

Instalación y configuración de Xen 3.0 en Debian GNU/Linux (etch)

Alberto Molina Coballes <alberto.molina@hispalinux.es>
José Domingo Muñoz Rodríguez <josedom24@gmail.com>
IES Gonzalo Nazareno. Dos Hermanas (Sevilla)

17 de mayo de 2007

Resumen

En este documento se describe la instalación y configuración elemental de Xen para la utilización de máquinas virtuales bajo GNU/Linux. En concreto la mayor parte del desarrollo se realiza con la última versión estable de la distribución Debian GNU/Linux, denominada etch. Por último, se incluye una breve discusión sobre los diferentes modos de configuración de las redes virtuales. Este documento se elaboró para el curso *Máquinas virtuales para la puesta en marcha de un portal educativo* organizado por el Centro del Profesorado de Sevilla en Septiembre de 2006.

©Alberto Molina Coballes y José Domingo Muñoz Rodríguez. Algunos Derechos reservados.

Esta obra está bajo una licencia Attribution-ShareAlike 2.5 de Creative Commons. Para ver una copia de esta licencia, visite:

<http://creativecommons.org/licenses/by-sa/2.5/>

Índice

1. Introducción	3
2. Instalación de Xen (dom0)	5
2.1. Instalación de paquetes en Debian etch	5
2.1.1. Instalación del <i>hypervisor</i> de Xen	5
2.1.2. Instalación de un núcleo apropiado	6
2.1.3. Configuración del gestor de arranque (GRUB)	6
2.2. Instalación desde binarios	7
2.3. Configuración de dom0	7
2.4. El fichero <code>/etc/xen/xend-config.sxp</code>	8
3. Instalación de Xen (domU)	9
3.1. Utilización de <code>xen-tools</code>	9
3.2. Inicio del nuevo dominio	12
4. Xenman	14
5. Configuraciones de red	14
5.1. Puente	15
5.2. Encaminador	18
5.3. NAT	20
6. Trucos y demás ...	22

La primera versión de este documento se escribió para el curso *Máquinas virtuales para la puesta en marcha de un portal educativo* organizado por el [Centro del Profesorado de Sevilla](#). Una vez terminado el documento, se publicó con licencia libre con el ánimo de ser útil a los que comienzan en Xen.

Es muy probable que encuentres errores y erratas a lo largo del documento, por lo que los autores te agradecerían cualquier tipo de comunicación al respecto.

- Primera versión: Septiembre de 2006
- Segunda versión: Diciembre de 2006
- Tercera versión: Mayo de 2007

1. Introducción

Según la wikipedia [1]:

Xen es una máquina virtual de código abierto desarrollada por la Universidad de Cambridge. La meta del diseño es poder ejecutar instancias de sistemas operativos con todas sus características, de forma completamente funcional en un equipo sencillo. Xen proporciona aislamiento seguro, control de recursos, garantías de calidad de servicio y migración de máquinas virtuales en vivo. Los sistemas operativos deben ser modificados explícitamente para correr Xen (aunque manteniendo la compatibilidad con aplicaciones de usuario). Esto permite a Xen alcanzar virtualización de alto rendimiento sin un soporte especial de hardware.

Las arquitecturas soportadas actualmente por Xen son x86/32 y x86/64 y la última versión desarrollada es la 3.0.3.

La técnica utilizada por Xen se denomina para-virtualización, lo que consigue comportamientos de las máquinas virtuales cercanos al de una máquina real. Para realizar dicha para-virtualización, es necesario cargar en modo núcleo (*kernel space*) el denominado "hypervisor" que se encarga de la gestión de recursos para los diferentes sistemas operativos de una misma máquina. La contrapartida en el caso de Xen es que el sistema operativo huésped (*guest*) debe modificarse para trabajar con el hypervisor en lugar de con el hardware directamente. A día de hoy sólo pueden trabajar con Xen sistemas operativos libres, ya que en estos casos es factible modificar el núcleo (*kernel*) de forma apropiada¹. Los sistemas operativos "propietarios" deben sacar versiones específicas para trabajar con Xen².

Existe la posibilidad de trabajar con Xen sin necesidad de modificar el sistema operativo huésped, pero para ello es necesario utilizar un microprocesador con tecnología de virtualización. Algunos microprocesadores salidos recientemente al mercado incluyen esta característica, son los conocidos como Intel VT (Vanderpool) y AMD-V (Pacifica). No hablaremos en este documento de la utilización de Xen en este caso que abre muchas más posibilidades, sino que nos ceñiremos al uso de Xen en equipos con microprocesadores convencionales.

Xen tiene diversos usos, aunque destacaremos los dos siguientes:

- Se trata de una excelente herramienta didáctica para el aprendizaje de redes de computadoras, ya que permite al alumno configurar de forma virtual una red completa en un solo equipo.
- Es una alternativa cada vez más real para la utilización de servidores dedicados *virtuales* para empresas que ofrezcan servicios a Internet y quieran

¹En la versión 3.0 está soportado sólo Linux, aunque próximamente lo estarán NetBSD, FreeBSD y Solaris

²Microsoft ha anunciado que su próximo sistema operativo "servidor" podrá actuar como anfitrión de Xen

ahorrar costes sin perder seguridad.

Las diferentes máquinas virtuales que se ejecutan en una máquina reciben el nombre de **dominios** en la terminología de Xen. Existe un dominio privilegiado que es sobre el que se instala el “hypervisor” de Xen y que equivale al sistema operativo anfitrión (*host*) de otros monitores de máquinas virtuales como los de la empresa VMware. Este dominio privilegiado recibe el nombre de **dom0** y el resto de dominios reciben el nombre genérico de **domU**.

