



# Kernel based Virtual Machine

## Retours d'expériences

Jacquelin Charbonnel (CNRS LAREMA)

JoSy virtualisation - Strasbourg, juin 2011

*version 1.2*

# Plan

- Présentation
- Mise en oeuvre
- KVM dans un laboratoire : retour d'expérience
- KVM et la PLM : retour d'expérience
- PLACO

# Présentation de KVM

# KVM

- Kernel Based Virtual Machine
  - solution de full virtualization
  - open source
  - intégré au noyau Linux
  - dérivé de QEmu
  - nécessite Intel VT ou AMD-V
- Composition
  - 1 module kernel kvm.ko
  - 1 module spécifique au processeur kvm-intel.ko ou kvm-amd.ko
  - QEMU : émulateur de machine virtuelle

# KVM

- Possibilités
  - machines virtuelles Linux ou Windows
  - l'OS des vm est non modifié
  - 1 vm = 1 hardware
    - disques, cartes réseau, carte vidéo, périphériques USB, etc.
- Evolution très rapide
  - surtout depuis le support officiel de RedHat (RHEL 5.4)

# Entrées/sorties

- KVM supporte 2 systèmes d'E/S :
  - classique : simule la présence d'un matériel
    - existant réellement
    - généralement répandu
    - pour lequel le système invité a déjà un pilote
  - virtio
    - système d'entrées/sorties virtuelle présenté à l'OS de la vm
    - canaux de communications particuliers vers le matériel du système hôte :
      - accès mémoire,
      - disque,
      - horloge temps réel
      - réseau.
    - + performant
    - nécessite des drivers spécifiques sur la vm

# libvirt

- API de virtualisation
  - API C (Linux, Solaris, Windows)
  - interfacée avec les principaux langages
- supporte XEN, KVM, OpenVZ, VirtualBox, VMware ESX & GSX
- fonctionnalités :
  - management des vm, des réseaux virtuels,
  - management à distance

# Format des images disques

- raw
  - simple et interopérable
- COW
  - supporté pour compatibilité
  - Copy On Write format
- qcow, qcow2
  - QEMU format, compression, chiffrement, snapshot
- vmdk
  - VMWare



# Copy On Write

- Snapshot
  - 1 disque initial, en RW
  - n disques en RO pour stocker ponctuellement l'état du disque initial
- *Backing-files*
  - une paire de disques :
    - 1 disque initial, en RO
    - 1 disque pour stocker les modifs apportées au disque initial
  - la VM utilise en permanence ces 2 disques

# Paramétrage

- stockage
  - mapping d'un device du host
  - image disque
    - préallocation optionnelle suivant le format
- architecture réseau
  - NAT
  - Bridge
- nombre de processeurs de la VM :
  - de 1 à 16
- architecture CPU de la VM
  - i686, x86\_64, mips, sparc, ppc

# Paramétrage

- RAM
  - startup memory
  - max memory
- méthode d'installation
  - depuis un lecteur de CD-ROM
  - depuis une image ISO sur le host
  - depuis un répertoire réseau : HTTP, FTP, NFS
  - par PXE
- optimisation suivant le type d'OS guest
  - generic
  - linux : debian, fedora, redhat, ubuntu
  - windows : vista, 2000, 2003, 2008, XP x86, XP x86\_64
  - unix : freeBDS, openBSD
  - solaris : Sun solaris, openSolaris

# Mise en oeuvre

# Installation

```
# grep -E '(vmx|svm)' /proc/cpuinfo
```

- RH, Centos, Fedora

```
# yum -y install kvm libvirt virt-manager virt-viewer  
# reboot # pour charger kvm.ko et démarrer libvirtd
```

- Debian, Ubuntu

```
# apt-get install kvm libvirt virt-manager virt-viewer  
# reboot
```

# virt-manager

File Edit View Help

View: All virtual machines

Name	ID	Status	CPU usage	Memory usage
coquille4	qemu	Active	0.00 %	0.00 MB 0 %
linux	-	Shutoff	0.00 %	512.00 MB 0 %

Delete New Open

# Création d'une VM

## Virtual Machine Name

Please choose a name for your virtual machine:

Name:

 **Example:** system1

# Virtualization Method

You will need to choose a virtualization method for your new virtual machine:

Paravirtualized:

Lightweight method of virtualizing machines. Limits operating system choices because the OS must be specially modified to support paravirtualization, but performs better than fully virtualized.

