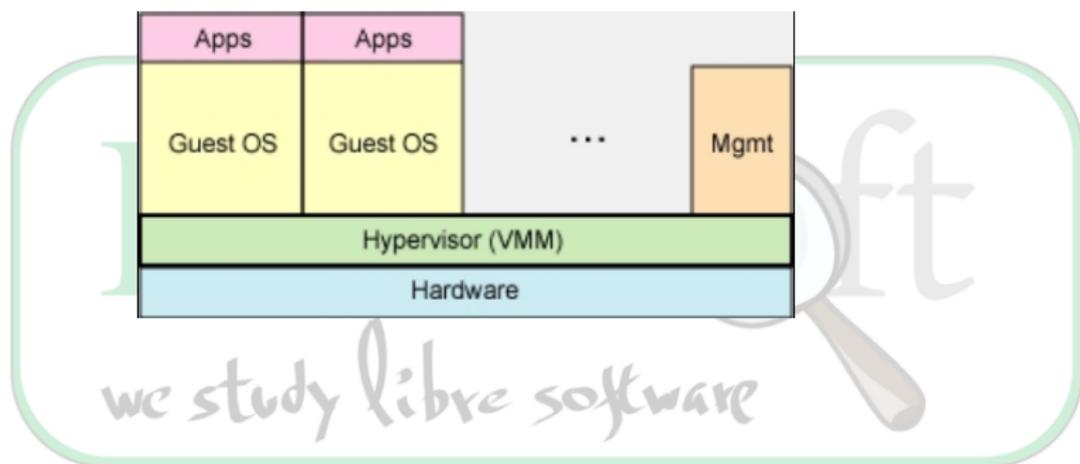
# Emulación: gráfico



# Virtualización completa

- Parecido a la emulación, anfitriones sin modificar.
- Se diferencia de la emulación en que los sistemas operativos a virtualizar están diseñados para correr en la misma arquitectura que el anfitrión.
- Combinado con hardware: CMT, Intel VT, AMD-V, estas CPUs controlan el acceso a instrucciones de virtualización.
- **Ventajas:** Flexibilidad, ejecuta diferentes sistemas operativos de distintos fabricantes.
- **Desventajas:** No se pueden emular otras arquitecturas. Rendimiento.
- *Ejemplos:* VirtualBox, KVM, Xen + Intel VT.

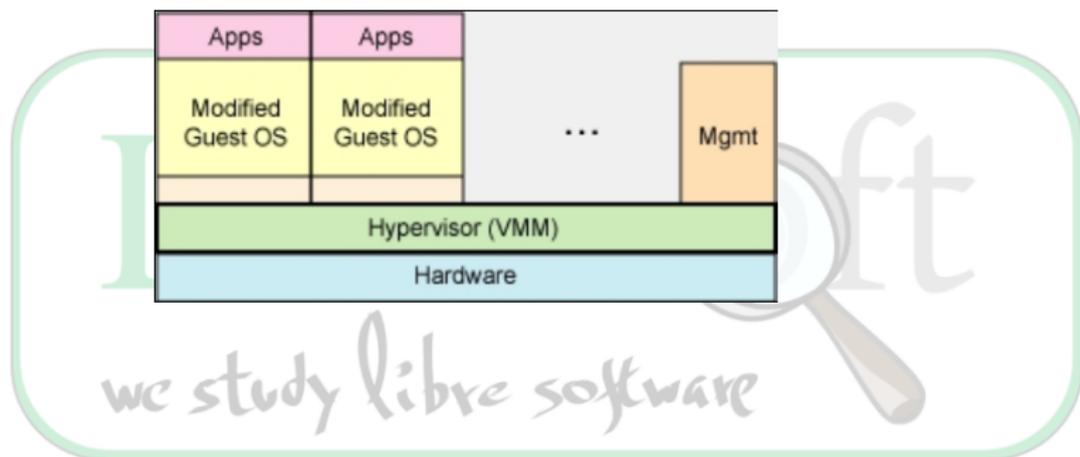
# Virtualización completa: gráfico



# Paravirtualización

- El hipervisor ofrece una versión modificada del anfitrión.
- La VM ofrecida tiene la misma arquitectura que el anfitrión.
- **Ventajas:** Rendimiento (ligero y rápido), escalabilidad y facilidad de gestión; fuerte aislamiento; permite virtualizar sin necesidad de CPUs especiales.
- **Desventajas:** Para la misma arquitectura. Requiere modificar el SO del invitado.
- **Ejemplo:** Xen con CPUs estándar.

