La para-virtualisation Xen sur Debian

Retour d'expérience

Gabriel Moreau

LEGI - Laboratoire des Écoulements Géophysiques et Industriels

UMR 5519 - CNRS / Université de Grenoble (UJF/G-INP) - France

http://www.legi.grenoble-inp.fr/

9 juin 2011

Le laboratoire LEGI

Contexte

- 175 personnes (70 chercheurs)
- 450 adresses MAC sur le réseau
- 3 OS (Windows, GNU/Linux, MacOSX)
- des portables à la machine de calcul ayant 192 coeurs
- 200To de Stockage
- 2 sites dont un, petit mais costaud !
- 19 VLAN (17 sur le site principal)
- 5 Classes C publique - 4 C privé - 1 B privé
- gros utilisateurs des centres de calculs / grands instruments
- 2 informaticiens support
- 3 informaticiens développement dédié

Le laboratoire LEGI

Contexte

Système d'exploitation

Stratégie serveur 100% GNU/Linux orientée sur la distribution Debian

- de Sarge à Squeeze
- du 32 au 64 bits
- tous piloté via un serveur `cfengine2`
- tous différents mais tous pareils
- du portable au noeud de calcul
- pas de serveur Windows ni MacOSX
- virtualisation dans Xen de presque tous les serveurs

Le laboratoire LEGI

Contexte

Machines dom0 Xen

- 2 DELL 1850 bi-xeon 2.80GHz - 72G disque + 1.8Go RAM
- 3 NEC bi-core2 1.86GHz - 750G disque + 4Go RAM
- 4 DELL R610 bi-core2 2.27GHz - 500Go disque + 24Go RAM - baies AoE
- 2 DELL R410 bi-core2 2.13GHz - 500Go disque + 16Go RAM
- 2 DELL R200 core2 3GHz - 750Go disque + 4Go RAM

Environ 45 machines virtuelles domU

Déploiement Xen sous Debian

Installation d'un poste

Installation d'un poste

- Classique via le CD netinstall de Debian
- Automatique via FAI (abandonné pour le moment)
- Clef USB netinstall avec ajout Debian Preseed (équivalent du kickstart de Red-Hat)
  - souple
  - une clef par type de machine
  - installation minimale
  - ne modifie pas les paramètres du BIOS ;-(

Déploiement Xen sous Debian

Xenification

Xenification

Transformation du poste en serveur maître dom0

- déclarer le poste dans DNS/DHCP
- déclarer le poste dans cfengine
- `cfagent -qvK`
- reboot !

