

Virtualización: Introducción y toma de contacto

*ALEJANDRO CALDERÓN MATEOS
ACALDERO @ ARCOS.INF.UC3M.ES*





Contenidos



1. ¿Qué es la virtualización?
2. ¿Es difícil usar la virtualización?
3. ¿Para qué utilizar la virtualización?
4. ¿Qué soluciones hay de virtualización?
5. ¿Qué factores considerar para la virtualización?





¿QUÉ ES LA VIRTUALIZACIÓN?





Wikipedia: virtualización

Virtualization - Wikipedia, the free encyclopedia - Mozilla Firefox

ES Español (España, internacional)

http://en.wikipedia.org/wiki/Virtualization

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Tech-Info Work News (Tech) News (live) Blog+

Gizmodo ES: enano: ordenador de ta... Virtual Linux Virtualization - Wikipedia, the free ...

Sign in / create account

Your continued donations keep Wikipedia running!

Virtualization

From Wikipedia, the free encyclopedia

See also: *virtual machine* and *hypervisor*

For the fictional teleportation process, see *Scanner (Code Lyoko)#Virtualization*.

In **computing**, **virtualization** is a broad term that refers to the **abstraction** of computer resources. One useful definition, from independent IT analyst firm Enterprise Management Associates, is "a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple logical resources; or it can include making multiple physical resources (such as storage devices or servers) appear as a single logical resource."

However, the term is an old one: It has been widely used since the 1960s or earlier, and has been applied to many different aspects and scopes of computing — from entire computer systems to individual capabilities or components. The common theme of all virtualization technologies is the *hiding of technical detail*, through **encapsulation**. Virtualization creates an external interface that hides an underlying implementation, e.g. by **multiplexing** access, by combining resources at different physical locations, or by simplifying a control system. Recent development of new virtualization platforms and technologies has refocused attention on this mature concept.

Like such terms as **abstraction** and **object orientation**, *virtualization* is used in many different contexts. This article reviews common uses of the term, divided into two main categories:

- **Platform virtualization** involves the simulation of **virtual machines**.
- **Resource virtualization** involves the simulation of combined, fragmented, or simplified resources.

Of course, virtualization is also an important concept in non-computer contexts. Many **control systems** implement a virtualized interface to a complex device; thus a modern car's gas pedal does much more than just increase the flow of fuel to the engine; and a **fly-by-wire** system presents a simplified "virtual airplane" which may have little to do with the physical implementation.

Contents [hide]

- 1 Platform virtualization
- 2 Resource virtualization
- 3 Virtualization examples
- 4 References
- 5 See also
- 6 External links

Platform virtualization

The original sense of the term *virtualization*, dating from the 1960s, is in the creation of a **virtual machine** using a combination of hardware and software. For convenience, we will call this **platform virtualization**. The term *virtual machine* apparently dates from the **IBM M44/44X** experimental paging system. The creation and management of virtual machines has also been referred to as creating *pseudo machines*, in the early **CP-40** days, and *server virtualization* more recently. The terms *virtualization* and *virtual machine* have both also acquired additional meanings through the years.

Platform virtualization is performed on a given hardware platform by "host" software (a *control program*), which creates a simulated computer environment (a *virtual machine*) for its "guest" software. The "guest" software, which is often itself a complete **operating system**, runs just as if it were installed on a stand-alone hardware platform. Typically, many such virtual machines are simulated on a given physical machine. For the "guest" system to function, the simulation must be robust enough to support all the guest system's external interfaces, which (depending on the type of virtualization) may include hardware drivers.

There are several approaches to platform virtualization, listed below based on how complete a hardware simulation is implemented. (The following terms are not universally-recognized as such, but the underlying concepts are all found in the literature.)

Emulation or simulation

the virtual machine simulates the complete hardware, allowing an unmodified "guest" OS for a completely different CPU to be run. This approach has long been used to enable the creation of software for new processors before they were physically available. Examples include **Bochs**, **PearPC**, PPC version of **Virtual PC**, **QEMU** without acceleration, and the **Hercules emulator**. Emulation is implemented using a variety of techniques, from **state machines** to the use of **dynamic recompilation** on a full virtualization platform.

Encontrar: alejand Siguiente Anterior Resaltar todo Coincidencia de mayúsculas/minúsculas Se alcanzó el principio de la página, se continúa desde el final

Terminado

145.97.39.155

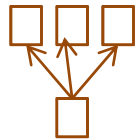
<http://en.wikipedia.org/wiki/Virtualization>



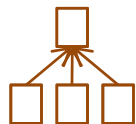
Virtualización en computación

- **Virtualización** es término muy amplio que se refiere a la **abstracción de los recursos de un computador**.
- Una técnica para **ocultar las características físicas** de los **recursos** de computación **de la forma** en la cual **otros** sistemas, aplicaciones o usuarios finales **interactúan** con esos recursos.

– Esto incluye hacer un único recurso físico (como un servidor, un sistema operativo operativo, una aplicación o un dispositivo de almacenamiento) aparentar que funciona como varios recursos lógicos



– Esto también incluye hacer que múltiples recursos físicos (como dispositivos de almacenamiento o servidores) aparezcan como un único recurso lógico.