Fully virtualized:

Involves hardware simulation, allowing for a greater range of virtual devices and operating systems (does not require OS modification).

CPU architecture:

Hypervisor:

 **Cancel**

 **Back**

 **Forward**



# Installation Method


Please indicate where installation media is available for the operating system you would like to install on this virtual machine:

- Local install media (ISO image or CDROM)
- Network install tree (HTTP, FTP, or NFS)
- Network boot (PXE)


Please choose the operating system you will be installing on the virtual machine:

OS Type: Linux


OS Variant: Fedora 12

 Not all operating system choices are supported by Red Hat. Please see the link below for supported configurations:

[Red Hat Enterprise Linux 5 virtualization support](#)

 Cancel

 Back

 Forward

# Installation Media

Please indicate where installation media is available for the operating system you would like to install on this virtual machine:

ISO image location:


ISO location:


CD-ROM or DVD:


Path to install media:

Fedora 12 x86\_64 DVD



 **Cancel**

 **Back**

 **Forward**

# Storage

Please indicate how you'd like to assign space from the host for your new virtual machine. This space will be used to install the virtual machine's operating system.


Block device (partition):

Location:  Browse...


 **Example:** /dev/hdc2


File (disk image):


Location:  Browse...


Size:   MB


Allocate entire virtual disk now

 **Warning:** If you do not allocate the entire disk now, space will be allocated as needed while the virtual machine is running. If sufficient free space is not available on the host, this may result in data corruption on the virtual machine.

 **Tip:** You may add additional storage, including network-mounted storage, to your virtual machine after it has been created using the same tools you would on a physical system.

 **Cancel**

 **Back**


 **Forward**

# Network

Please indicate how you'd like to connect your new virtual machine to the host network.


Virtual network

Network:  ▾

 **Tip:** Choose this option if your host is disconnected, connected via wireless, or dynamically configured with NetworkManager.


Shared physical device


Device:  ▾


 **Tip:** Choose this option if your host is statically connected to wired ethernet, to gain the ability to migrate the virtual system. (To share a physical device, configure it as a bridge.)

Set fixed MAC address for your virtual machine?

MAC address:

 Cancel

 Back

 Forward

# Memory and CPU Allocation

## Memory:

Please enter the memory configuration for this virtual machine. You can specify the maximum amount of memory the virtual machine should be able to use, and optionally a lower amount to grab on startup. Warning: setting virtual machine memory too high will cause out-of-memory errors in your host domain!

Total memory on host machine: 3.92 GB

Max memory (MB):

Startup memory (MB):

## CPUs:

Please enter the number of virtual CPUs this virtual machine should start up with.

Logical host CPUs: 4

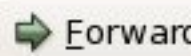
Maximum virtual CPUs: 16

Virtual CPUs:

**i Tip:** For best performance, the number of virtual CPUs should be less than (or equal to) the number of physical CPUs on the host system.

 **Cancel**

 **Back**

 **Forward**





Run



Pause



Shut Down



Fullscreen

Console

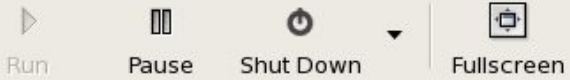
Overview

Hardware

```
NET: Registered protocol family 17
registered taskstats version 1
No TPM chip found, activating TPM-bypass!
  Magic number: 6:81:429
rtc_cmos 00:01: setting system clock to 2010-12-20 11:26:48 UTC (1292844408)
Initalizing network drop monitor service
ata2.00: ATAPI: QEMU DVD-ROM, 0.9.1, max UDMA/100
ata2.00: configured for MWDMA2
scsi 1:0:0:0: CD-ROM                QEMU          QEMU DVD-ROM    0.9. PQ: 0 ANSI: 5
sr0: scsi3-mmc drive: 4x/4x xa/form2 tray
Uniform CD-ROM driver Revision: 3.20
sr 1:0:0:0: Attached scsi generic sg0 type 5
Freeing unused kernel memory: 1324k freed
Write protecting the kernel read-only data: 6344k

Greetings.
anaconda installer init version 12.46 starting
mounting /proc filesystem... done
creating /dev filesystem... done
starting udev...done
mounting /dev/pts (unix98 pty) filesystem... done
mounting /sys filesystem... done
trying to remount root filesystem read write... done
mounting /tmp as tmpfs... done
```

-



Console Overview Hardware

fedora<sup>f</sup>

Please select the nearest city in your time zone:



Map showing city selection interface with zoom controls and a list of cities.