El hardware que actualmente puede utilizar un domU en Xen es:

- Interfaz de red
- Disco duro
- Memoria RAM
- Dispositivos de E/S

Para el resto de dispositivos: tarjeta gráfica, disquete, CD-ROM, USB, ACPI, ... hay que elegir el dominio que va a utilizarlo, pero no pueden “compatirse”. Esto puede parecer una gran limitación para determinados usos de las máquinas virtuales, pero no para los aquí mencionados.

En la figura 1 puede verse de forma esquemática los elementos que intervienen así como relación entre los diferentes dominios y el hardware.

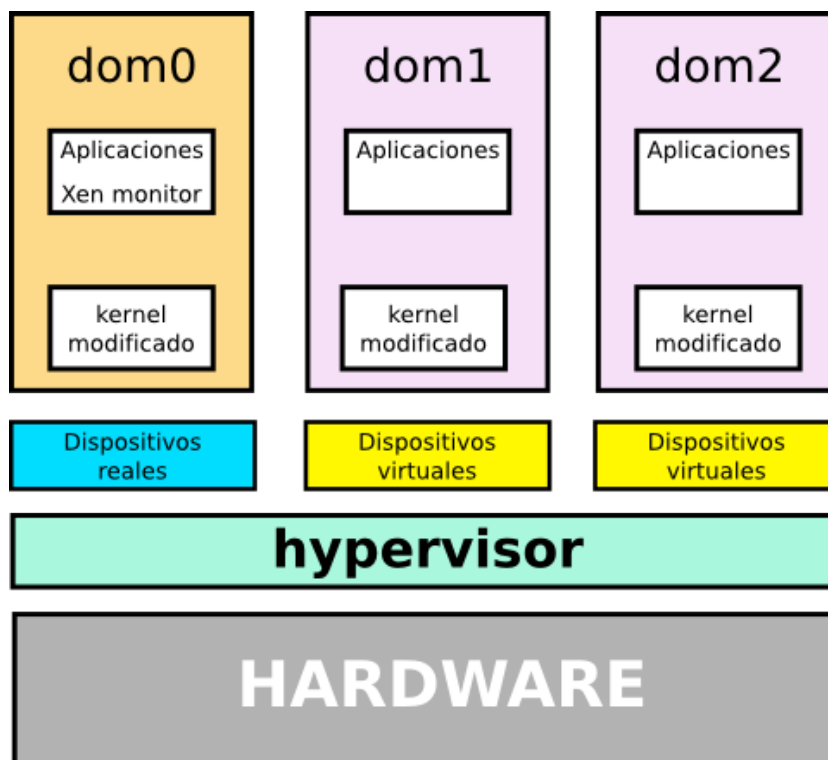


Figura 1: Esquema de funcionamiento del hypervisor y los diferentes dominios en Xen.

Una característica no comentada aún, pero muy destacable en Xen es que se trata de una herramienta libre, licenciada bajo la GPL y otras. Hoy en día se considera la alternativa con mayores posibilidades de hacer frente en el ámbito de la virtualización a los productos de la empresa VMware.

2. Instalación de Xen (dom0)

La instalación del monitor de máquinas virtuales Xen (dom0) comprende tres elementos principalmente:

- El *hypervisor* de Xen, que realiza la paravirtualización.
- Núcleo (kernel) modificado que pueda trabajar sobre un *hypervisor*.
- Herramientas para el manejo de las máquinas virtuales: para crearlas, iniciarlas, etc.

La situación ideal es utilizar una distribución de GNU/Linux que incluya todos los componentes empaquetados, lo que nos garantiza la consistencia de nuestro sistema. Sin embargo también es posible instalar Xen 3.0 en cualquier distribución que cumpla una serie de requisitos, utilizando para ello los binarios que se facilitan en la página del proyecto. En los siguientes apartados se explicaran ambas formas.

En nuestro caso la distribución GNU/Linux utilizada será Debian (etch) que incluye todos los componentes necesarios.

2.1. Instalación de paquetes en Debian etch

Existe la posibilidad de instalar todos los paquetes necesarios simplemente con:

```
aptitude install xen-linux-system-2.6.18-4-xen-686
```

aunque preferimos hacerlo paso a paso, para tratar de ver más claramente el sentido de cada paquete.

2.1.1. Instalación del *hypervisor* de Xen

El paquete que incluye el *hypervisor* es `xen-hypervisor-3.0.3-1-i386`, aunque podría cambiar hasta la estabilización de etch. Lo primero que haremos será instalarlo con:

```
aptitude install xen-hypervisor-3.0.3-1-i386-pae
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
```

```

Leyendo la información de estado extendido
Iniciando el estado de los paquetes... Hecho
Leyendo las descripciones de las tareas... Hecho
Construir la base de datos de etiquetas... Hecho
Se instalarán automáticamente los siguientes paquetes NUEVOS:
  bridge-utils libc6-xen xen-utils-3.0.3-1 xen-utils-common
Se ELIMINARÁN automáticamente los siguientes paquetes:
  libc6-i686
Se instalarán los siguiente paquetes NUEVOS:
  bridge-utils libc6-xen xen-hypervisor-3.0.3-1-i386-pae xen-utils-3.0.3-1
  xen-utils-common
Se ELIMINARÁN los siguientes paquetes:
  libc6-i686
0 paquetes actualizados, 5 nuevos instalados, 1 para eliminar y 0 sin actualizar.
Necesito descargar 1815kB de ficheros. Después de desempaquetar se usarán 2232kB.
¿Quiere continuar? [Y/n/?]