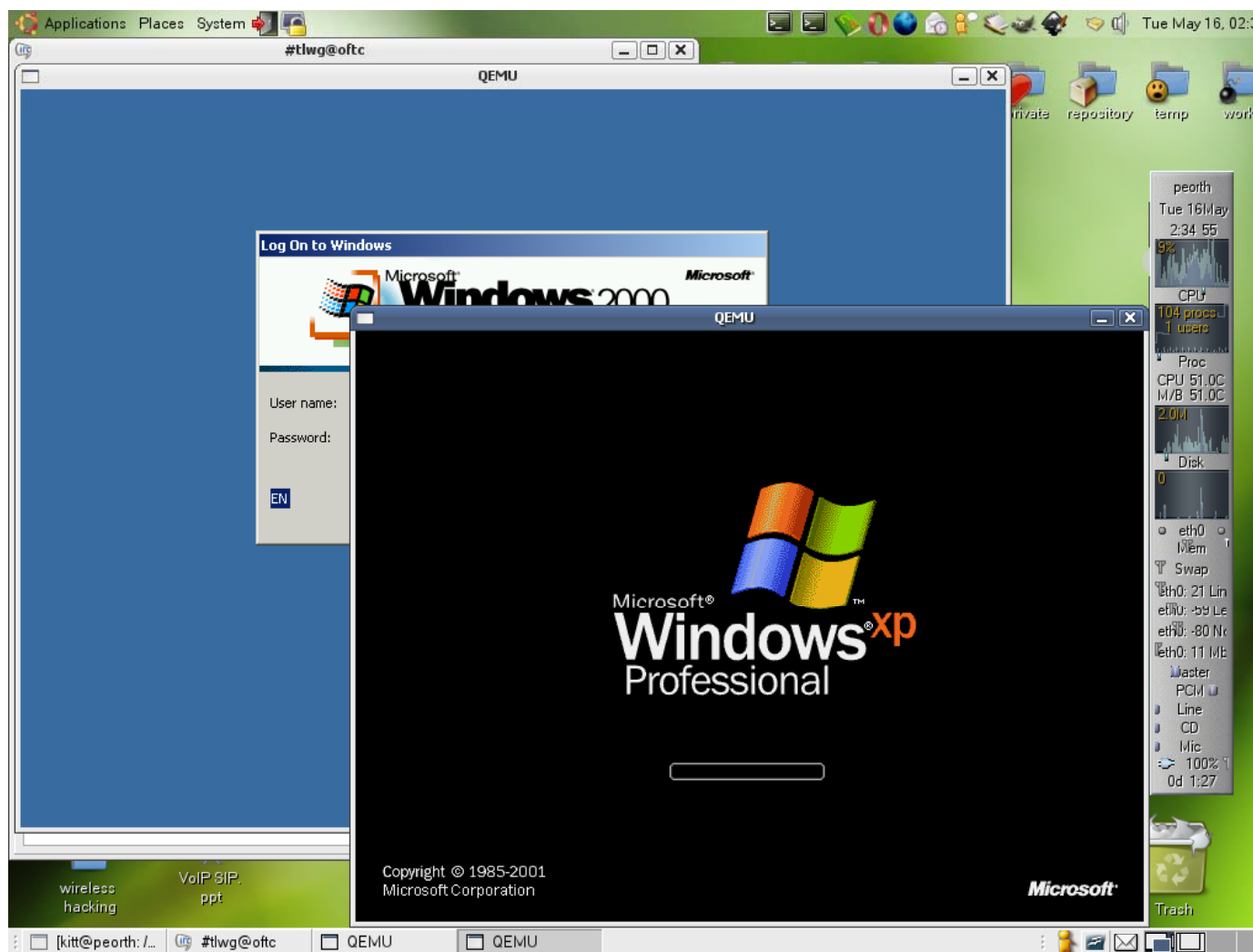
Virtualización de una plataforma

- El término virtualización no es nuevo:
 - Se ha usado desde los años 60
- Se ha aplicado a diferentes aspectos y ámbitos de la computación:
 - Desde un computador completo hasta componentes

- Nos centraremos en la **virtualización de una plataforma** en términos de máquinas virtuales en Linux



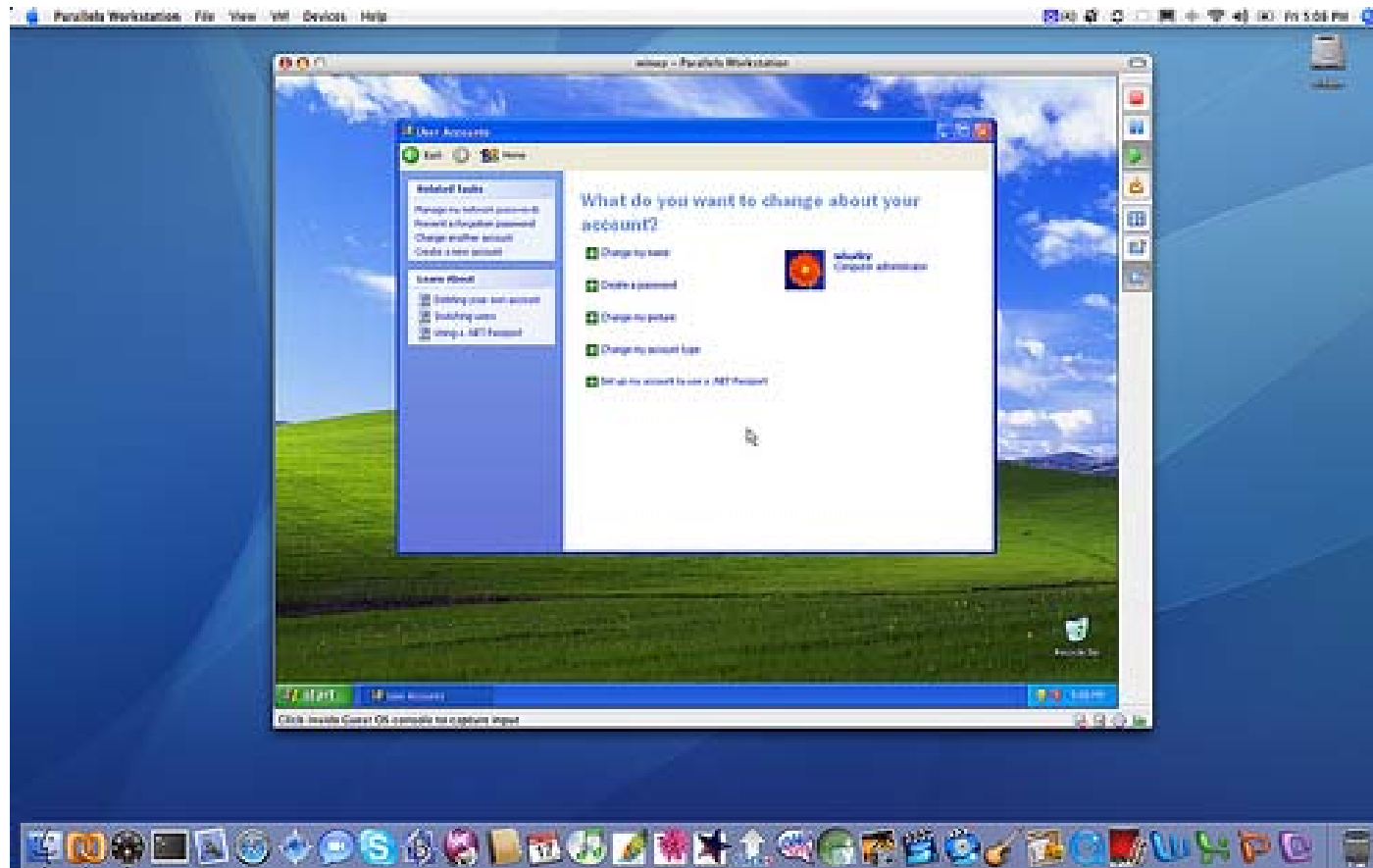
Windows sobre Linux



<http://www.kitty.in.th/files/376/qemu.png>



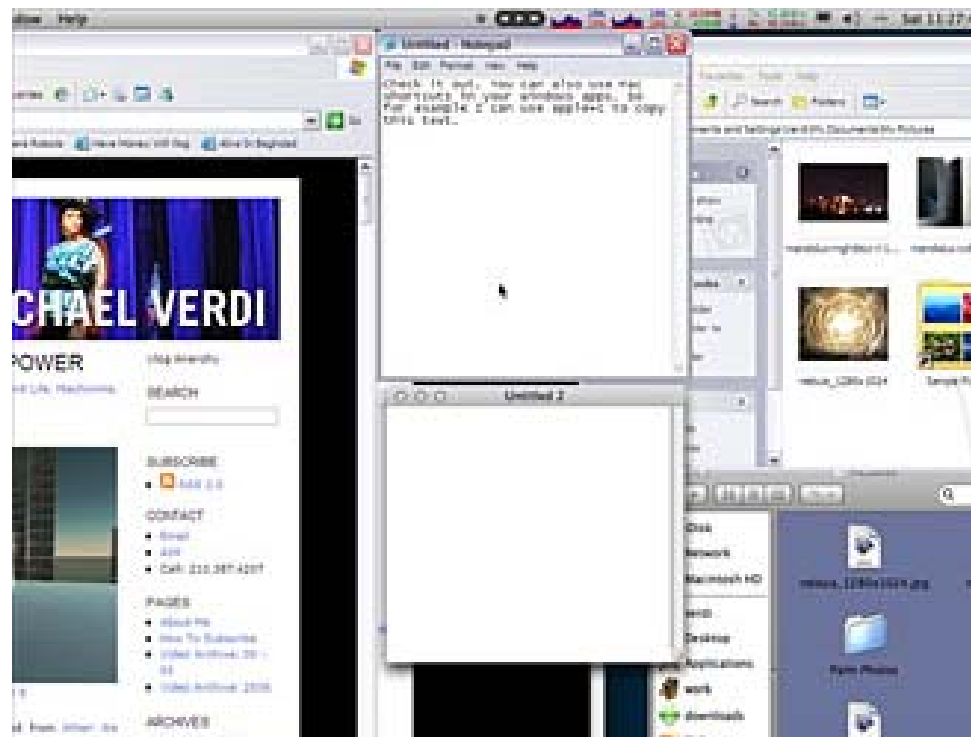
Windows sobre MacOS



<http://redmonk.com/cote/2006/06/21/apples-new-cluefull-test-parallels/>



Windows *sobre* Mac: Coherence mode



<http://michaelverdi.com/index.php/2006/12/02/parallels-screencast/>





¿ES DIFÍCIL USAR LA
VIRTUALIZACIÓN?