Déploiement Xen sous Debian

Configuration Réseau du dom0

Choix du type de réseau

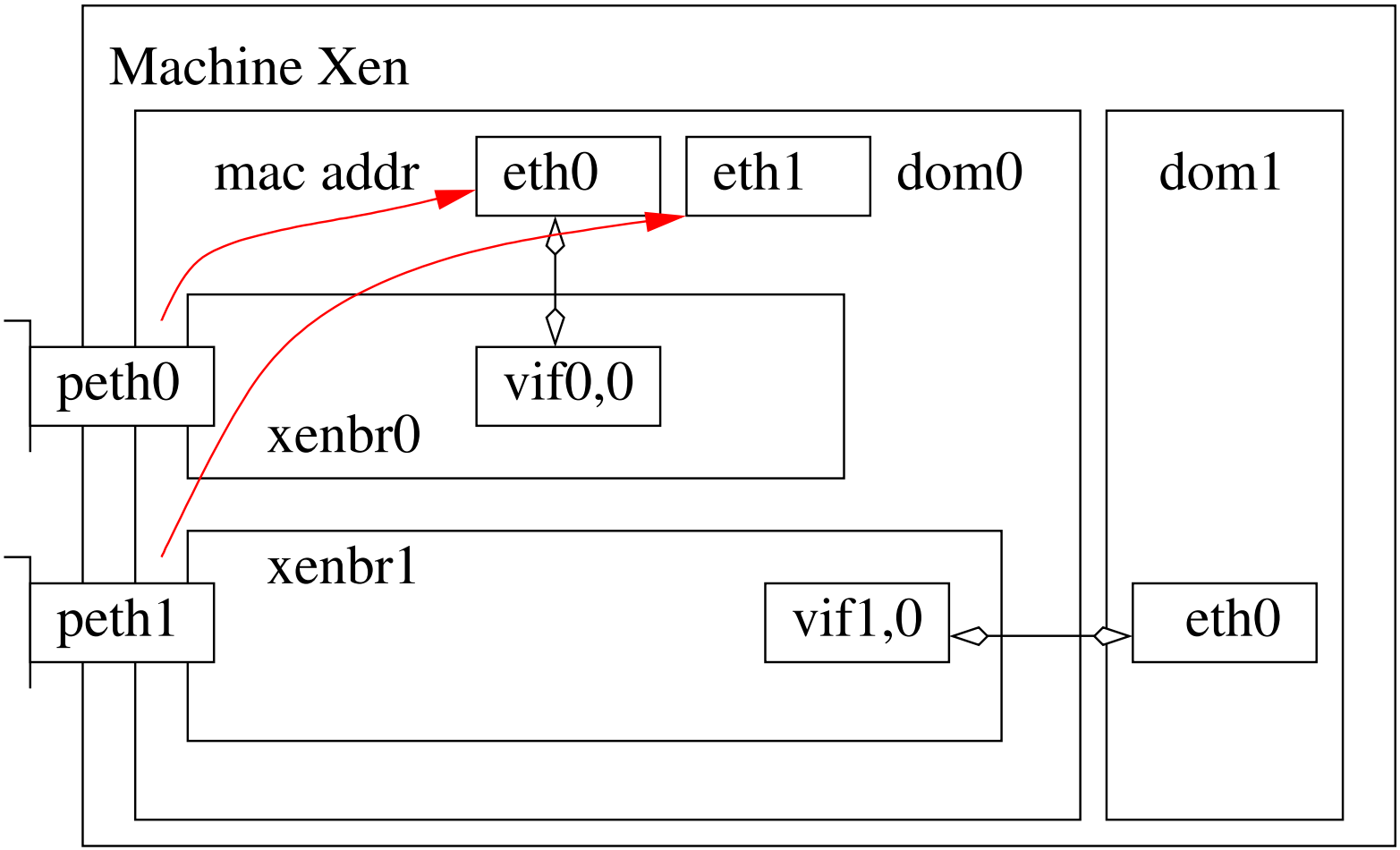
- Deux ports réseau physique par serveur, `eth0` dédié exclusivement au dom0
- Accès réseaux aux domU via des ponts virtuels (bridge)
- Gestion par la distribution du réseau (maintenant conseillé par Xen)