Ville sélectionnée : Paris, Europe

Europe/Paris

System clock uses UTC

← Précédent    Suivant →

# Réseau virtuel

- par défaut, un réseau *default*
  - un switch virbr0
  - un serveur DHCP
  - NAT
- possibilité de créer d'autres réseaux
  - réseau isolé
  - réseau routé



# Création d'un réseau

## Nommage de votre réseau virtuel

Veillez choisir un nom pour votre réseau virtuel :

Nom du réseau :

 **Exemple** : réseau1

 Annuler

 Précédent

 Suivant

## Choix d'un espace d'adressage IPv4

Vous devrez choisir un espace d'adressage IPv4 pour le réseau virtuel :

Réseau : 192.168.100.0/24

**i Conseil** : le réseau devrait être choisi parmi l'un des intervalles d'adresses privées IPv4. Par exemple 10.0.0.0/8, 172.16.0.0/12 ou 192.168.0.0/16


Masque réseau : 255.255.255.0

Diffusion : 192.168.100.255

Passerelle : 192.168.100.1

Taille : 256 adresses

Type : Privé

 Annuler

 Précédent

 Suivant

## Sélection de l'intervalle DHCP

Veillez choisir l'intervalle d'adresses que le serveur DHCP pourra allouer aux machines virtuelles attachées au réseau virtuel.

Activer le DHCP :

Début :

Fin :

**i Conseil :** à moins que vous ne souhaitiez réserver certaines adresses pour permettre la configuration d'un réseau statique pour des machines virtuelles, vous devriez laisser ces paramètres à leurs valeurs par défaut.

 Annuler

 Précédent

 Suivant

## Connexion au réseau physique

Veillez indiquer si ce réseau virtuel devrait être connecté au réseau physique.

- Réseau virtuel isolé
- Réacheminement vers un réseau physique

Destination : Périphérique physique eth0

Mode : Routé

 Annuler

 Précédent

 Suivant

# Disposition

- VM enregistrées dans la console :

```
# ls -l /etc/libvirt/qemu/  
-rw----- 1 root root 1244 Dec 20 10:44 linux.xml  
drwx----- 3 root root 4096 Aug 11 00:07 networks  
-rw----- 1 root root 1252 Dec 20 12:25 vm-fedora.xml
```

- disques virtuels :

```
# ls -l /var/lib/libvirt/images/  
total 8200008  
-rw----- 1 root root 4194304000 Dec 20 11:27 linux.img  
-rw----- 1 root root 4194304000 Dec 20 13:09 vm-fedora.img
```

```
# cat /etc/libvirt/qemu/vm-fedora.xml
<domain type='kvm'>
  <name>vm-fedora</name>
  <uuid>4233a0c8-ff2d-379b-a11b-fae3ed139b87</uuid>
  <memory>524288</memory>
  <vcpu>1</vcpu>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' cache='none' />
      <source file='/var/lib/libvirt/images/vm-fedora.img' />
      <target dev='vda' bus='virtio' />
    </disk>
    <interface type='network'>
      <mac address='54:52:00:24:0c:f9' />
      <source network='default' />
      <model type='virtio' />
    </interface>
  </devices>
</domain>
```

# Commandes utiles

- virt-manager
- virsh
  - gère les opérations usuelles sur les vm
  - ne gère pas (encore)
    - tous les formats de disque
    - l'installation de vm
    - les *backing-files*
- qemu-img
  - gestion des images disques
- virt-install
  - installation de vm
- virt-clone
  - clonage de vm
- qemu-kvm
  - exécution de vm

# virsh

## couteau suisse de libvirt

- gère les vm (*domains*)
  - enregistre / efface
  - démarre / stoppe
  - suspend / redémarre
  - sauve / restaure
  - shutdown / reboot
  - crée / intègre / annule des snapshots
  - affiche / édite la config au format xml
  - gère les périphériques (interfaces réseau, périphériques)
  - liste les vm actives, enregistrées
  - migre les vm sur un autre host
- gère la config réseau virtuelle
- gère les éléments de stockage (*storage pools, volumes*)
- stats diverses



# virsh

## couteau suisse de libvirt

```
# virsh --help
```

```
virsh [options] [commandes]
```

help	imprimer l'aide
create	créer un domaine depuis un fichier XML
start	démarrer un domaine (précédemment défini)
destroy	détruire un domaine
define	définir (mais ne pas démarrer) un domaine depuis un fichier XML
dumpxml	informations du domaine en XML
edit	edit XML configuration for a domain
list	lister les domaines
migrate	migrer un domaine vers un autre hôte
reboot	redémarrer un domaine
restore	restaurer un domaine à partir d'un état sauvé dans un fichier
resume	réactiver un domaine
save	enregistrer l'état du domaine dans un fichier
shutdown	arrêter un domaine proprement
suspend	suspendre un domaine
undefine	supprimer un domaine inactif
snapshot-create	créer un snapshot
snapshot-delete	détruit un snapshot
snapshot-revert	revient à un snapshot
...	