```

El único fichero que instala el paquete del hypervisor (documentación aparte) es:

```
/boot/xen-3.0.3-1-i386-pae.gz
```

que deberemos invocar en primer lugar al iniciar el kernel de xen, ya que se trata del *hypervisor* propiamente, y debe hacer las modificaciones adecuadas en el "kernel space" para soportar la para-virtualización.

2.1.2. Instalación de un núcleo apropiado

Hacemos una búsqueda de estos núcleos en los repositorios con:

```
aptitude search ~nxen~nlinux-image
```

Instalamos el correspondiente al tipo de procesador que tengamos, en nuestro caso:

```
aptitude install linux-image-xen-686
```

Este paquete no es el que incluye el kernel, pero por dependencias nos instala los paquetes que incluyen realmente el kernel y los módulos del mismo.

Podemos observar en la salida de la instalación que se crea de forma adecuada una imagen initrd y que se actualiza GRUB para incluir el nuevo núcleo.

2.1.3. Configuración del gestor de arranque (GRUB)

Si todo ha ido bien, debe aparecer una entrada como la siguiente en el fichero de configuración de GRUB (`/boot/grub/menu.lst`):

```
title Xen 3.0.3-1-i386-pae / Debian GNU/Linux, kernel 2.6.18-4-xen-686
root (hd0,0)
kernel /boot/xen-3.0.3-1-i386-pae.gz
module /boot/vmlinuz-2.6.18-4-xen-686 root=/dev/hda1 ro console=tty0
module /boot/initrd.img-2.6.18-4-xen-686
savedefault
```

Podemos ver que en la entrada `kernel` lo que se incluye realmente es el hypervisor, ya que éste debe cargarse en primer lugar, posteriormente se cargan el kernel y la imagen `initrd`.

2.2. Instalación desde binarios

En el momento de realizar este manual, la última versión de Xen disponible es la 3.0.3. Puede descargarse un archivo comprimido conteniendo todos los elementos necesarios de la página del proyecto³ o de www.xensource.com (previo registro).

En primer lugar hay que tener instalados en el sistema los siguientes paquetes:

- `iproute`
- `bridge-utils`
- `hotplug` (que en las últimas distribuciones viene incluido en el paquete `udev`)

Descargamos el fichero `xen-3.0.3-testing-install-x86_32.3.tgz` y lo descomprimos en un fichero temporal. Accedemos al directorio donde se ha descomprimido y ejecutamos:

```
./install.sh
```

El “hypervisor” se denomina `xen-3.0-unstable.gz` y la imagen del núcleo es `vmlinuz-2.6.16.29-xen` (no necesita `initrd`), por lo que habrá que modificar la entrada para GRUB incluyendo estos elementos.

Esta instalación es algo más sencilla, pero tiene la contrapartida de que no puede desinstalarse limpiamente Xen.

2.3. Configuración de dom0

La configuración elemental del `dom0` es independiente de la forma de instalación.

³<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/downloads/>

TLS El rendimiento de Xen se puede ver mermado si el kernel que manejamos hace uso de "Thread Local Storage"(TLS), por lo que es conveniente deshabilitarlo con:

```
mv /lib/tls /lib/tls.disabled
```

tal como se recomienda en el manual de usuario de Xen [2]. Debemos tener en cuenta que cada vez que se actualice el paquete libc6 (libc6-xen), volverá a activarse TLS, por lo que deberemos repetir la acción anterior.

Limitación de memoria para dom0 En algunos sistemas el reparto de memoria entre dom0 y el resto de dominios da problemas. Si al crear un nuevo dominio se obtiene un mensaje que nos informa de que no hay memoria disponible, se puede solucionar limitando la memoria para dom0. Esto se realiza mediante un parámetro al arrancar el núcleo. En particular se modificaría la línea que invoca al kernel de Xen (en /boot/grub/menu.lst), por ejemplo para limitar la memoria a 256 MB:

```
kernel /boot/xen-3.0.3-1-i386-pae.gz dom0_mem=256M
```

Dispositivos loop El número máximo de dispositivos tipo loop que existen en el sistema es por defecto ocho, puesto que cada máquina virtual que instalemos va a usar dos (en caso de que estén en un fichero y no en una partición), este valor se suele quedar corto, por lo que hay que aumentarlo.

El controlador loop puede estar compilado dentro del núcleo o —como es en nuestro caso— compilado como módulo. En el primer caso se puede aumentar el número de dispositivos loop añadiendo el parámetro `max_loop=128` en la línea `kernel` del fichero de configuración de GRUB. En el caso de que esté compilado como módulo, debemos incluir `loop` en el fichero `/etc/modules`, y creamos el fichero `/etc/modprobe.d/loop` con la línea:

```
options loop max_loop=128
```

para permitir montar hasta 128 dispositivos en este modo.

2.4. El fichero /etc/xen/xend-config.sxp

El aspecto principal que se configura en este fichero es la configuración de red. El único cambio que haremos en este momento es dejar descomentadas las siguientes líneas:

```
(network-script network-bridge)
(vif-script vif-bridge)
(dom0-min-mem 196)
(dom0-cpus 0)
```


Para que dom0 funcione en modo puente. Esto será posible si disponemos de varias direcciones IP libres para los siguientes dominios —por ejemplo porque nos encontramos en una red local con direcciones privadas—.