/etc/xen/xend-config.sxp

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

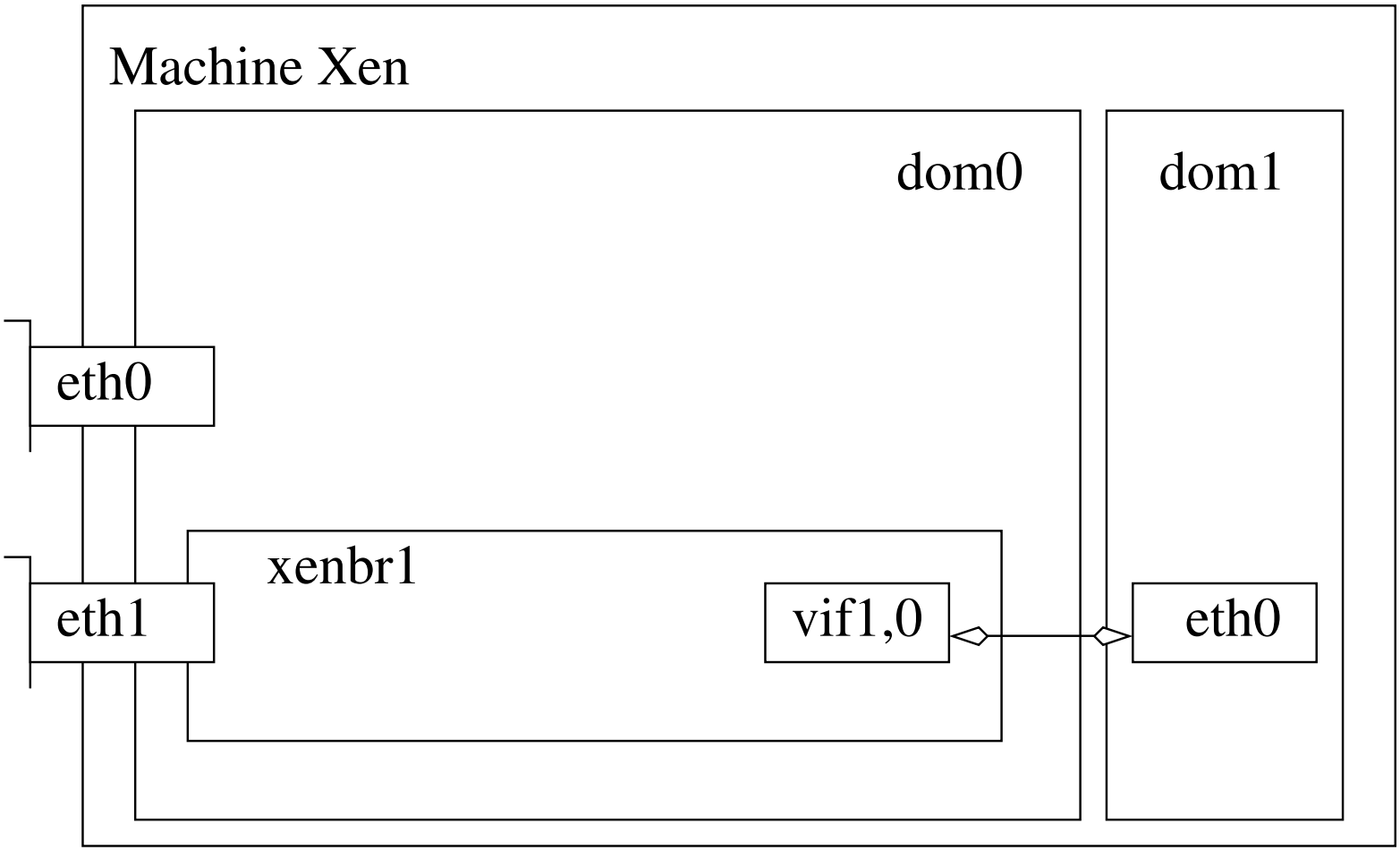
Déploiement Xen sous Debian

Configuration Réseau - Avant - Gestion par Xen



Déploiement Xen sous Debian

Configuration Réseau - Maintenant - Gestion par la distribution



Déploiement Xen sous Debian

Configuration Réseau - Avant - Gestion par Xen

- Pas très bon
- Bidouille eth0/peth0
- Marchait bien mais pose aussi des problèmes...
- On a aussi très souvent IPMI sur eth0 !
- Bref, solution historique Xenifiante (pour être polie)

- Spécialisation des interfaces
- eth0 : accès dom0 (VLAN_D0) + IPMI BIOS (VLAN_BMC)
- eth1 : accès domU (VLAN_D1, VLAN_D2...)
- Des débits, bond0 entre eth1 et eth2...
- Gestion des bridges par la distribution et plus par les scripts de Xen !