# Paravirtualización: gráfico



# Virtualización a nivel de SO

- La virtualización se hace con la imagen tradicional del SO (sin hipervisor).
- El SO está modificado para permitir múltiples procesos en diferentes espacios de usuario aislados unos de otros.
- **Ventajas:** Rápido, capa ligera de virtualización. Rendimiento cercano al nativo. Densidad.
- **Desventajas:** Es difícil implementar el aislamiento fuerte.
- *Ejemplos:* Linux VServers, jaulas de FreeBSD, zonas/contenedores de OpenSolaris

## Otros tipos de virtualización

- **Virtualización de bibliotecas:** biblioteca **Wine** (subconjunto de la API de Win32 para poder ejecutar aplicaciones Windows)
- **Virtualización de aplicación:** entorno de ejecución virtual (con una API para la ejecución en diferentes plataformas).  
*Ejemplo:* **Java Virtual Machine**.
- **Virtualización de almacenamiento:** se provee de un *pool* de discos para el almacenamiento. *Ejemplos:* **LVMs** en Linux y **ZFS** en OpenSolaris.
- **Virtualización de escritorio:** se implementa el escritorio como servicio. *Ejemplo:* **SunVDI**.
- **Virtualización de red:** la tecnología **Crossbow** (nativa de OpenSolaris 2009.06) permite crear switches e interfaces de red virtuales.

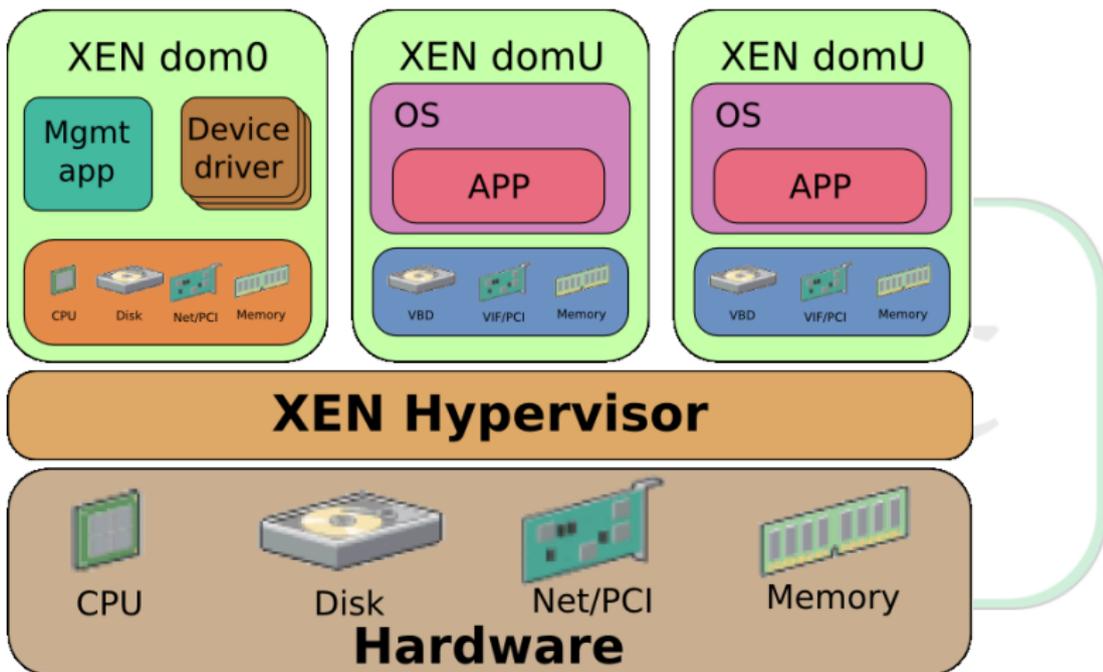
# Xen: Características

- Xen utiliza paravirtualización, lo que significa que el sistema operativo huésped tiene que estar modificado para usar el hipervisor.
- También permite virtualización asistida por hardware (IntelVT o AMD-V) para huéspedes sin modificar: es un desarrollo significativo ya que permite que sistemas operativos propietarios (como Microsoft Windows) puedan ser virtualizados.
- Es capaz de hacer migración de máquinas virtuales.
- Sobre arquitectura x86, el código de un kernel Xen anfitrión ejecuta en el anillo de protección 0 mientras que los dominios huéspedes ejecutan en el anillo 1 o anillo 3 (anillos: dominios de protección jerárquica)