No es difícil... si todo funciona

En línea de mandatos

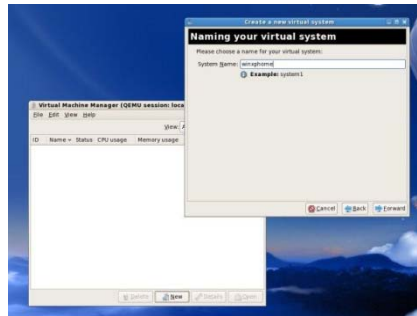


Con interfaz gráfica

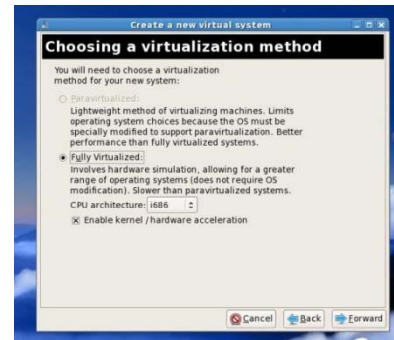


Intefaz gráfica (creación): virt-manager (kvm, xen, ...)

1



2



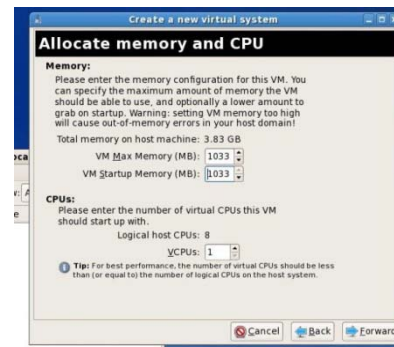
3



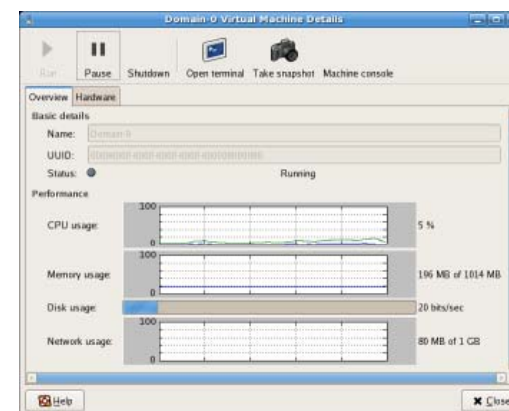
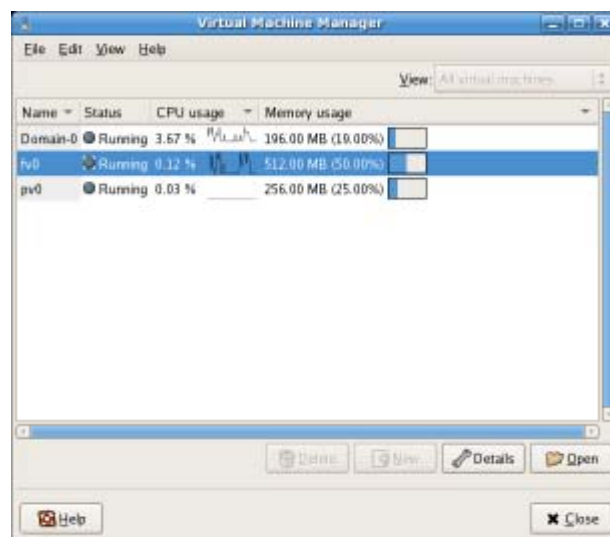
4



5



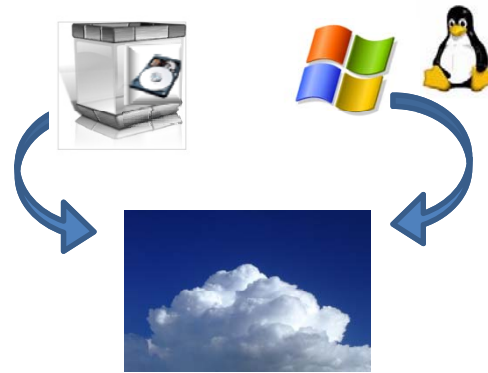
Intefaz gráfica (gestión): virt-manager (kvm, xen, ...)



Línea de mandatos (creación)



2. Crear un disco duro
(o partición) virtual



3. Arrancar la M.V.
para instalar el S.O. en
el disco/partición virtual



4. Arrancar la M.V.
para instalar el S.O. en
el disco/partición virtual





KVM+qemu

1. Disponer de un Linux con soporte KVM:
 - <http://kvm.qumranet.com/kvmwiki>





KVM+qemu

2. Crear la imagen de disco:

```
# qemu-img create -f qcow vdisk.img 10G
```





KVM+qemu

3. Instalar el sistema operativo:

```
# sudo qemu-kvm-x86_64 \  
-hda vdisk.img \  
-cdrom /path/to/boot-media.iso \  
-boot d \  
-m 384 -no-acpi 🇵🇪
```





KVM+qemu

4. Ejecutar el sistema operativo ya instalado:

```
# sudo qemu-kvm-x86_64 \  
-hda vdisk.img \  
-boot d \  
-m 384
```





XEN: fedora (cli)

1. Disponer de un Linux con soporte XEN:
 - <http://www.cl.cam.ac.uk/research/srg/netos/xen/>





XEN: fedora (cli)

2. Crear la imagen de disco:



```
# dd if=/dev/zero of=/xen-vd/fc4.img bs=1024k count=10000
# dd if=/dev/zero of=/xen-vd/fc4-swap.img bs=1024k count=1000

# mkfs.ext3 / xen-vd/fc4.img
# /xen-vd/fc4.img is not a block special device.
# Proceed anyway? (y,n) <-- y
# mkswap /xen-vd/fc4-swap.img
```





XEN: fedora (cli)

3.a. Pre-instalar el sistema operativo:

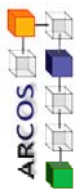


```
# mount -o loop /xen-vd/fc4.img /mnt/vm

# mkdir -p /var/cache/yum
# mkdir -p /var/log
# yum -c http://people.redhat.com/~katzj/yum-xen.conf \
    --installroot=/mnt/vm/ -y groupinstall Base

# umount /mnt/vm
```





XEN: fedora (cli)

<http://people.redhat.com/~katzj/yum-xen.conf>

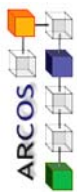


```
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
exclude=*-debuginfo
gpgcheck=0
obsoletes=1
reposdir=/dev/null

[base]
name=Fedora Core 4 - $basearch - Base
mirrorlist=http://fedora.redhat.com/download/mirrors/fedora-core-4
enabled=1

[updates-released]
name=Fedora Core 4 - $basearch - Released Updates
mirrorlist=http://fedora.redhat.com/download/mirrors/updates-released-fc4
enabled=1
```





XEN: fedora (cli)

3.c. /etc/xen/fedora.sxp:

```
name = "fedora"
kernel = "/boot/<kernel>-xenU"
root = "/dev/hda1"
memory = 64
disk = ['file:/xen-vd/fedora.img,hda1,w', 'file:/xen-vd/fedora-
swap.img,hda2,w']

nics=1
dhcp = "off"
ip="192.168.0.103"
netmask="255.255.255.0"
gateway="192.168.0.1"
hostname="fedora.acme.com"
```





XEN: fedora (cli)

4. Ejecutar el sistema operativo ya instalado:

```
# xm create -c /etc/xen/fedora.sxp
```



No es difícil... si todo funciona



Línea de mandatos:

- **Fácil** automatización



Interfaz gráfica:

- **Fácil** creación y gestión





¿PARA QUÉ UTILIZAR LA
VIRTUALIZACIÓN?