Déploiement XEN sous DEBIAN

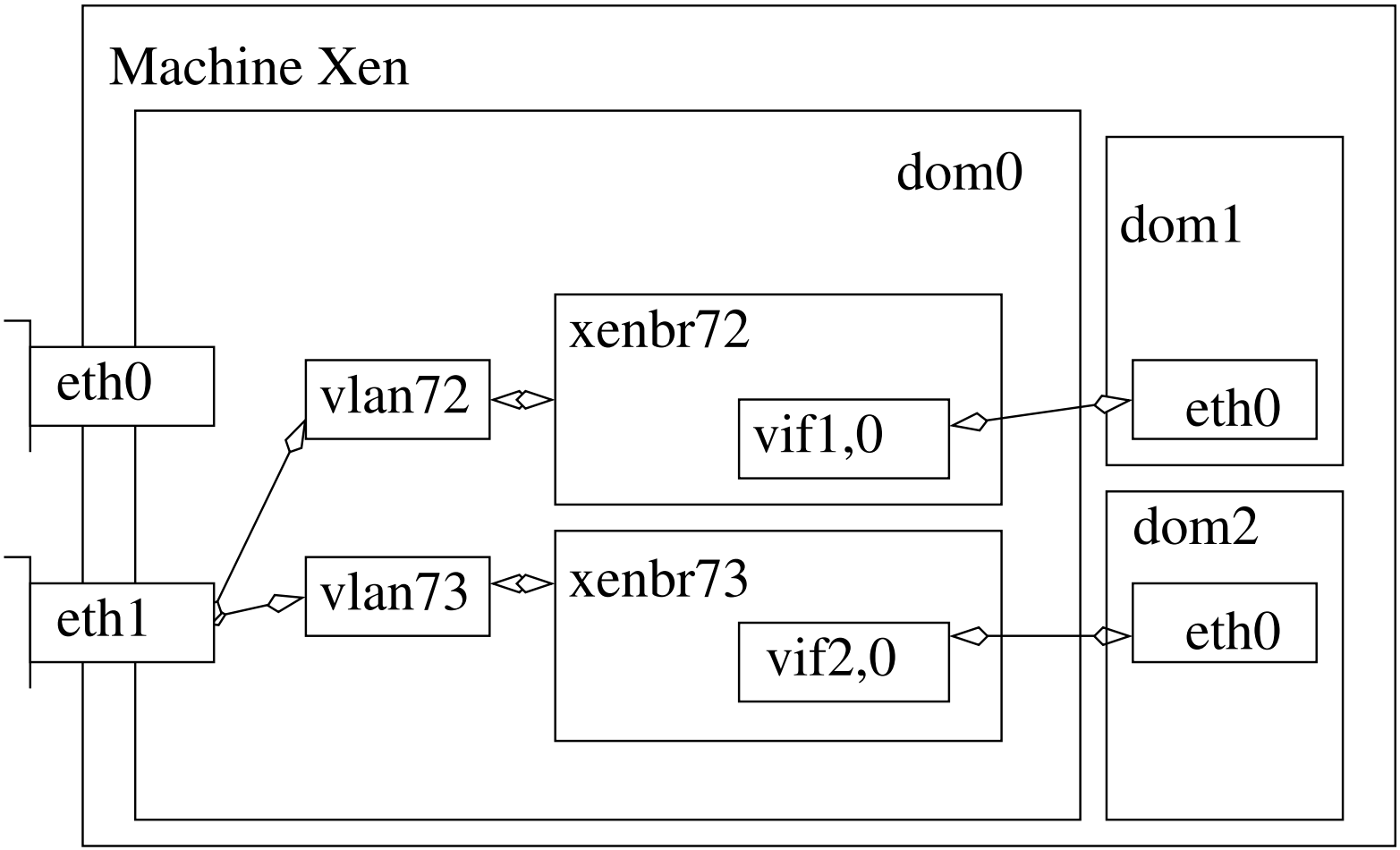
Configuration Réseau - Fichier /etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 192.168.63.240
   netmask 255.255.255.0
   broadcast 192.168.63.255
   gateway 192.168.63.254

