

# Virtualización en GNU/Linux

- Antonio Mario Molina Saorín
- 12 de Julio de 2011
- TC Caldum

# Qué es la virtualización

- Término antiguo:
  - 1960 -> IBM M44/44X
  - Máquina experimental para crear pseudo máquinas (→ Máquinas Virtuales)
- Es la creación de la versión virtual de algo (pc, dispositivo de almacenamiento, de red...)
- Existen muchos tipos de virtualización
  - Hardware, memoria, almacenamiento (entre otros)
- Nos centraremos en la *virtualización de hardware o de plataforma*

# Virtualización hardware

- Un equipo físico (*host* o *anfitrión*) ejecuta una o varias máquinas virtuales (*guest* o *invitado*)
- A su vez, hay varios tipos de virtualización de plataforma:
  - Emulación (permite plataformas distintas)
  - Virtualización completa (SO sin modificar)
  - Paravirtualización (modificación del SO)
  - Virtualización en el sistema operativo
- Nos centraremos en *Virtualización completa*

# Virtualización completa (I)

- Consiste en virtualizar todo el hardware de forma que:
  - Los SOs *guest* no tienen que modificarse
  - Cualquier SO que podamos ejecutar en el host podemos ejecutarlo en el guest sin modificaciones
  - El juego de instrucciones completo se ejecuta sin modificaciones
- Problema:
  - En x86 hay instrucciones que no se pueden virtualizar: no se pueden hacer traps sobre ciertas instrucciones privilegiadas

# Virtualización completa (II)

- Posibles soluciones a este problema:
  - Paravirtualización: Xen
  - Traducción binaria: VMware, VirtualBox
- Desventajas
  - Paravirtualización: **hay que modificar los SOs** de los clientes para usar una API especial
  - Traducción binaria: las instrucciones críticas son localizadas estática o dinámicamente y son emuladas por software en el hipervisor. Esto tiene una **penalización en el rendimiento**

# Virtualización completa (III)

- Intel y AMD -entre otros- introdujeron extensiones para permitir virtualización completa sin esta penalización:
  - Intel → VT-x (Vanderpool)
  - AMD → AMD-V (Pacífica)
- Esto permite una virtualización completa asistida por hardware
- Para un rendimiento óptimo → drivers paravirtualizados → Virtualización híbrida

# Virtualización asistida por hard.

- Es realmente un tipo de virtualización completa pero:
  - Aprovecha ciertas características hardware (del procesador) para mejorar rendimiento en VMs
- Gracias a las extensiones de virtualización de los procesadores → VMs con sistemas operativos sin modificar y con gran rendimiento
- Ejemplos:
  - VMware ESX, MS. HyperV, Xen 3.0, ¡¡KVM!!

# Caso práctico

Vamos a ver cómo instalar y usar el software de virtualización que viene incluido en el núcleo desde la versión 2.6.20:

# KVM

(Kernel-based Virtual Machine)



# Prerrequisitos

- En el PC/Servidor donde vamos a instalar KVM necesitamos:
  - Procesador con soporte de virtualización activado
    - Chequear que nuestro procesador dispone de dicha capacidad:  

```
cat /proc/cpuinfo | grep vmx      (Intel)
```

```
cat /proc/cpuinfo | grep svm      (AMD)
```
    - Chequear en BIOS que está activado
  - Suficiente memoria RAM:
    - depende de los Sistemas Operativos -SOs- guests que vayamos a instalar

# Instalación de KVM

- `aptitude -y install kvm bridge-utils uml-utilities`

- Editamos `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet manual
```

```
# El bridge
auto br0
```

```
iface br0 inet static
    address 192.168.1.5
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
    broadcast192.168.1.255
    bridge_ports eth0
    bridge_fd 9bridge_hello 2
    bridge_maxage 12
    bridge_stp off
```

# KVM y QEMU

- KVM es el hypervisor, que está integrado en el propio kernel del Linux
- Para poder usarlo necesitamos un front-end
- Para esto último, KVM usa **QEMU**
- La sintaxis, como veremos, es similar a *qemu*
- Para crear discos duros virtuales usaremos `qemu-img`
- Vamos a ver un ejemplo práctico.  
Comenzaremos creando el disco duro virtual.

# Creación de hds virtuales

- Creamos un disco duro de 10 GB
  - `qemu-img create -f qcow2 ubuntu.qcow 10G`
- qcow2 es el formato nativo de qemu
- Este formato permite:
  - Snapshots
  - Encriptación AES
  - Compresión basada en zlib
  - La imagen del disco no ocupa todo el tamaño:
    - Va creciendo conforme hace falta más disco
  - Soporta otros formatos: raw, vmdk, vvfat, vpc, ...

# Iniciando nuestra primera VM

- Arrancamos nuestra VM con este comando:
- ```
kvm -no-acpi -m 384 -cdrom /dev/sr0 -hda  
ubuntu.qcow -boot d
```

  - Sin ACPI
  - 384 MB RAM
  - El CD de la VM (guest) es /dev/sr0 en el host
  - El disco duro es el archivo .qcow indicado
  - Arrancará desde CD-ROM (*boot d*)

# Controlando la CPU

- Con el comando anterior sólo estamos usando un core
- Podemos ver los cores que tenemos con:
  - `cat /proc/cpuinfo`
- Podemos asignar más de un core con el parámetro *-smp*:
  - `kvm -m 384 -smp 4 -cdrom /dev/sr0 -hda ubuntu.qcow -boot d`
  - Quitamos *-no-acpi* para que al hacer un halt la VM se apague
  - Le indicamos que queremos 4 cores:
    - Podemos comprobarlo dentro de la VM

# Para que probéis

- En <http://wiki.qemu.org/Download> hay imágenes preparadas para descargar y probar con *kvm*
- Existen UIs para manejar las VMs. El más famoso es *virt-manager*, desarrollado por *RedHat*:
  - `# aptitude install virt-manager`
- *Virt-manager* es un interfaz gráfico que permite crear máquinas virtuales así como lanzarlas y gestionarlas tanto en local como en remoto

# Otras posibilidades

- Existen otras alternativas a KVM:
  - Xen
    - Antes sólo paravirtualización
    - Ahora también soporta full-virtualization
  - Virtualbox
    - Virtualbox es similar a VMWare Workstation.
    - Es muy sencilla de manejar.
    - Todo se hace a nivel gráfico.
    - # aptitude install virtualbox



# Antes de finalizar

- Licenciado bajo Creative Commons
- Reconocimiento - No Comercial - Compartir igual
- Puedes:
  - Copiar, reproducir, mostrar públicamente y modificarlo, siempre y cuando cites al autor (o sea, yo),
  - no lo uses para fines comerciales,
  - y las obras derivadas se mantengan bajo esta misma licencia.

# Preguntas



# Gracias por vuestra atención