Razones por las que usar virtualización

- **CONSOLIDACIÓN DE SERVIDORES**
- **RECUPERACIÓN DE DESASTRES**
- **PRUEBAS O FORMACIÓN**
- **PORTABILIDAD DE APLICACIONES**



Consolidación

- Consolidación de servidores:
 - Ahorro de costes (y mejor aprovechamiento)
 - Simplificación de la administración y gestión



1 000 €



1 00



2 000 €



20 €

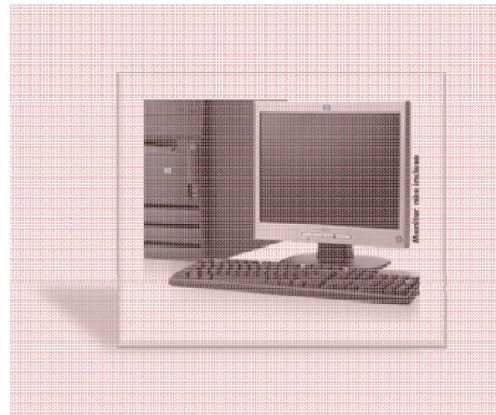


1 000 €



Tolerancia a fallos

- Recuperación de desastres:
 - Ejecución de una máquina en espera
 - Rearranque automático



Entornos de prueba

- Pruebas o formación:
 - Ejecución de otro entorno de trabajo
 - Facilitar restauración del sistema (no inmediato, pero si fácil)

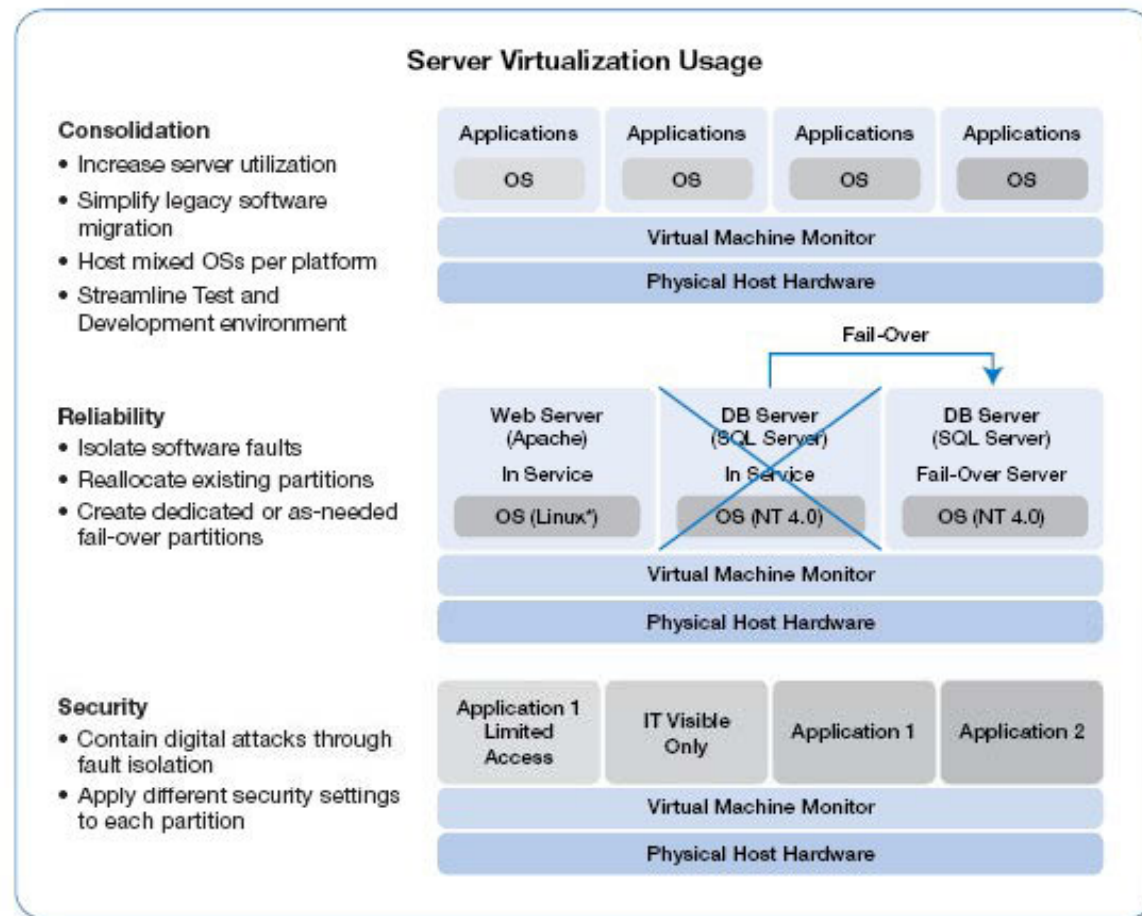


Portabilidad de aplicaciones

- Portabilidad de aplicaciones:
 - Ejecución de aplicaciones del sistema X en el sistema Y
 - Ejecución de aplicaciones del sistema X versión A en versión A+1



Resumen de las principales ventajas



Razones por las que NO usar virtualización

- **DIFICULTAD EN EL DIMENSIONAMIENTO**
- **NECESIDAD DE MAYORES RECURSOS EN UN SOLO ORDENADOR**
- **DOBLE NIVEL DE ADMINISTRACIÓN**
- **PERDIDA DE RENDIMIENTO**



Dimensionamiento

- Necesidad de un correcto dimensionamiento:
 - Los servidores (virtuales) cambian sus necesidades (uso de memoria, cpu, etc.)
 - Un incorrecto dimensionamiento puede tener impacto en todos los servidores



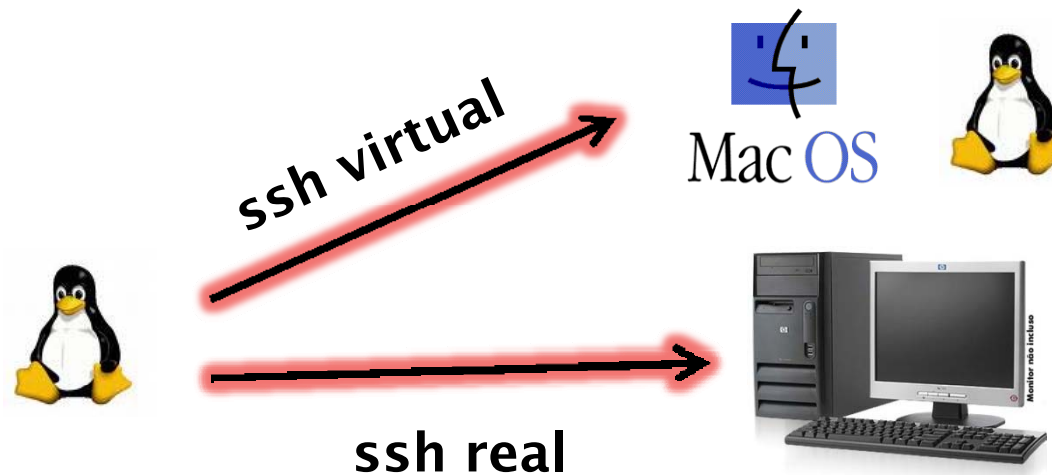
Mayores recursos

- Necesidad de mayores recursos en un solo ordenador:
 - 8 servidores de 1 GB consolidados en 1 servidor de 8 GB
 - No siempre es fácil (y barato) comprar un servidor con n tarjetas de red, n GB de RAM, n TB de disco, etc.