auto eth1
iface eth1 inet manual
   up ip link set dev eth1 arp off

auto xenbr1
iface xenbr1 inet manual
   bridge_ports eth1
   bridge_stp off
   bridge_fd 0
```

- Active manuellement l'interface eth1
- Supprime ARP sur eth1
- Lie le pont (bridge) xenbr1 sur eth1

Déploiement Xen sous Debian

Configuration Réseau - Avec VLAN



Déploiement Xen sous Debian

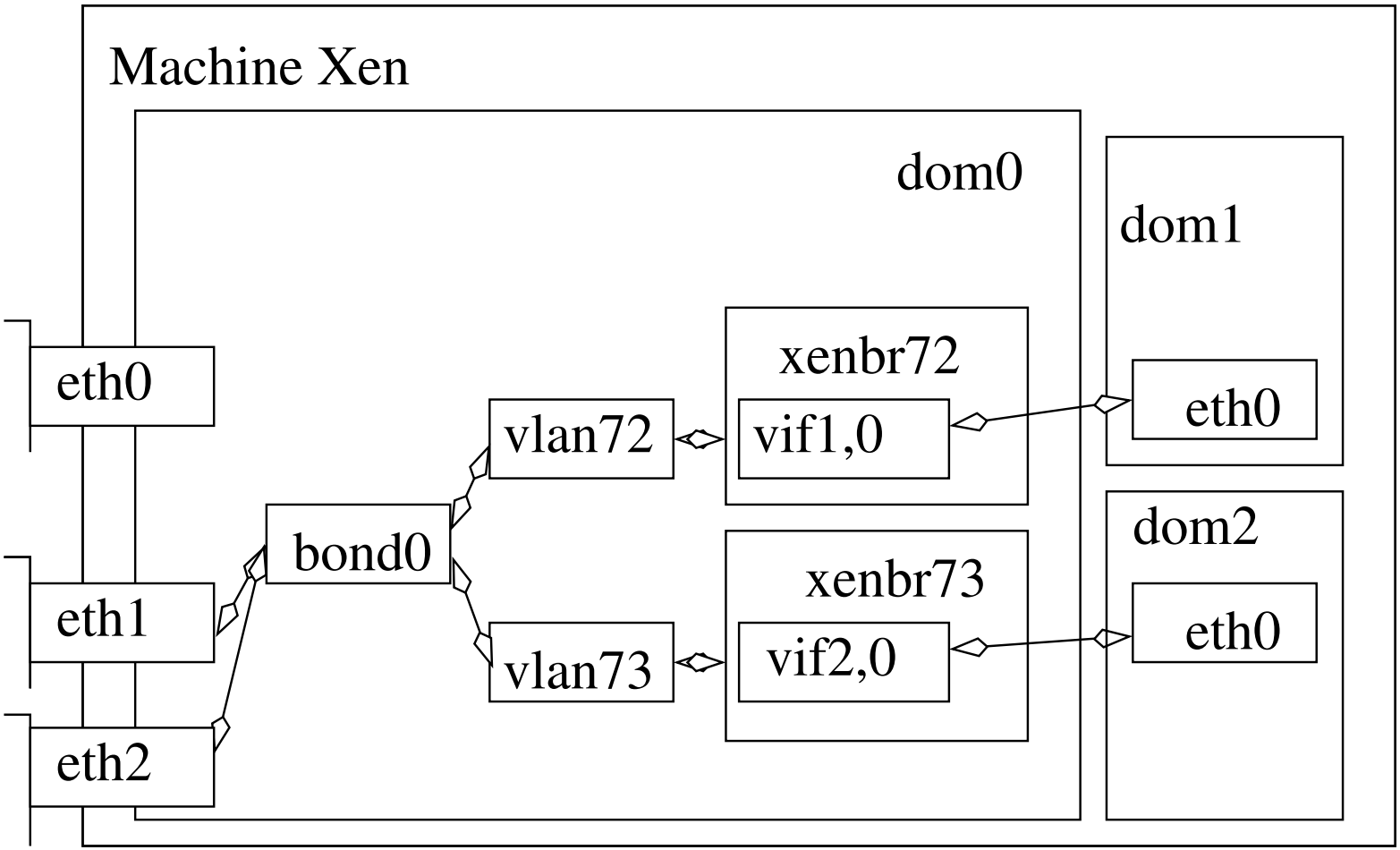
Configuration Réseau - VLAN - Fichier /etc/network/interfaces

```
auto eth1
iface eth1 inet manual
   up ip link set dev eth1 arp off
   up modprobe 8021q

allow-xenbr72 vlan72
iface vlan72 inet manual
   vlan_raw_device eth1

auto xenbr72
iface xenbr72 inet manual
   bridge_ports vlan72
   bridge_stp off
   bridge_fd 0
   pre-up ifup --allow "$IFACE" vlan72
   post-down ifdown --allow "$IFACE" vlan72
```

- Activer la pile 802.1Q
- Rien sur eth0 (port Untagged sur commutateur HP)
- dom0 gère les VLAN pour les domU (port Tagged sur commutateur HP)

Déploiement Xen sous Debian

Configuration Réseau - Performance - Avec agrégation de lien (Bonding)



Déploiement Xen sous Debian

Configuration Réseau - BOND - Fichier /etc/network/interfaces

```
allow-bond0 eth1
iface eth1 inet manual

allow-bond0 eth2
iface eth2 inet manual

auto bond0
iface bond0 inet manual
   bond_mode 802.3ad
   bond_miimon 100
   bond_downdelay 200
   bond_updelay 200
   slaves eth1 eth2
   up ip link set dev bond0 arp off
   up modprobe 8021q

allow-xenbr72 vlan72
iface vlan72 inet manual
   vlan_raw_device bond0
```

- Activer le mode LACP sur le commutateur sur les ports de bonding (HP)
- Compatible avec le Spanning Tree des commutateurs (HP)

Déploiement Xen sous Debian

Configuration Réseau - Bilan

- On sais créer autant de VLAN qu'il le faut
- Un pont (bridge) par VLAN
- On fait du bonding pour augmenter le débit
- On rattache les machines virtuelles domU a autant de bridge qu'on le souhaite...

Encore un peu plus ?

Déploiement Xen sous Debian

Configuration Réseau - IPMI - Fichier /etc/network/interfaces