## Xen: Prestaciones

- Independencia entre los sistemas virtualizados. Se puede reiniciar y crear independientemente.
- Uso mejorado del hardware: balanceo de recursos. Una máquina virtual puede hacer uso de los recursos que no utilizan las otras máquinas virtuales.
- Backup sencillo. Sólo con copiar la máquina virtual se puede levantar en un nuevo servidor. Xen también permite la migración en caliente, siendo muy flexible y minimizando el tiempo de recuperación en caso de fallo.
- Se pueden modificar parámetros como la RAM, el número de CPUs y el espacio en disco para cada necesidad específica de cada máquina virtual.
- Entornos de prueba y desarrollo: múltiples máquinas virtuales en un único servidor físico para probar y desarrollar.

# Cómo funciona



# Kernel-based Virtual Machine (KVM)

- KVM es un infraestructura de virtualización exclusiva del kernel Linux.
- Virtualización nativa (completa) utilizando los micros Intel VT y AMD-V.
- Estructura modular: la primera versión fue incluida en Linux 2.6.20 (febrero 2007).
- Una amplia variedad de sistemas operativos funcionan virtualizados con KVM: Linux, BSD, Solaris, Windows, ReactOS y existe una versión parcheada para poder ejecutar MacOS X.
- No requiere emulación: un programa en espacio de usuario (Qemu) utiliza la interfaz `/dev/kvm` para configurar el espacio de direcciones de las VMs hospedadas.

# Proyectos xVM OpenSolaris

- VirtualBox: VMs para todo. Diferentes plataformas.
- xVM server: Xen en OpenSolaris 2009.06.
- Dominios lógicos (LDOMs): Hipervisor tipo 1, “completo”, para plataformas SPARC.
- Zonas: virtualización ligera para OpenSolaris.
- Otros: Sun XVM Ops Center, Sun xVM VDI.

*we study libre software*

## Contenedores y zonas

- Tipo de virtualización que ejecuta varias instancias del mismo sistema operativo (y el mismo kernel).
- Zona Global y Zonas Locales: no se puede acceder al sistema de ficheros global (un chroot mejorado).
- Sin hipervisor ni hardware específico para virtualización.
- Despliegue y administración muy simple: zonecfg y zoneadm.
- x86: con soporte para “branding”.
- Desde 2005 forma parte de Solaris 10 (y OpenSolaris).
- Parecido a las jaulas de FreeBSD, OpenVZ (contenedores Linux) o Linux-VServer.

# Virtualización con LDOMs

- “Virtualización completa”, basada en un hipervisor tipo 1.
- Requiere CPUs especiales (Chip Multithreading = CMT).
- Cada dominio es una máquina virtual completa con un conjunto configurable de recursos.
- Los sistemas operativos que corren dentro del Dominio Lógico pueden iniciarse, pararse y reiniciarse de forma independiente.
- OS base: Solaris 10 / Opensolaris 2009.06
- OS de máquina virtual: Solaris 10, Opensolaris 2009.06, Sparc Linux y otros que soporten esta arquitectura.
- Pero por ahora, tan solo SunOS está soportado.

# Chip Multithreading (CMT)

- UltraSparc T1 / T2.
- Multithread. Un thread es similar a una CPU.
- Ejemplo: T1 tiene 8 cores con 4 threads/core.
- SSL directamente soportado por el hardware (1 MAU/core).
- LDOMs pueden asignar cores a las VMs.
- El hipervisor ejecuta en el firmware del servidor.
- Free/Open Hardware: <http://www.opensparc.net>
- Sun Fire T / Enterprise T / Blade T servers.



## Referencias

- Amit Singh “An Introduction to Virtualization” (2004):  
<http://www.kernelthread.com/publications/virtualization/>
- Comparison of platform virtual machines:  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_platform\\_virtual\\_machines](http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines)
- Jeanne Matthews *et al.* *Running Xen*. Prentice Hall, 2008  
(Chapter 1: “Background and Virtualization Basics”)