Más administración

- Doble nivel de administración:
 - Administración de las máquinas reales
 - Administración de las máquinas virtuales
 - Monitorización y control de la asociación máquina virtual y máquina real
 - Si una máquina real presenta problemas, entonces es necesario migrar las máquinas virtuales asociadas a ella



Pérdida de rendimiento

- 'Ligera' pérdida de rendimiento:
 - En CPU puede ser poco: entre un 3% y un 12%
 - ¿Tarjeta gráfica?
 - ¿Disco duro compartido entre varias máquinas virtuales?





¿QUÉ SOLUCIONES HAY DE
VIRTUALIZACIÓN?





Tipos de virtualización (punto de vista más técnico)

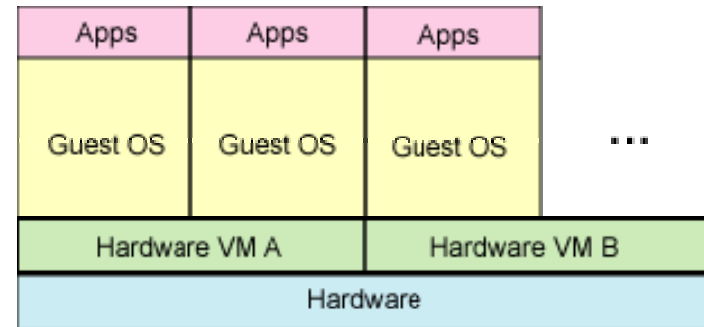
- **Emulación Hardware**
- **Virtualización completa**
- **Paravirtualización**
- **Virtualización a nivel del sistema operativo**





Emulación Hardware

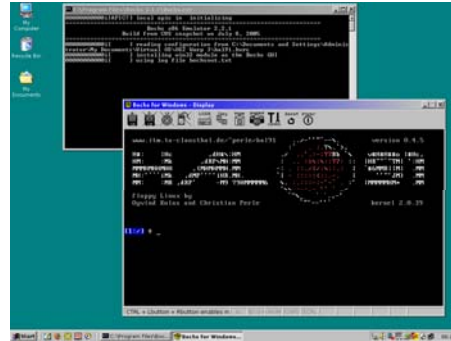
- Se crea una MV en el sistema host para emular el hardware que interesa
- Desventaja: lentitud (del orden de x100)
- Ventaja: se puede ejecutar un sistema operativo para CPU1 en CPU2 sin ser modificado



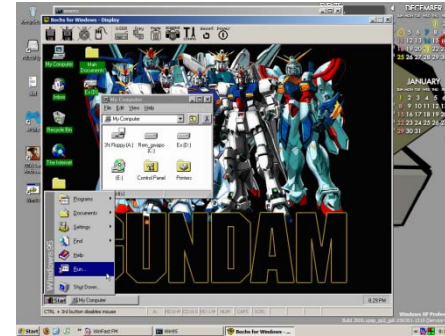
Emulación Hardware



- Bochs



Linux/W95



W95/WXP

- Qemu



W2k/Linux(F)



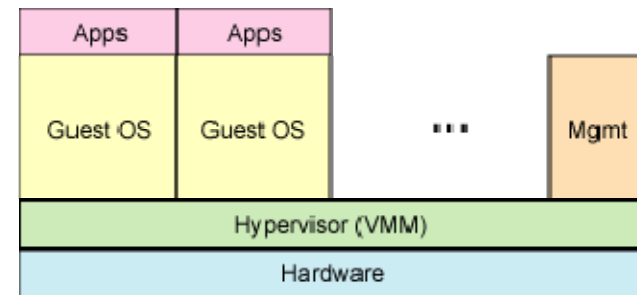
Linux(K)/WXP

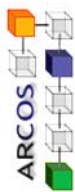




Virtualización completa

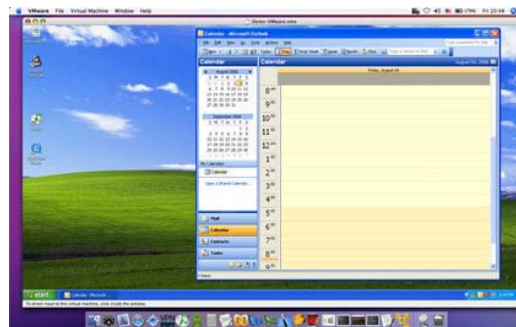
- El hardware es compartido por todos los sistemas operativos invitados (*Guests*) a través del hipervisor
- Ventaja: no se necesita modificar el sistema operativo
- Desventaja: es necesario usar trampas para capturar los accesos del sistema operativo invitado al hardware