```
auto eth0
iface eth0 inet static
   address 192.168.63.240
   netmask 255.255.255.0
   network 192.168.63.0
   broadcast 192.168.63.255
   gateway 192.168.63.254
   up modprobe ipmi_devintf
   up modprobe ipmi_si
   up ipmitool lan set 1 vlan id 83
   up ipmitool lan set 1 ipsrc static
   up ipmitool lan set 1 ipaddr 10.83.63.240
   up ipmitool lan set 1 netmask 255.255.0.0
```

- Tagger le VLAN IPMI sur le port eth0 du commutateur
- Activer les modules IPMI dans le noyau
- Chez NEC, IPMI est sur le lan set 2 !

Déploiement Xen sous Debian

Configuration Réseau du dom0 - Bilan

A ce jour

Le dom0 est quasiment un commutateur pilotable à distance... Il est possible de gérer de nombreux VLAN ayant des noms significatifs permettant d'avoir une configuration lisible. Avec IPMI, il est facile de rebooter à distance s'il y a plantage intégral du système.

Futur

Open vSwitch va permettre de créer un vrai commutateur virtuel qui s'étendra sur plusieurs dom0.

`http://openvswitch.org/`

Déploiement Xen sous Debian

Création d'une machine virtuelle

Création d'une machine virtuelle - LVM

Stockage du serveur virtuel : par fichier / par partition.

- Par fichier $\longrightarrow$ copie simple / backup simple / performance faible
- Par partition $\longrightarrow$ copie complexe / backup simple si LVM / performance bonne

Déploiement Xen sous Debian

Création d'une machine virtuelle

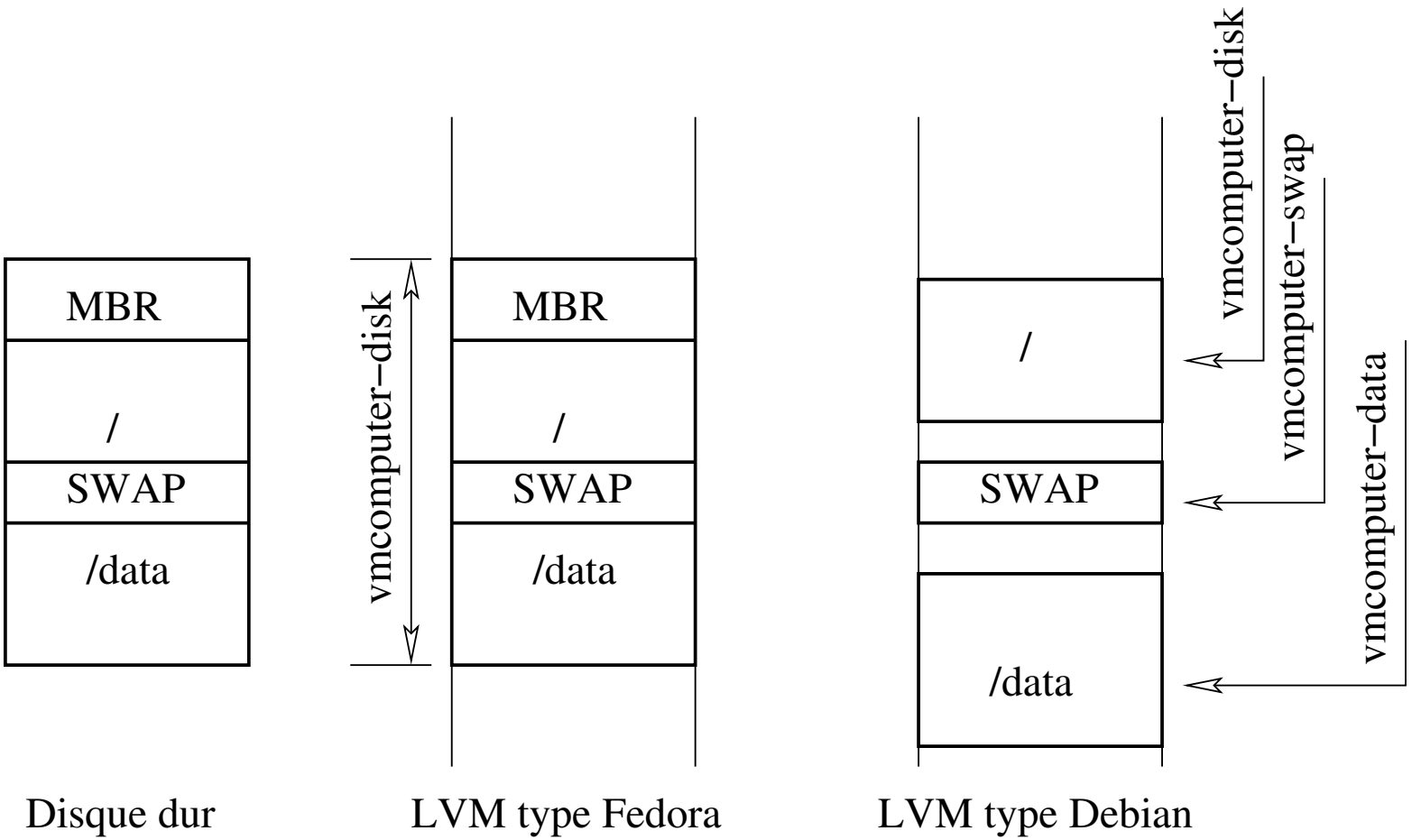
Création d'une machine virtuelle - LVM

Gérer 36 partitions sur un disque de PC $\longrightarrow$ horreur ! La table des partitions des PC est statique et ancestrale (partition primaire, logique, étendue...).
Solution $\longrightarrow$ LVM. Gestion souple et dynamique des partitions + une solution de backup exceptionnelle pour notre problème.
Création d'un partition LVM et d'un volume group. Choix dans cette présentation : vg0