# virt-install

ID=\$1

RAMSIZE=\$2

ROOTSIZE=\$3

SWAPSIZE=\$4

MACid=\$5

BRIDGE=virbr0

#INSTALL=--cdrom=/dev/cdrom

INSTALL=--pxe

```
virt-install --accelerate --hvm --connect qemu:///system \  
    --network=bridge:$BRIDGE $INSTALL \  
    --name $ID --ram=$RAMSIZE \  
    --vcpus=1 \  
    --os-type=linux --os-variant=rhel5 \  
    --disk path=/data/kvm/$ID/$ID-root.img,size=$ROOTSIZE \  
    --disk path=/data/kvm/$ID/$ID-swap.img,size=$SWAPSIZE \  
    --mac=54:52:00:7d:57:$MACid
```

# virt-clone

```
IMGDIR=/var/lib/libvirt/images
```

```
basevm=$1
```

```
newvm=$2
```

```
virt-clone \
```

```
  --original $basevm.xml \
```

```
  --name $newvm \
```

```
  --file $IMGDIR/$newvm/$newvm-root.raw \
```

```
  --file $IMGDIR/$newvm/$newvm-swap.raw
```

# qemu-img

le couteau suisse pour gérer les images disques

- crée, convertit, redimensionne des images disques
  - raw
  - cow, qcow, qcow2
  - vdi (VirtualBox format)
  - vmdk
  - vpc (VirtualPC format)
  - cloop (Linux Compressed Loop image)
- gère les snapshots
- gère les *backing-files*

# qemu-img

le couteau suisse pour gérer les images disques

## NAME

qemu-img - QEMU disk image utility

## SYNOPSIS

usage: qemu-img command [command options]

## OPTIONS

The following commands are supported:

check [-f fmt] filename

create [-f fmt] [-o options] filename [size]

commit [-f fmt] filename

convert [-c] [-f fmt] [-O output\_fmt] [-o options] filename [filename2[...]] output\_filename

info [-f fmt] filename

snapshot [-l | -a snapshot | -c snapshot | -d snapshot] filename

rebase [-f fmt] [-u] -b backing\_file [-F backing\_fmt] filename

resize filename [+ | -]size

# qemu-kvm

- lance une vm spécifiée par options (pas de .xml)

```
qemu-kvm \  
-hda /data/VM/sganarellevm2/root.img \  
-hdb /data/VM/sganarellevm2/swap.img \  
-m 4096 \  
-net nic \  
-net tap,ifname=tap1,script=no \  
-daemonize
```

# Réseaux virtuels

- Exemple sur CentOS/Fedora/RHEL

```
def_bridge()
{
    cat > /etc/sysconfig/network-scripts/ifcfg-$1 << EOL
DEVICE=$1
TYPE=Bridge
ONBOOT=yes
EOL
    ifup $1
}

plug_eth()
{
    cat > /etc/sysconfig/network-scripts/ifcfg-$1 << EOL
DEVICE=$1
ONBOOT=yes
BRIDGE=$2
EOL
}
```

# Réseaux virtuels

```
def_bridge local  
def_bridge public  
def_bridge data
```

```
brctl show
```













```
plug_eth eth1 local  
plug_eth eth2 data  
plug_eth eth3 public
```

```
/etc/init.d/network restart
```



# KVM dans un laboratoire retour d'expérience

View: All virtual machines

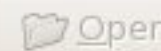
Name	ID	Status	CPU usage	Memory usage
▼ coquille1	qemu	Active	0.25 %	1.96 GB 8 %
icinga	39	 Running	0.13 %	500.00 MB 2 %
laremagw	10	 Running	0.06 %	512.00 MB 2 %
svn	40	 Running	0.00 %	500.00 MB 2 %
tonton	42	 Running	0.06 %	500.00 MB 2 %
▼ coquille2	qemu	Active	0.25 %	2.96 GB 18 %
alceste	2	 Running	0.12 %	500.00 MB 3 %
base_centos5	-	 Shutoff	0.00 %	512.00 MB 0 %
icinguang	4	 Running	0.00 %	500.00 MB 3 %
ltsp2	-	 Shutoff	0.00 %	766.00 MB 0 %
roundcube	1	 Running	0.00 %	500.00 MB 3 %
sogo	5	 Running	0.00 %	512.00 MB 3 %
sshfpl	13	 Running	0.00 %	512.00 MB 3 %
ubuntultsp	14	 Running	0.12 %	512.00 MB 3 %