Tras configurar el dom0, habrá que **reiniciar con el nuevo núcleo** y comprobar que Xen está corriendo de forma adecuada.

Si todo funciona correctamente debe observarse durante el inicio del sistema los mensajes de Xend y la creación del puente.

En la sección 5 se verán las diferentes opciones básicas que hay en Xen para organizar la red virtual entre los diferentes dominios.

3. Instalación de Xen (domU)

Un nuevo dominio en Xen puede construirse de diversas maneras, ya que una de las grandes ventajas de Xen es que utiliza dominios que son directamente sistemas de ficheros (en particiones o ficheros (imágenes)) y que por tanto pueden modificarse montándolos en un directorio de nuestro sistema de ficheros.

En muchos casos es útil crear una imagen —con el método que se desee— y utilizar ésta como patrón para crear todos los nuevos dominios con pequeñas modificaciones.

3.1. Utilización de xen-tools

El paquete xen-tools incluye una serie de scripts para facilitar la creación de nuevos dominios, aunque sólo permite instalar algunas distribuciones GNU/Linux. Este paquete depende de debootstrap, el cual se instalará también.

Editamos el fichero `/etc/xen-tools/xen-tools.conf` y modificamos las siguientes líneas (Suponiendo que estamos en una red local 10.0.0.0/24):

```
#
# Directorio por defecto para las imágenes
#

dir = /vm

#
# Método para obtención de los paquetes del nuevo dominio
#

debootstrap = 1

#
# Opciones de disco y tamaño
```

```

#

size      = 4Gb          # Disk image size.
memory    = 64Mb         # Memory size
swap      = 128Mb        # Swap size
fs         = ext3         # use the EXT3 filesystem for the disk image.
dist      = etch         # Default distribution to install.
image     = sparse       # Specify sparse vs. full disk images.

#
# Configuración de red
#

gateway   = 10.0.0.254
netmask   = 255.255.255.0

#
# Núcleo para el nuevo dominio
#
kernel    = /boot/vmlinuz-2.6.18-4-xen-686
initrd    = /boot/initrd.img-2.6.18-4-xen-686

#
# Réplica a utilizar por debootstrap
#
mirror    = ftp://ftp.rediris.es/debian/

```

En este fichero incluimos todos los parámetros que sean comunes para todas las imágenes que vamos a crear y en la línea de instrucciones añadiremos los parámetros específicos de cada dominio. El fichero va autocomentado y lo único que es necesario aclarar es el método de instalación. Se utiliza la herramienta `debootstrap` que instala un sistema Debian a través de la red. Esta aplicación suele usarse para construir entornos enjaulados (*chroot jail*) y es apropiada para la creación de dominios nuevos con Xen.

Una vez configurado `xen-tools`, ejecutamos la siguiente instrucción:

```
xen-create-image --hostname brun --ip 10.0.0.2
```

Donde especificamos el nombre y la dirección IP del nuevo dominio.

La situación ideal es cuando instalamos un dominio que corresponde con la misma versión que el `dom0`, ya que `debootstrap` en lugar de utilizar el repositorio remoto, simplemente copiará los paquetes de la caché de `dom0` al `domU`, para luego terminar de configurar el nuevo dominio. En nuestro caso utilizamos un `domU` que es también Debian `etch`.

La salida producida por la instrucción anterior es algo como:

General Infomation

```
-----  
Hostname      : brun  
Distribution  : etch  
Filesystem Type : ext3
```

Size Information

```
-----  
Image size    : 4Gb  
Swap size     : 128Mb  
Image type    : sparse  
Memory size   : 64Mb  
Kernel path  : /boot/vmlinuz-2.6.18-4-xen-686  
initrd path   : /boot/initrd.img-2.6.18-4-xen-686
```

Networking Information

```
-----  
IP Address 1  : 10.0.0.2  
Netmask       : 255.255.255.0  
Gateway       : 10.0.0.254
```

Creating swap image: /vm/domains/brun/swap.img
Done

Creating disk image: /vm/domains/brun/disk.img
Done

Creating ext3 filesystem on /vm/domains/brun/disk.img
Done

Installing your system with debootstrap mirror ftp://ftp\
.rediris.es/debian/
Done

Running hooks
Done

No role script specified. Skipping

Creating Xen configuration file
Done
All done

Logfile produced at:
/var/log/xen-tools/brun.log

Que genera el fichero de configuración /etc/xen/brun.cfg.

La imagen y el fichero de swap del nuevo dominio están en:

```
/vm/domains/brun/
```

Tal y como podemos comprobar:

```
ayla:/vm/domains/brun# ls -lh
total 2,2G
-rw-r--r-- 1 root root 4,0G 2006-08-31 22:05 disk.img
-rw-r--r-- 1 root root 128M 2006-08-31 22:05 swap.img
```

Puesto que el fichero `disk.img` consiste en la imagen de un sistema de ficheros puede montarse y comprobarse su contenido, haciendo un simple:

```
mount -o loop /vm/domains/brun/disk.img /mnt
```

Y comprobando el contenido del directorio montado, vemos que tiene la estructura completa de un sistema GNU/Linux, aunque de tamaño reducido:

```
ayla:/mnt# ls
bin  dev  home  lib          media  opt   root  srv  tmp  var
boot etc  initrd lost+found  mnt    proc  sbin  sys  usr
ayla:/mnt# du -hs /
262M  /
```