```
apt-get install lvm2
pvcreate /dev/sdb
vgcreate /dev/sdb vg0
```

Déploiement Xen sous Debian

Une partition LVM par partition du domU



Déploiement Xen sous Debian

Création d'une machine virtuelle

Via Debootstrap

Le script crée les partitions LVM, formate, télécharge la distribution sur le net et l'installe. . . en 3 mn environ.

```
xen-create-image ...
```

Manuel

On a un template de machine dans le dom0. On crée la partition système et la partition de swap et on copie via rsync.

```
lvcreate -n dom1-disk -L 6G vg0
lvcreate -n dom1-swap -L 64M vg0
mkfs + mkswap
mount /dev/vg0/dom1-disk /tmp/target
rsync -av /srv/xen/templates/squeeze-amd64/ /tmp/target/
```

Déploiement Xen sous Debian

Création d'une machine virtuelle - Fichier de configuration /etc/xen/dom1.cfg

```
kernel  = '/srv/xen/boot/vmlinuz-2.6.32-5-xen-amd64'
ramdisk = '/srv/xen/boot/initrd.img-2.6.32-5-xen-amd64'
memory  = '512'
vcpus   = 2
root  = '/dev/xvda2 ro'
disk  = [
         'phy:/dev/vg0/legilnx46-servload-swap,xvda1,w',
         'phy:/dev/vg0/legilnx46-servload-disk,xvda2,w',
         ]
name  = 'dom1'
vif   = [
         'mac=00:16:3E:73:63:B5, bridge=xenbr73',
         ]
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
extra = "console=hvc0 xencons=tty"
```

Le fichier est modifiable pour rajouter de la mémoire, changer l'adresse MAC, changer les partitions du disque, déplacer une machine virtuelle d'un serveur sur un autre...

Déploiement Xen sous Debian

Création d'une machine virtuelle

Points importants chez Debian

- Le noyau est hébergé sur le dom0
- Par défaut, les noyaux du dom0 et des domU sont identiques
- Dissocier les noyaux du domU de celui du dom0 (penser migration, upgrade, mix de version)
- Une ligne par partition (simplifie les scripts de sauvegarde)

On finit l'installation comme une machine classique, avec `cfengine`.

Déploiement Xen sous Debian

Création d'une machine virtuelle - Multi VLAN - Config dom0