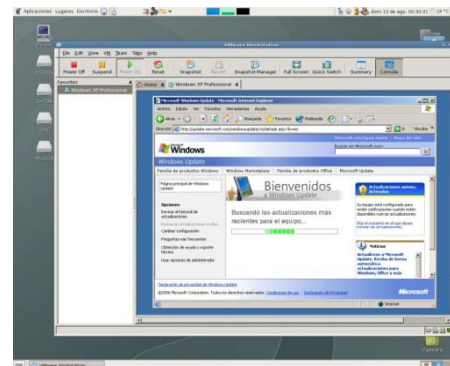


Virtualización completa

- VMware



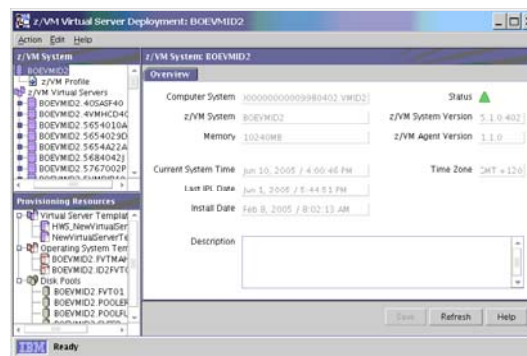
WXP/MacOS



WXP/Linux



System z™
• z/VM



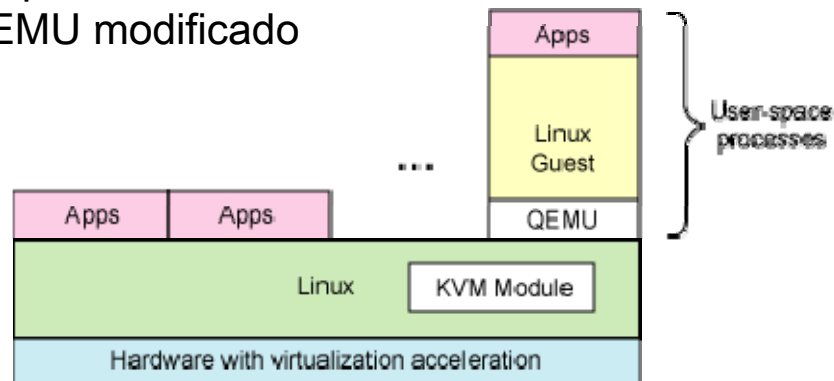
z/Linux sobre z/VM



Virtualización completa

• KVM

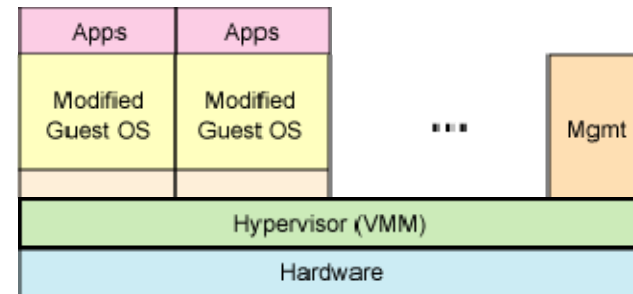
- KVM es la primera solución de virtualización que forma parte ya del kernel (desde el 2.6.20)
- KVM transforma el kernel de linux en un hipervisor usando un módulo de kernel
 - Este módulo permite que otros sistemas operativos invitados puedan ejecutar en espacio de usuario
 - El módulo de kernel KVM ofrece el hardware virtualizado a través del dispositivo /dev/kvm
 - El sistema operativo se comunica con el módulo usando un proceso QEMU modificado





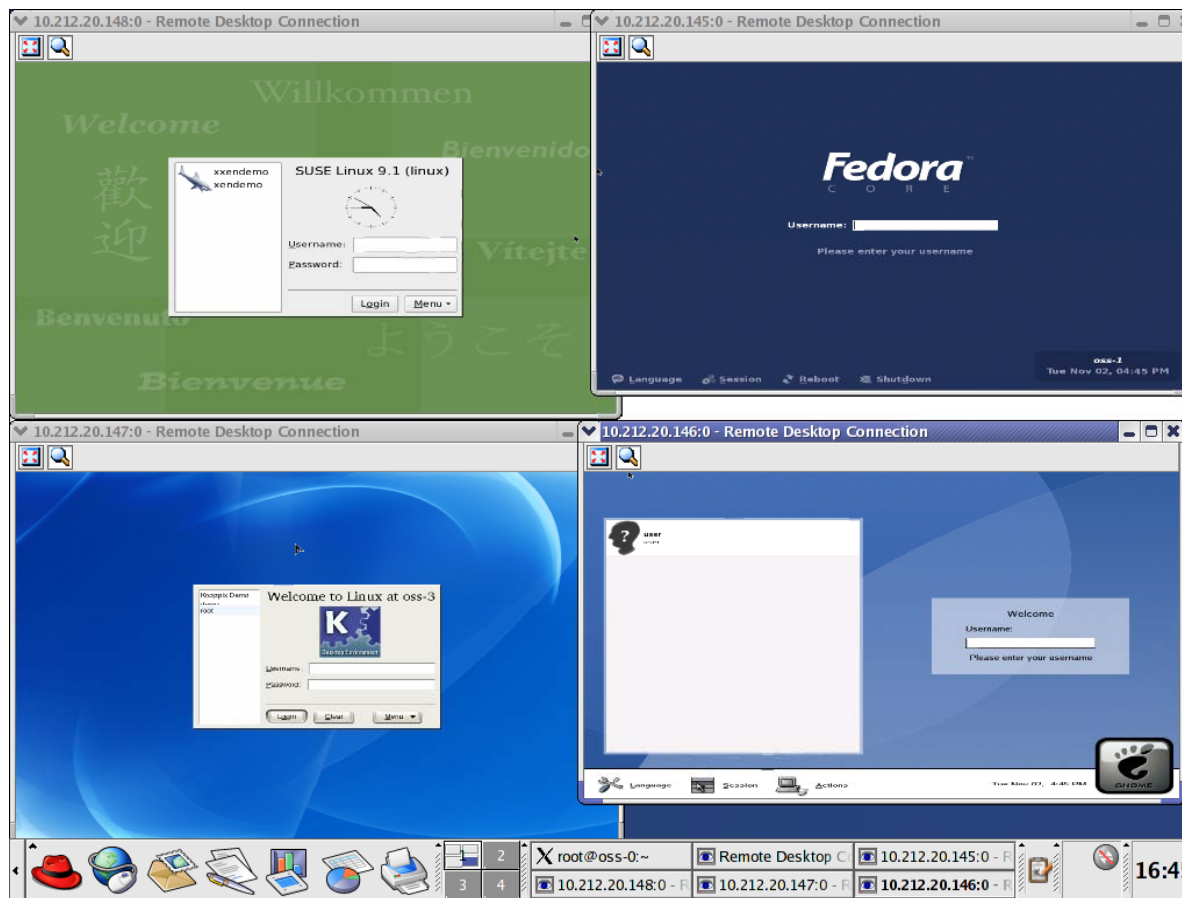
Paravirtualización

- Similar a la virtualización completa
- Desventaja: El sistema operativo ha de modificarse para interactuar con el hipervisor
- Ventaja: El sistema operativo colabora con el hipervisor (el tiempo perdido en trampas para interceptar peticiones se disminuye)



Paravirtualización

- XEN



Linux(S,F,D)/Linux(F)





Paravirtualización

- UML

```

root@resnat72:~# ./logicfas rc,0 rc,6 rc,8 rc,5 rc,custom rc,custom,gz rc,custom" reboot rescuept
reset resolv.conf rm resolv.conf route rsh rshd script scsi scsi_info seagate
sed serial serial_cs services setserial settings.sh shared shutdown slattach
sleep snarf sort split stty swapoff swapon sync tail tar tcic tee telnet telnetd
termcap test tocsrsh tocsrshd tocsrshd.FAQ touch true tune2fs umount unde
undebrm unpack_z unrpem update utap vi vi,help view uc utaplock

...Login as root.

tty$0 tocsrsh login: root
Password: ile rev,2,01
Today is Boottime, the 60th day of The Aftermath in the YOLB 3169
# ifconfig eth0 192.168.5.100
# insmod tun
insmod: tun: no module by that name found
# ifconfig tap2 192.168.5.5 netmask 255.255.255 up
# bash -c echo 1 > /proc/sys/net/ipv4/ip_forward
# route add -host 192.168.5.100 dev tap2
# bash -c echo 1 > /proc/sys/net/ipv4/conf/tap2/proxy_arp
# arp -B 192.168.5.100 tap2 pub
# cat /dev/null > /dev/null
# nc -l -p 1234

Hey it works!
Press return and see if this line gets sent to the other recursive UML...

...Login as root.

tty$0 tocsrsh login: root
Password: ile rev,2,01
Today is Boottime, the 60th day of The Aftermath in the YOLB 3169
# ifconfig eth0 192.168.5.200
# insmod tun
insmod: tun: no module by that name found
# ifconfig tap1 192.168.5.5 netmask 255.255.255 up
# bash -c echo 1 > /proc/sys/net/ipv4/ip_forward
# route add -host 192.168.5.200 dev tap1
# bash -c echo 1 > /proc/sys/net/ipv4/conf/tap1/proxy_arp
# nc -l -p 1234

Hey it works!
  