3.2. Inicio del nuevo dominio

Hasta ahora sólo hemos creado la imagen del nuevo dominio —y el fichero de swap que va a utilizar—, pero dicho dominio todavía no está funcionando como una máquina virtual. Para iniciar dicha máquina virtual utilizamos la instrucción `xm` (Xen management user interface):

```
xm create brun.cfg
Using config file "/etc/xen/brun.cfg".
Started domain brun
```

donde `brun.cfg` se corresponde con un fichero de configuración que esté en el directorio `/etc/xen` y que en este caso ha sido generado por `xen-create-image` y tiene el siguiente contenido:

```
#
# Configuration file for the Xen instance brun, created on
# Thu Aug 31 19:54:40 2006.
#
```

```

#
# Kernel + memory size
#
kernel = '/boot/vmlinuz-2.6.18-4-xen-686'
ramdisk = '/boot/initrd.img-2.6.18-4-xen-686'
memory = '64'

#
# Disk device(s).
#
root = '/dev/sda1 ro'
disk = [ 'file:/vm/domains/brun/disk.img,sda1,w',
         'file:/vm/domains/brun/swap.img,sda2,w' ]

#
# Hostname
#
name = 'brun'

#
# Networking
#
vif = [ 'ip=10.0.0.2' ]

#
# Behaviour
#
on_poweroff = 'destroy'
on_reboot = 'restart'
on_crash = 'restart'

```

Para comprobar que el nuevo dominio está realmente funcionando podemos escribir:

```

xm list

```

Name	ID	Mem (MiB)	VCPUs	State	Time (s)
Domain-0	0	332	1	r-----	143.0
brun	1	64	1	-----	118.2

Para abrir una sesión en el nuevo dominio haremos:

```

xm console brun

```

Puesto que no hemos definido contraseña, podremos entrar la primera vez sin ella —como usuario `root`—, pero a continuación deberemos ejecutar `passwd` y poner una nueva y segura contraseña.

Otros parámetros útiles de `xm` para manejar los dominios son:

- `pause dom-id` — El dominio `dom-id` deja de estar disponible
- `unpause dom-id` — El dominio `dom-id` vuelve a estar disponible
- `destroy dom-id` — El dominio `dom-id` se “desenchufa”. Para levantarlo de nuevo hay que usar “`xm create`”.
- `shutdown dom-id` — Para el dominio `dom-id` correctamente. Utilizado por los scripts para parar todos los dominios cuando se hace “`shutdown`” sobre `dom0`.
- `reboot dom-id` — Reinicia el dominio `dom-id`.

Los parámetros útiles para el `dom0` son:

- `top` — Monitoriza el estado de los dominios
- `dmesg` — Muestra los mensajes del hypervisor al arrancar —equivalente a `dmesg` para el núcleo—.
- `info` — Muestra la información relativa a Xen del núcleo.

4. Xenman

Existe una nueva herramienta que nos permite el manejo de las máquinas virtuales a través de una GUI similar a VMware Console. Para instalar esta aplicación, basta con hacer:

```
aptitude install xenman
```

Podemos ver una captura de pantalla de `xenman` en la figura 2.

5. Configuraciones de red

Lo explicado en esta sección puede encontrarse de forma más extensa en la parte relativa a redes del wiki de Xen [13].

Existen tres configuraciones elementales de red en Xen:

- Puente (`bridge`)
- Encaminador (`router`)