Delete



New



Open

- LAREMA
  - laboratoire de mathématiques
  - 50 chercheurs
- 2 serveurs de virtualisation sous CentOS 5.6
  - répartition des VM
  - chaque VM est
    - en production sur 1 host
    - en backup sur l'autre
- VM
  - messagerie SMTP+IMAP (postfix+dovecot)
  - web + webmail (apache, squirrelmail, roundcube)
  - firewall + reverse proxy + MX principal (iptables, apache, postfix)
  - monitoring labo + monitoring DSI (icinga)
  - serveurs ssh (tunnels pour les chercheurs du labo et de la fédération des Pays de Loire)

- N'est pas virtualisé :
  - serveurs applicatifs
    - pour performance
  - serveur de sauvegarde
    - pour isolation
  - serveur de calcul
    - pour performance
  - serveurs d'infrastructure : nfs, ntp, ldap, dhcp, dns
    - car ses services sont utilisés par les serveurs de virtualisation
- aucun gros volume de données sur disques virtuels
  - montage NFS depuis un serveur physique dédié
- disques virtuels
  - sur LVM sur le host

# backups

- intérieur des VM
  - config système sauvegardée tous les jours
  - intégré au système de backup général
  - archivage *daily, weekly, monthly*
- images disques
  - sauvegardée 1 fois par mois
  - sur le second host
  - sans archivage

# restauration

- reprendre la sauvegarde de l'image disque
- démarrer la VM
- rsync du dernier backup du contenu

# backups

- processus de backup des vm d'un host

1. pause de toutes les vm

- virsh suspend

2. snapshot LVM du LV contenant les disques

- lvcreate --snapshot

3. redémarrage de toutes les vm

- virsh resume


4. rsync des disques du snapshot sur l'autre host

- ionice -c3 -- rsync -a

5. suppression du snapshot LVM

- lvremove

6. rsync des définitions de vm (.xml)



- d'1 sec

# KVM sur la Plate-forme en Ligne pour les Mathématiques

retour d'expérience

# PLM

- Plate-forme en ligne pour les mathématiques
  - infrastructure répartie géographiquement
  - utilisée par 80 laboratoires (1600 utilisateurs)
- 4 sites (Bordeaux, Lille, Angers et Lyon)
  - 7 hosts sous CentOS
  - 40 VM
- initialement sous VMware server 1.x
- 2009 : début de migration vers KVM
- 2011 : fin de la migration KVM



# Pourquoi migrer

- fin de vie de VMware Server 1.x
- pas de console *native* avec VMware Server 2
  - console via un navigateur web
- 1 petit bug jamais résolu au niveau des snapshots de VM