```

Linux/Linux

SIMULACIÓN DE UN CLUSTER USANDO *USER MODE LINUX*

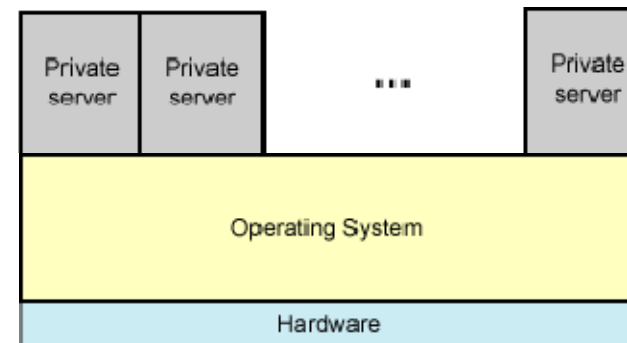
VICTOR INIESTA SAMPAYO





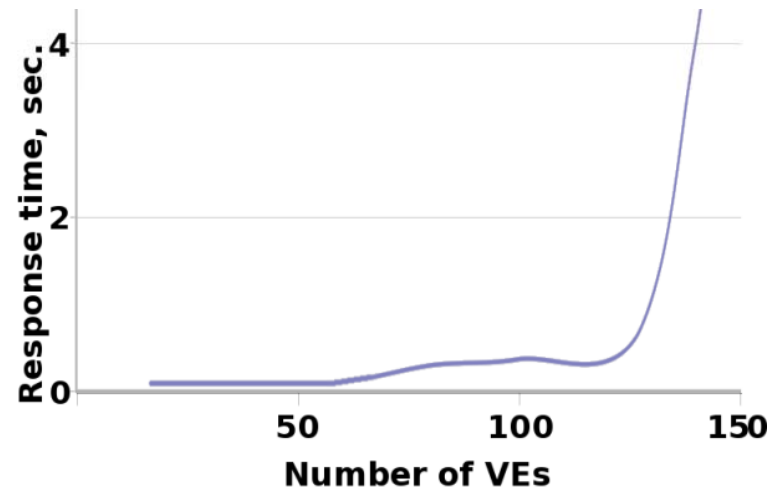
Virtualización a nivel del sistema operativo

- Enfoque totalmente diferente: el sistema operativo en si mismo ofrece servidores virtuales
- Desventaja: NO es posible ejecutar distintos sistemas operativos
- Ventaja: rendimiento



Virtualización a nivel del sistema operativo

- OpenVZ



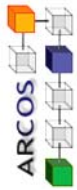
Densidad de OpenVZ en un equipo de 768 Mb ($\frac{3}{4}$ Gb) de RAM



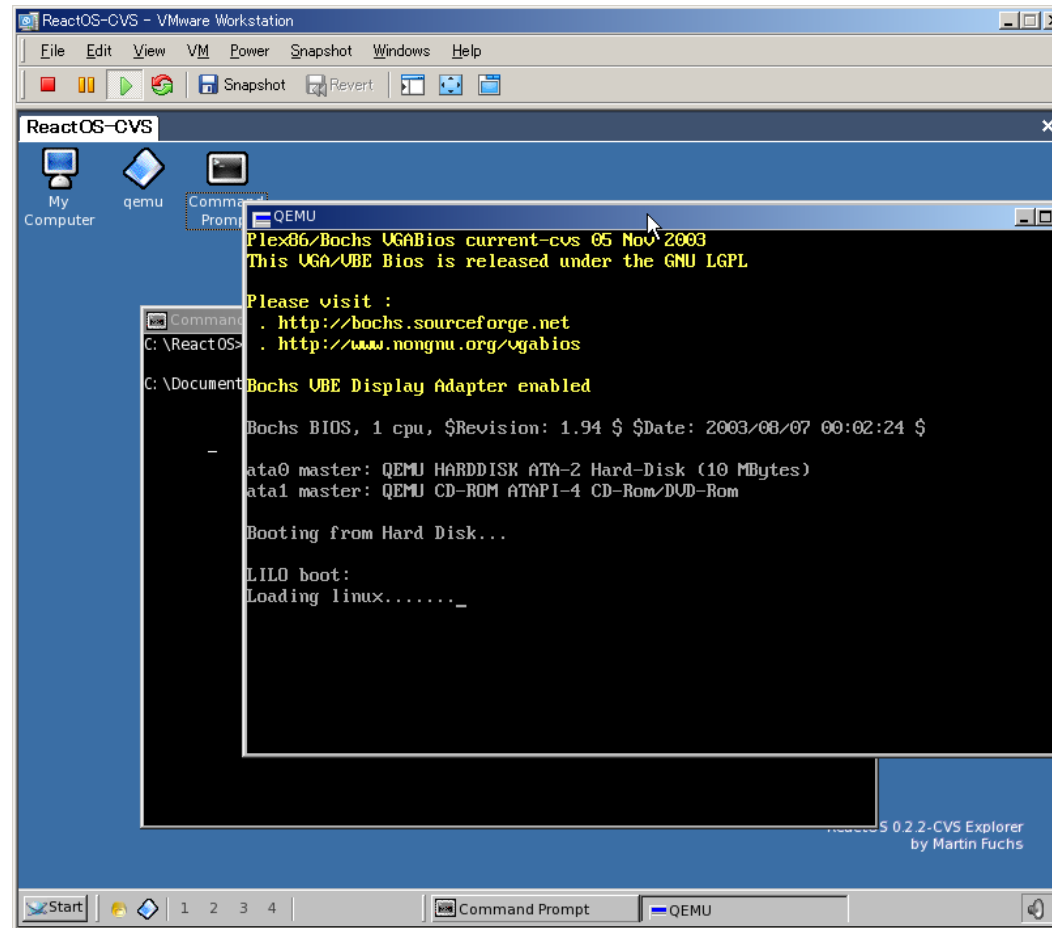


¿QUÉ FACTORES CONSIDERAR
PARA LA VIRTUALIZACIÓN?





Si uno quiere ser algo 'friki'... ☺



Linux en QEMU en ReactOS en VMWare en Window



Tecnología...

- Intel y AMD han desarrollado extensiones para virtualización, no compatibles entre sí pero son similar funcionalidad:
 - Permite que un hipervisor pueda ejecutar un sistema operativo huesped (Guest) con mínima penalización en rendimiento



- Intel ofrece las VT-x como las extensiones IVT para IA-32 (*Vanderpool*) y ofrece VT-i como las IVT para IA-64 (*Silverdale*)
 - En un futuro se unirá Virtualization for Directed I/O



- AMD ofrece AMD-V (*Pacifica*)
 - En un futuro ofrecerá IOMMU (VDIO)



Plataforma...