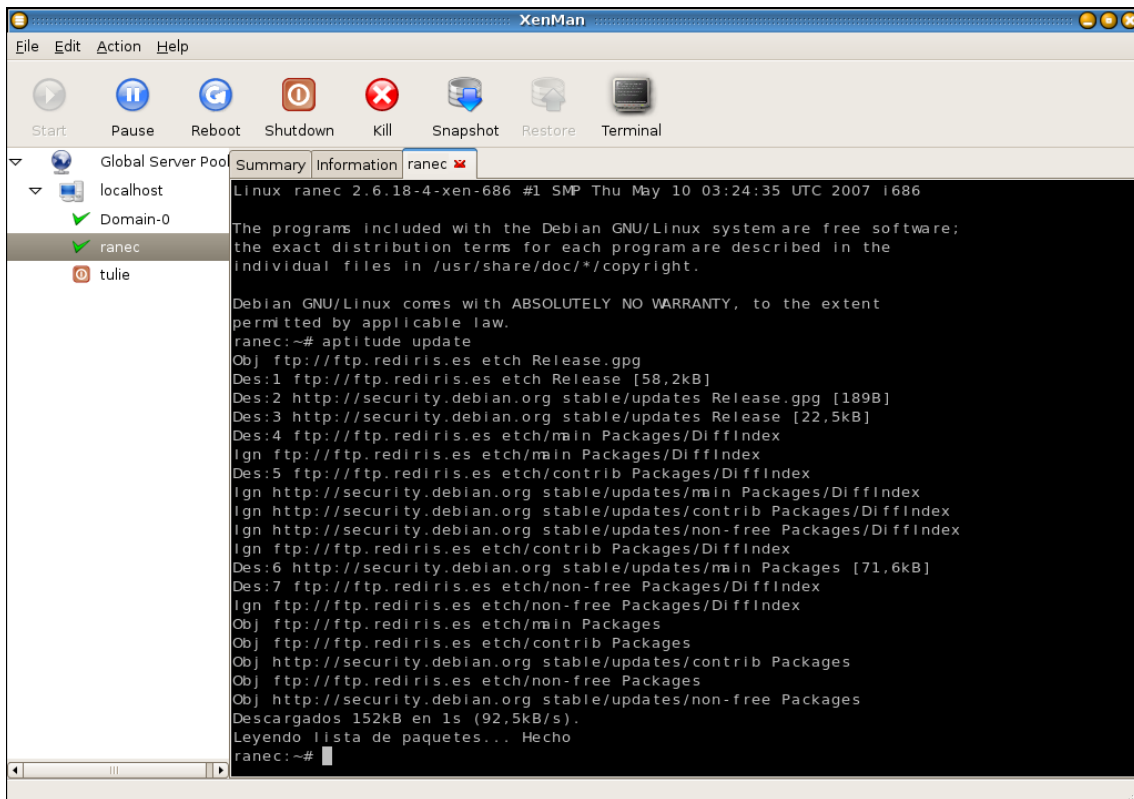


Figura 2: Manejo de una máquina virtual a través de Xenman

- NAT

Pueden construirse configuraciones mucho más complejas a partir de éstas, pero en principio hay que decidirse por utilizar una de ellas, modificando de forma adecuada el fichero `/etc/xen/xen-config.sxp`

5.1. Puente

Para utilizar un puente debemos tener desmomentadas las siguientes líneas en el fichero de configuración de Xen:

```
(network-script network-bridge)
(vif-script vif-bridge)
```

Los pasos que en realidad se realizan son:

Cuando se lanza `xend`, se ejecuta el script `network-bridge`, que:

- Crea un nuevo puente llamado `xenbr0`.
- La interfaz *real* `eth0` se echa abajo.
- Se copian la dirección IP y MAC de `eth0` a la interfaz virtual `veth0`.

- Se renombra la interfaz real como `peth0`
- Se renombra la interfaz virtual como `eth0`
- `peth0` y `vif0.0` se conectan al puente `xenbr0`.
- Se levantan `xenbr0`, `peth0`, `eth0` y `vif0.0`.

Cuando se inicia un `domU`, `xend` (que está corriendo en `dom0`) ejecuta el script `vif-bridge`, que:

- Conecta `vif<id#>.0` to `xenbr0`
- Levanta `vif<id#>.0`

Donde `id#` es el número de identificación del `domU`.

En la figura 3 puede verse de forma esquemática como quedan conectados los dominios tras configurar un puente en Xen con un `domU`.

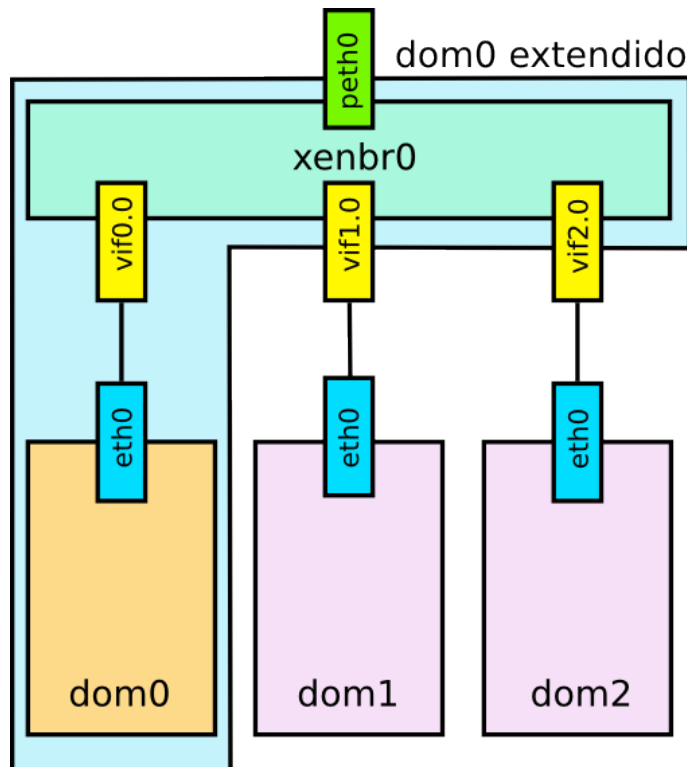


Figura 3: A través del puente se conecta un `domU` a la misma red que el `dom0`, utilizando su interfaz de red

`dom0` es un dominio como otro cualquiera con una interfaz de red (`eth0`) conectada al puente virtual (`xenbr0`). El resto de equipos tienen la misma configuración de red y pertenecen al mismo segmento de red que `dom0`. La imagen de máquinas reales más parecida sería la de varios equipos conectados a un conmutador (*switch*) dentro del mismo segmento de red.

La salida de `ifconfig` en este caso es (obviando el dispositivo de loopback (127.0.0.1)):

```
eth0      Link encap:Ethernet  HWaddr 00:40:D0:DD:B2:84
          inet addr:192.168.0.68 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1248 (1.2 KiB)  TX bytes:810 (810.0 b)

peth0     Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1674 (1.6 KiB)  TX bytes:3450 (3.3 KiB)
          Interrupt:3 Base address:0xc000

vif0.0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:810 (810.0 b)  TX bytes:1338 (1.3 KiB)

xenbr0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          inet6 addr: fe80::200:ff:fe00:0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1142 (1.1 KiB)  TX bytes:468 (468.0 b)
```

Donde podemos observar que la interfaz de red real `peth0` no tiene una dirección MAC real, la tiene `eth0`. Cuando se le añade un nuevo dominio, la interfaz `eth0` del nuevo dominio tendrá una dirección MAC real (con Xen como fabricante). Es decir, el comportamiento conjunto de todo es como varias máquinas reales conectadas a un switch, con `peth0` equivalente a la conexión de salida del switch.