# Bilan

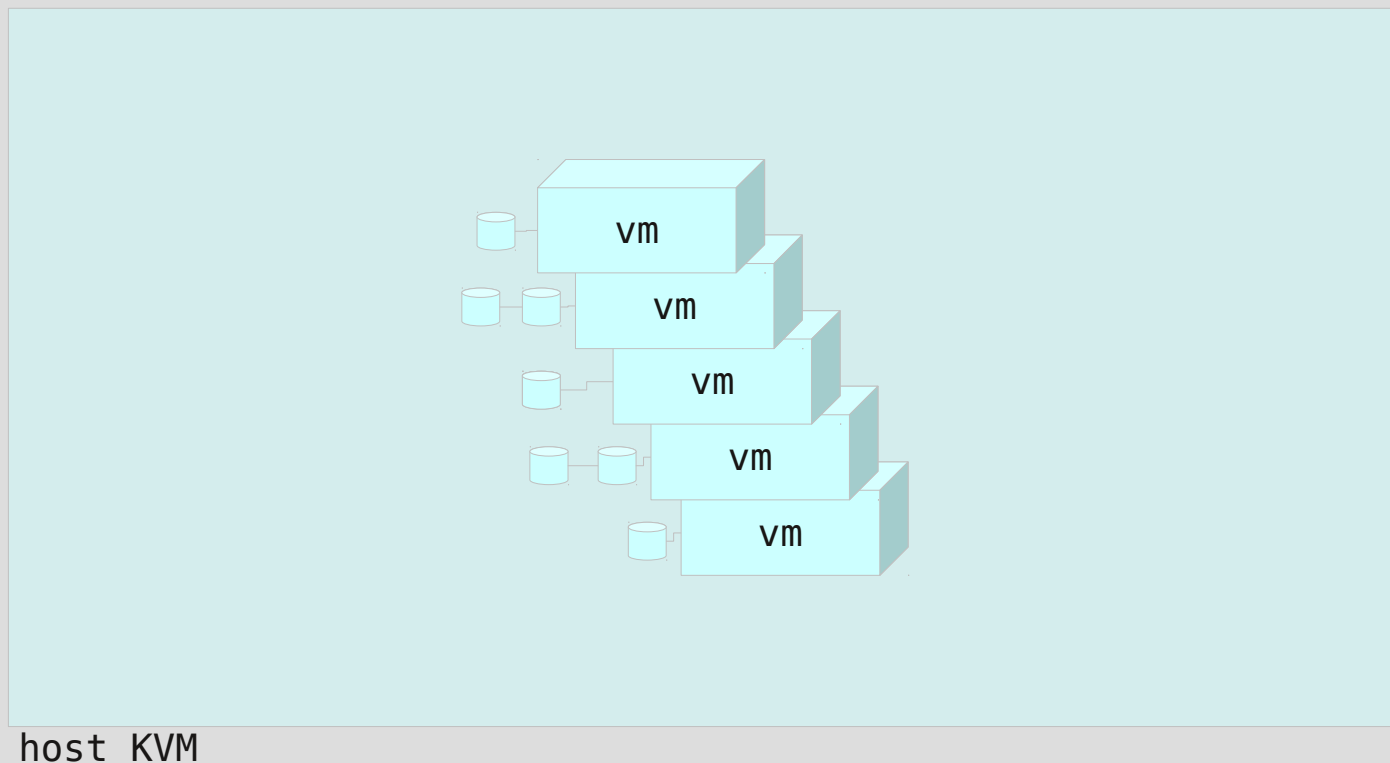
- meilleures performances
- installation + facile
  - intégré à la distribution CentOS
  - rien à recompiler à chaque nouveau kernel
  - pas de tools à installer sur les VM

# Principes

- backups : même principe
  - disques sur Logical Volumes
  - sauvegarde quotidienne + archivage du contenu
  - sauvegarde en 1 exemplaire du contenant (période longue)
- éviter les VM avec de nombreux ou gros disques
- 2 interfaces réseau pour une VM
  - 1 interne pour communiquer avec le host (nfs, ntp, dns, etc.)
  - 1 externe pour accéder aux services *publics*
- avoir des VM *déplaçables* d'un site à l'autre
  - VM : dhcp
  - dnsmasq sur le host