- Windows

<http://www.microsoft.com/whdc/system/platform/virtual/default.mspix>



- Linux

<http://virt.kernelnewbies.org/>

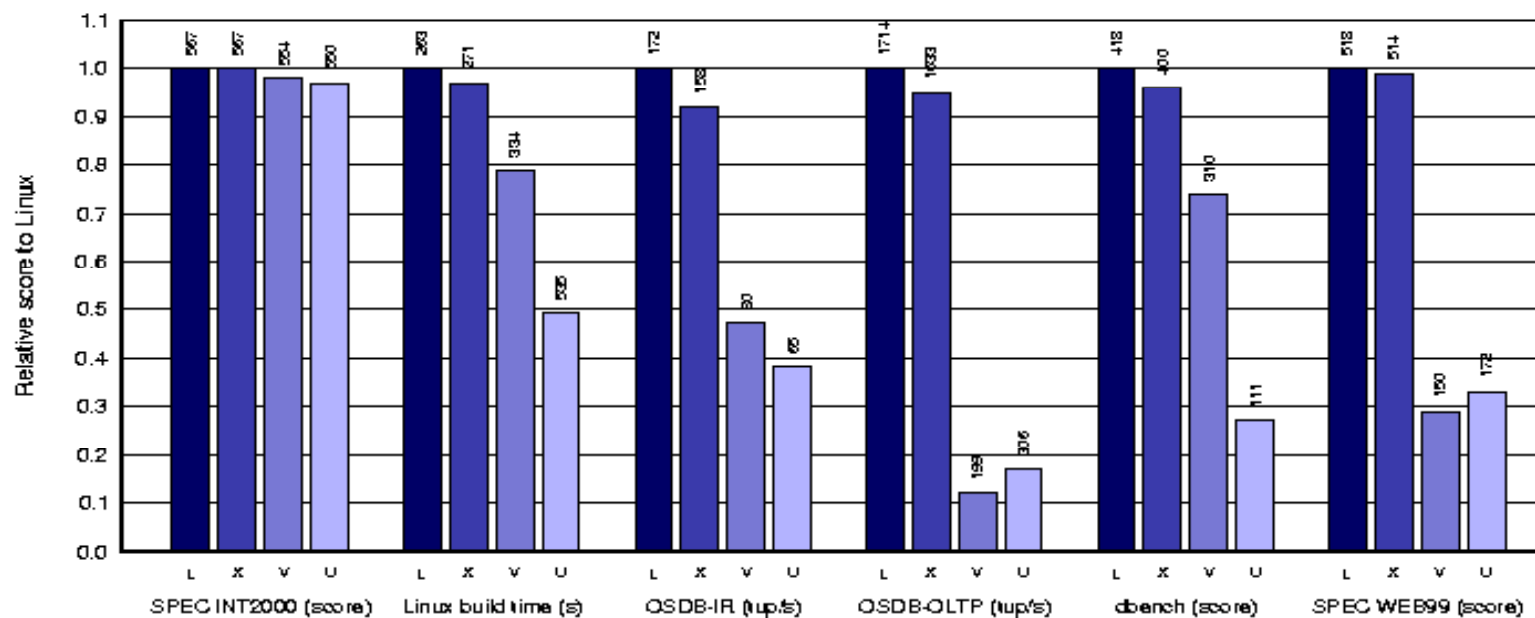


- Mac

<http://www.parallels.com/en/products/workstation/mac/>



Prestaciones: Rendimiento...



- Rendimiento relativo a:
 - (L) Linux nativo
 - (X) Xen/Linux
 - (V) VMware Workstation 3.2
 - (U) User Mode Linux



Comparativa

	full virt	paravirt	containers (OS virt)	license	architectures	performance	SMP guests	CPU / memory hotplug	standalone host	notes
XEN	✓	✓		GPL	i686, x86-64, IA64, PPC	paravirt very fast, full virt medium				full virt needs VT / AMD-V
KVM	✓	✓		GPL	i686, x86-64	paravirt very fast, full virt medium				full and para virt need VT / AMD-V
lguest		✓		GPL	i686	slow/medium				
rhyp		✓		GPL	i686, x86-64, PPC	fast	(?)			research project
MoL	✓			GPL	PPC	fast				32 bit only
UML		✓		GPL	i686, x86-64, PPC	slow				upstream
L4Linux		✓		GPL	i686, ARM	medium				
qemu	✓			GPL	i686, x86-64, IA64, PPC	slow/medium, fast with kQEMU				
OpenVZ			✓	GPL	i686, x86-64, IA64, PPC, SPARC	native				live migration
Linux-VServer			✓	GPL	i686, x86-64, IA64, PPC	native				poor performance isolation
VMware	✓			proprietary	i686, x86-64	medium				
LPAR	✓			proprietary	s390	native				
z/VM	✓	✓		proprietary	s390	very fast				typically runs under LPAR



Resumen (1)

1. ¿Qué es la virtualización?

- Una tecnología que nos puede facilitar el trabajo

2. ¿Es difícil usar la virtualización?

- No es difícil (si todo funciona) puesto que hay muchos 'asistentes'

3. ¿Para qué utilizar la virtualización?

- Para ejecutar varios sistemas operativos (y aplicaciones) simultáneamente pudiendo aprovechar diferentes usos de la máquina



Resumen (2)

1. ¿Qué soluciones hay de virtualización?

1. Hay bastantes y variadas: tipo emulación, tipo virtualización completa, paravirtualización, ...

2. ¿Qué factores considerar para la virtualización?

1. Qué plataformas (procesador, sistemas operativos, etc.) se deséa ejecutar
2. Qué plataforma será base para ejecutar (qué tecnologías se disponen)
3. Como se realizará la administración
4. ...



Virtualización: Introducción y toma de contacto

*GRUPO DE USUARIOS DE LINUX DE LA
UNIVERSIDAD CARLOS III DE MADRID*



ALGUNOS EXTRAS...





XEN: debian (cli)

1. Disponer de un Linux con soporte XEN:
 - <http://www.cl.cam.ac.uk/research/srg/netos/xen/>



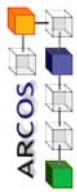


XEN: debian (cli)

2. Crear la imagen de disco:

```
# dd if=/dev/zero of=/xen-vd/debian.img bs=1024k count=10000  
# dd if=/dev/zero of=/xen-vd/debian-swap.img bs=1024k count=1000  
  
# mkfs.ext3 / xen-vd/debian.img  
# /xen-vd/debian.img is not a block special device.  
# Proceed anyway? (y,n) <-- y  
# mkswap /xen-vd/debian-swap.img
```





XEN: debian (cli)

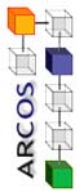
3.a. Pre-instalar el sistema operativo:



```
# mount -o loop /xen-vd/debian.img /mnt/vm
# debootstrap --arch i386 sarge \
    /mnt/vm http://ftp2.de.debian.org/debian

# cp -dpR /lib/modules/<kernel>-xenU /mnt/vm/lib/modules/
# mv /mnt/vm/lib/tls /mnt/vm/lib/tls.disabled
# umount /mnt/vm
```





XEN: debian (cli)

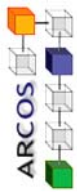
3.b. Instalar el sistema operativo:

```
# mount -o loop /xen-vd/debian.img /mnt/vm
# chroot /vserver/vm_base
# apt-setup
...
Archive access method for apt: <-- http

# cat >> /etc/apt/sources.list
deb http://ftp2.de.debian.org/debian/ stable main
deb-src http://ftp2.de.debian.org/debian/ stable main
deb http://security.debian.org/ stable/updates main
^D
# apt-get update ; base-config ; apt-get upgrade

# exit
# umount /mnt/vm
```





XEN: debian (cli)

3.c. /etc/xen/debian.sxp:

```
name = "debian"
kernel = "/boot/<kernel>-xenU"
root = "/dev/hda1"
memory = 64
disk = ['file:/xen-vd/debian.img,hda1,w', 'file:/xen-vd/debian-
swap.img,hda2,w']

# network
nics=1
dhcp = "off"
ip="192.168.0.102"
netmask="255.255.255.0"
gateway="192.168.0.1"
hostname="debian.acme.com"

extra="3"
```





XEN: debian (cli)

4. Ejecutar el sistema operativo ya instalado:



```
xm create -c /etc/xen/debian.sxp
```