Esta configuración es útil por ejemplo cuando la máquina se encuentra en una red local y podemos utilizar todas las direcciones IP que queramos dentro del mismo rango.

5.2. Encaminador

Para que Xen se comporte como un encaminador, hay que descomentar las líneas:

```
(network-script network-route)
(vif-script      vif-route)
```

Cuando se lanza xend, se ejecuta el script `network-route`, que:

- Activa el bit de forward en `dom0`⁴

Cuando se inicia un domU, xend ejecuta (dentro de `dom0`) `vif-route`, que:

- Copia la dirección IP de `eth0` a `vif<id#>.0`
- Levanta `vif<id#>.0`
- Añade una ruta estática en el domU, apuntando a la interfaz `vif<id#>.0`

En la figura 4 puede verse de forma esquemática como quedan conectados los dominios tras configurar el `dom0` como un encaminador. La salida de `ifconfig`

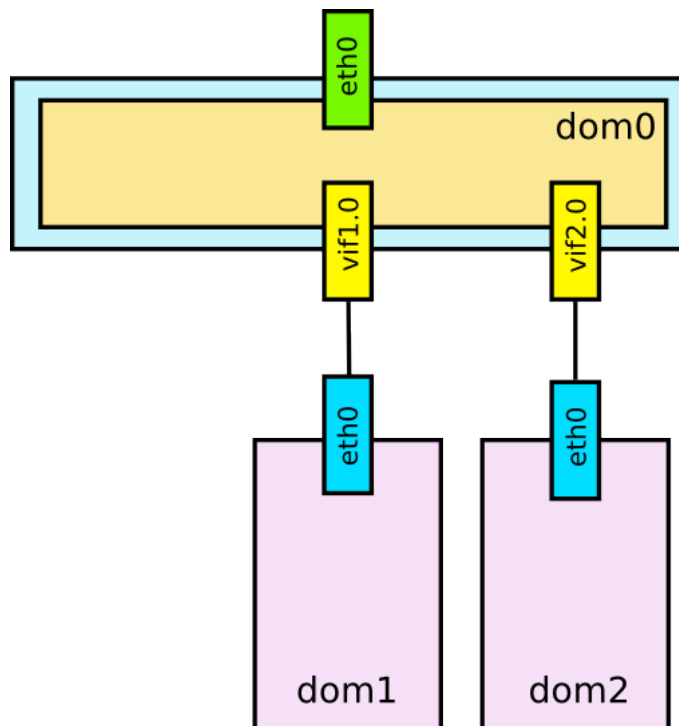


Figura 4: A través de un encaminador se conecta un domU a la misma red que el `dom0`, pero a nivel de red

en el `dom0` es (obviando el dispositivo de loopback (127.0.0.1):

⁴/proc/sys/net/ipv4/ip_forward

```

eth0   Link encap:Ethernet  HWaddr 00:40:D0:65:B2:84
       inet addr:192.168.0.64 Bcast:192.168.0.255 Mask:255.255.255.0
       inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:73 errors:0 dropped:0 overruns:0 frame:0
       TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:6681 (6.5 KiB)  TX bytes:4766 (4.6 KiB)
       Interrupt:3 Base address:0xc000

vif1.0 Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
       inet addr:192.168.0.64 Bcast:192.168.0.255 Mask:255.255.255.255
       inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:1719 errors:0 dropped:0 overruns:0 frame:0
       TX packets:1726 errors:0 dropped:19 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:93686 (91.4 KiB)  TX bytes:131506 (128.4 KiB)

```

y la de route -n:

```

Kernel IP routing table
Destination      Gateway          Genmask          Flags   Iface
192.168.0.100    0.0.0.0         255.255.255.255 UH      vif1.0
192.168.0.0     0.0.0.0         255.255.255.0   U      eth0
0.0.0.0         192.168.0.1    0.0.0.0         UG      eth0

```

Donde 192.168.0.100 es la dirección IP asignada al domU.

En el domU hay que poner como gateway el mismo que tenga el dom0, en este caso 192.168.0.1

Cuando las tramas salen por eth0, tienen la misma dirección MAC, para todas las direcciones IP. Dependiendo de la configuración del equipo esto puede ser un problema, que puede solucionarse incluyendo en el script network-route la línea:

```
echo 1 >/proc/sys/net/ipv4/conf/eth0/proxy_arp
```

Desde un equipo externo podremos comprobar que aparecen dos direcciones IP asociadas a la misma dirección MAC:

```
ranec:~# cat /proc/net/arp
```

```

IP address      HW address      Mask            Device
192.168.0.64    00:40:D0:65:B2:84  *              eth1
85.136.XX.X     00:05:00:E3:4D:DB  *              eth0
192.168.0.100   00:40:D0:65:B2:84  *              eth1

```

Es decir, cuando disponemos de varias direcciones IP dentro del mismo rango, tendremos que optar por utilizar una configuración de puente o una de enrutador, dependiendo de si el equipo al que estamos conectando nos acepta o no diferentes direcciones MAC, para las diversas direcciones IP. En el primer caso trabajaremos en el nivel de enlace de la pila TCP/IP y en el segundo caso en el nivel de red.

5.3. NAT

No existe documentación oficial sobre la configuración de redes con NAT en Xen, así que lo que se explica a continuación es fruto simplemente de nuestra experiencia al configurarlo. Pueden existir por tanto, errores o inexactitudes.