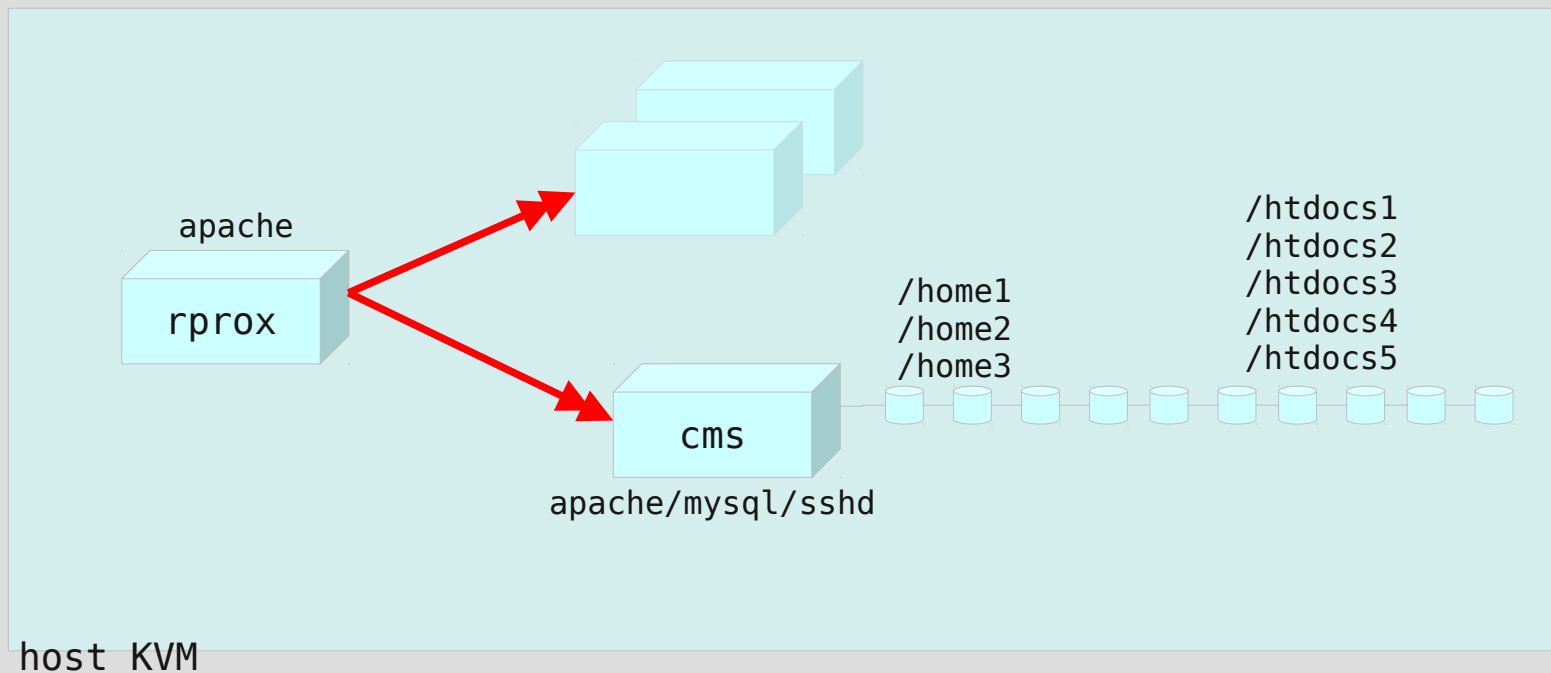
# Exemple : hébergement de serveurs dédiés

- vm affectées à des projets
  - les admin ne sont pas les admin de la plate-forme
- les données sont dans des disques virtuelles



# Exemple : hébergement de sites web

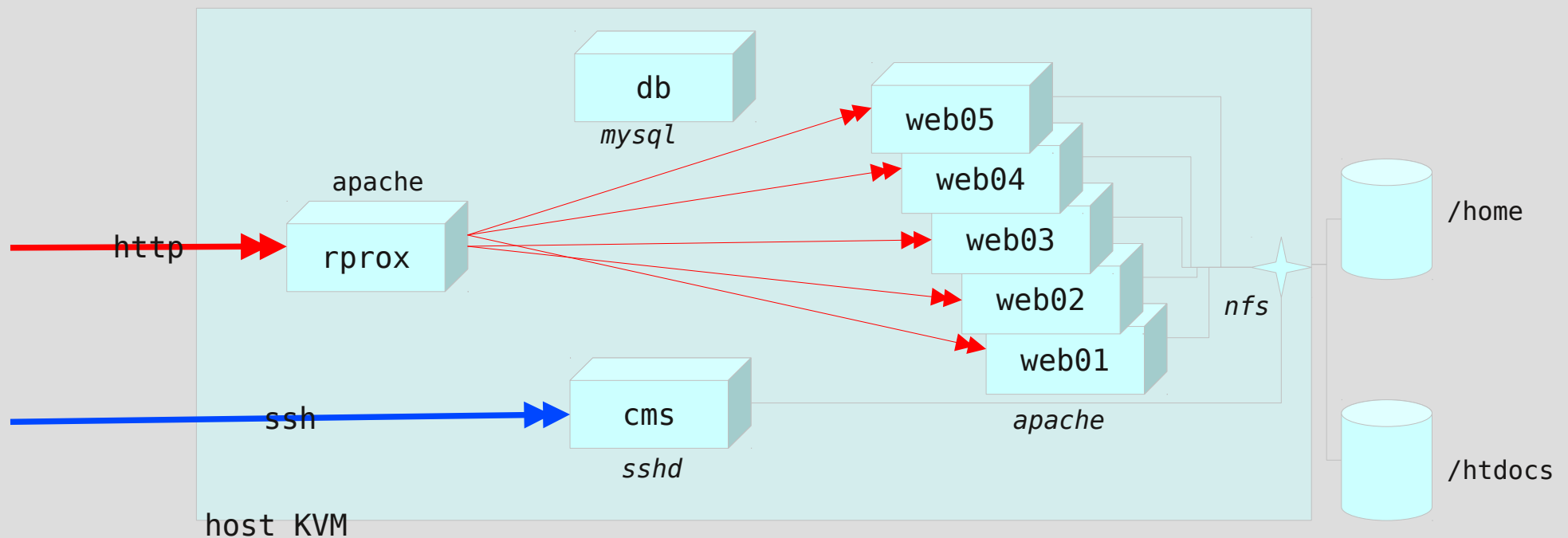
- architecture initiale
  - monolithique : 1 serveur apache
  - 1 site = 1 virtual host
  - jusqu'à 10 disques de 4Go : lourds à gérer
  - 1 site peut impacter la performance de tous les autres



# Exemple : hébergement de sites web

- architecture actuelle

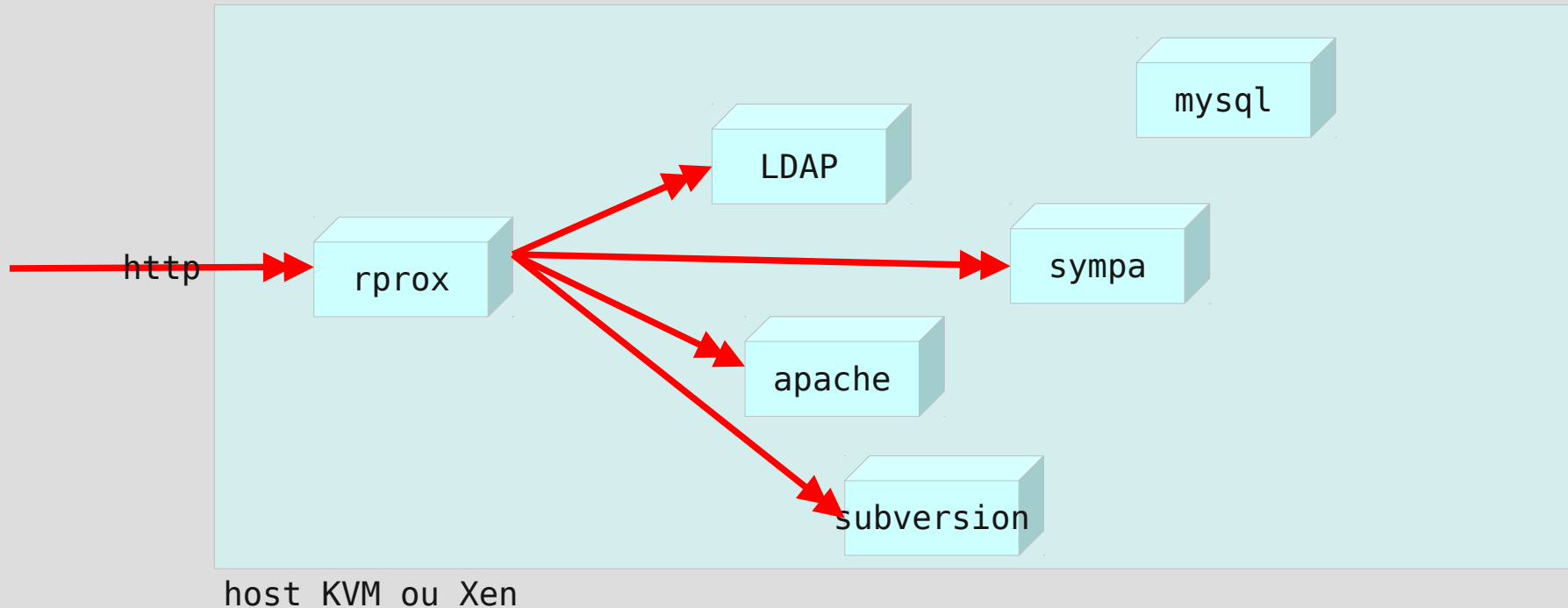
- RAM de chaque webxx : 2Go
- granularité CPU / IO / RAM
- plusieurs versions possibles (php, etc.)



# KVM et le projet PLACO

# PLACO

- générateur de plates-formes collaboratives
- chaque brique est une VM
- base authentification unique : OpenLDAP
- 2 hyperviseurs possibles : Xen et KVM



<http://placo.mathrice.fr>  
<http://placodev.mathrice.fr>