Para que Xen utilice NAT, hay que descomentar las líneas:

```
(network-script network-nat)
(vif-script      vif-nat)
```

En este caso, el dom0 estará conectado a dos subredes diferentes, típicamente a una con direcciones públicas en `eth0` y otra con direcciones privadas en `vif<#?.0>`. A través del mecanismo de traducción de direcciones NAT, irá cambiando las direcciones origen/destino de cada paquete de forma adecuada.

Cuando se lanza `xend`, se ejecuta el script `network-nat`, que:

- Activa el bit de `forward` en `dom0`

Cuando se inicia un `domU`, `xend` ejecuta (dentro de `dom0`) `vif-nat`, que:

- Pone una dirección IP en `vif<id#>.0`, correspondiente al rango de la subred que se defina en `domU`⁵
- Levanta `vif<id#>.0`
- Añade las reglas de `iptables` correspondientes para hacer NAT.

El esquema es exactamente igual al de la figura 4, aunque ahora las direcciones IP de los `domU` no están en el mismo rango que la dirección IP “externa” del `dom0`.

La salida de `ifconfig` en el `dom0` es:

```
eth0  Link encap:Ethernet  HWaddr 00:40:D0:65:B2:84
      inet addr:192.168.0.64 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::240:d0ff:fe65:b284/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

⁵En el caso nuestro la dirección es `10.0.0.128`, que será la que configuraremos en nuestro `domU` como `gateway`.

```
RX packets:3946 errors:0 dropped:0 overruns:0 frame:0
TX packets:2915 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:5385321 (5.1 MiB) TX bytes:219429 (214.2 KiB)
Interrupt:3 Base address:0xc000
```

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:8 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:560 (560.0 b) TX bytes:560 (560.0 b)

vif1.0 Link encap:Ethernet HWaddr FE:FF:FF:FF:FF:FF
inet addr:10.0.0.128 Bcast:0.0.0.0 Mask:255.255.255.255
inet6 addr: fe80::fcff:ffff:feff:ffff/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1 errors:0 dropped:0 overruns:0 frame:0
TX packets:3 errors:0 dropped:3 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:42 (42.0 b) TX bytes:230 (230.0 b)
```

y la de route -n:

```
Kernel IP routing table
Destination Gateway Genmask Flags Iface
10.0.0.1 0.0.0.0 255.255.255.255 UH vif1.0
192.168.0.0 0.0.0.0 255.255.255.0 U eth0
0.0.0.0 192.168.0.1 0.0.0.0 UG eth0
```

donde podemos ver que la ruta a través de vif1.0 es hacia una máquina (etiqueta "H") con IP 10.0.0.1 y no un segmento de red.

Referencias

- [1] <http://es.wikipedia.org/wiki/Xen>
- [2] Manual de usuario de Xen 3.0
<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user.pdf>
- [3] Rescuing systems using the Debian snapshot server
<http://www.debian-administration.org/articles/435>

- [4] Xen 3 for Debian.
<http://julien.danjou.info/xen.html>
- [5] Installing Xen 3.0 upon Debian Sarge.
<http://www.debian-administration.org/articles/304>
- [6] The Perfect Xen 3.0 Setup For Debian.
http://www.howtoforge.com/perfect_setup_xen3_debian
- [7] Virtualización fácil con Xen3.
<http://crysol.inf-cr.uclm.es/node/201>
- [8] Giving your Xen instances a working X setup.
<http://www.debian-administration.org/articles/322>
- [9] Xen 3.0.2 Setup.
http://mitopia.net/index.php/Xen_3.0.2_Setup
- [10] Debian Xen 3.0 en 5 minutos.
<http://crysol.inf-cr.uclm.es/node/352>
- [11] Create a Debian VM with debootstrap.
http://www.option-c.com/xwiki/Create_a_Debian_VM_with_debootstrap
- [12] Debian Sid gets Xen 3.0.
<http://www.debian-administration.org/articles/396>
- [13] Redes en Xen 3.0
<http://wiki.xensource.com/xenwiki/XenNetworking>
- [14] Lista de correo de Xen. Mensaje 30 de Febrero de 2006
<http://lists.xensource.com/archives/html/xen-users/2006-02/msg00030.html>
- [15] Lista de correo de Xen. Mensaje 949 de Marzo de 2006
<http://lists.xensource.com/archives/html/xen-users/2006-03/msg00949.html>

6. Trucos y demás ...

- Si en un dominio no funciona “CTRL-C” y demás señales, es necesario cambiar la siguiente línea del fichero /etc/inittab:

```
1:2345:respawn:/sbin/getty 38400 console
```

por:

```
1:2345:respawn:/sbin/getty 38400 tty1
```

- Si obtenemos un mensaje como:

```
Device 2049 (vbd) could not be connected.
```

probablemente se trate de un problema al intentar montar alguna de las imágenes en un dispositivo loop, bien porque no haya ninguno libre, no esté cargado el módulo, etc.

La instrucción `losetup` nos puede dar información sobre la situación de los dispositivos loop.

- Para comprobar si nuestro microprocesador tiene soporte de virtualización, tenemos que buscar las etiquetas `vmx` en Intel o `svm` en AMD en la información de la cpu, por ejemplo:

```
cat /proc/cpuinfo |grep svm
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall
nx mmxext fxsr_opt lm 3dnowext 3dnow pni cx16 lahf_lm svm
cr8_legacy                                     ^^^
```