```
vif    = [
          'mac=00:16:3E:73:63:B5, bridge=xenbr73',
          'mac=00:16:3E:74:63:B5, bridge=xenbr74',
          'mac=00:16:3E:75:63:B5, bridge=xenbr75',
          'mac=00:16:3E:76:63:B5, bridge=xenbr76',
          'mac=00:16:3E:77:63:B5, bridge=xenbr77',
          'mac=00:16:3E:78:63:B5, bridge=xenbr78',
         ]
```

Les VLAN sont gérés dans le dom0 (sécurité). Le domU est raccroché a de multiple VLAN pour *X* raison (DHCP / Routeur virtuel / SNIF réseau...).

Déploiement Xen sous Debian

Création d'une machine virtuelle - Multi VLAN - Config domU

STOP : pas de eth0, eth1...

- Des règles de nommage dans udev (`/etc/udev/rules.d/70-persistent-net.rules`)
- `VLAN 73` $\longrightarrow$ `eth73`
- `VLAN 74` $\longrightarrow$ `eth74`
- Des noms d'interface claire et maintenable
- Attention : udev veut des minuscules au niveau des adresses MAC !

```
# VLAN 73
SUBSYSTEM=="net", DRIVERS=="?*", ATTRSaddress=="00:16:3e:73:63:b5", NAME="eth73
# VLAN 74
SUBSYSTEM=="net", DRIVERS=="?*", ATTRSaddress=="00:16:3e:74:63:b5", NAME="eth74
...
```

Mise en production Xen sous Debian

Mise à jour d'une architecture Xen

Le principal problème

Avoir un noyau dom0 et des noyaux domU différent.

Solution Debian

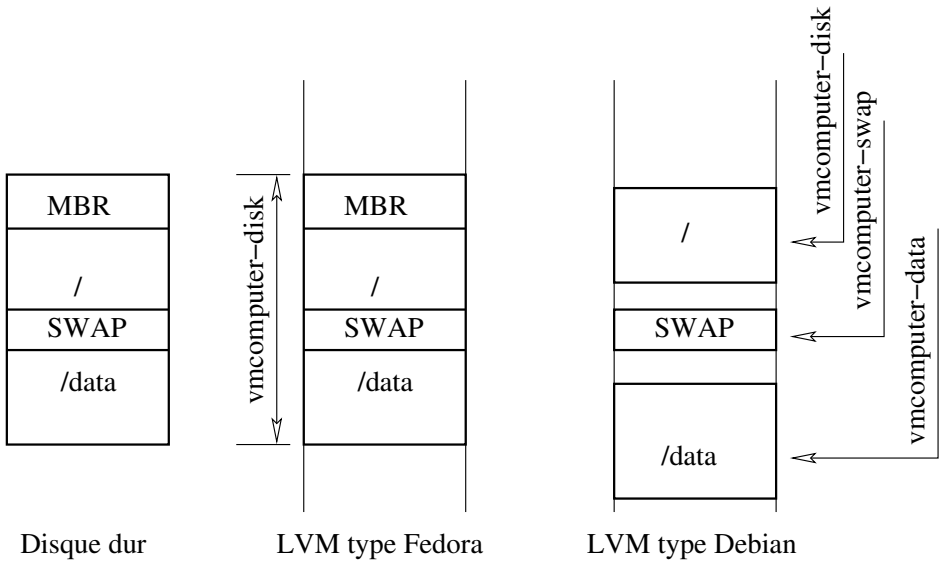
- Tous les noyaux sont dans le dom0
- Noyau du dom0 dans `/boot` et `/lib/modules`
- Noyaux des domU dans `/src/xen/boot` et `/srv/xen/lib/modules`
- Pensez à mettre à jour les modules du noyau dans les domU via `rsync`
- Pensez à mettre à jour entre les dom0 l'arborescence `/srv/xen`

Mise en production Xen sous Debian

Mise à jour d'un serveur Xen

Autres pistes : noyau dans le domU

Sous Fedora, Xen charge un utilitaire `pygrub` qui amorce le secteur MBR (et GRUB) se trouvant à la base de la partition du domU (possible sous Debian /usr/lib/xen-4.0/bin/pygrub).



Mise en production Xen sous Debian

Sauvegarde d'un domU - Cas Debian

Snapshots LVM

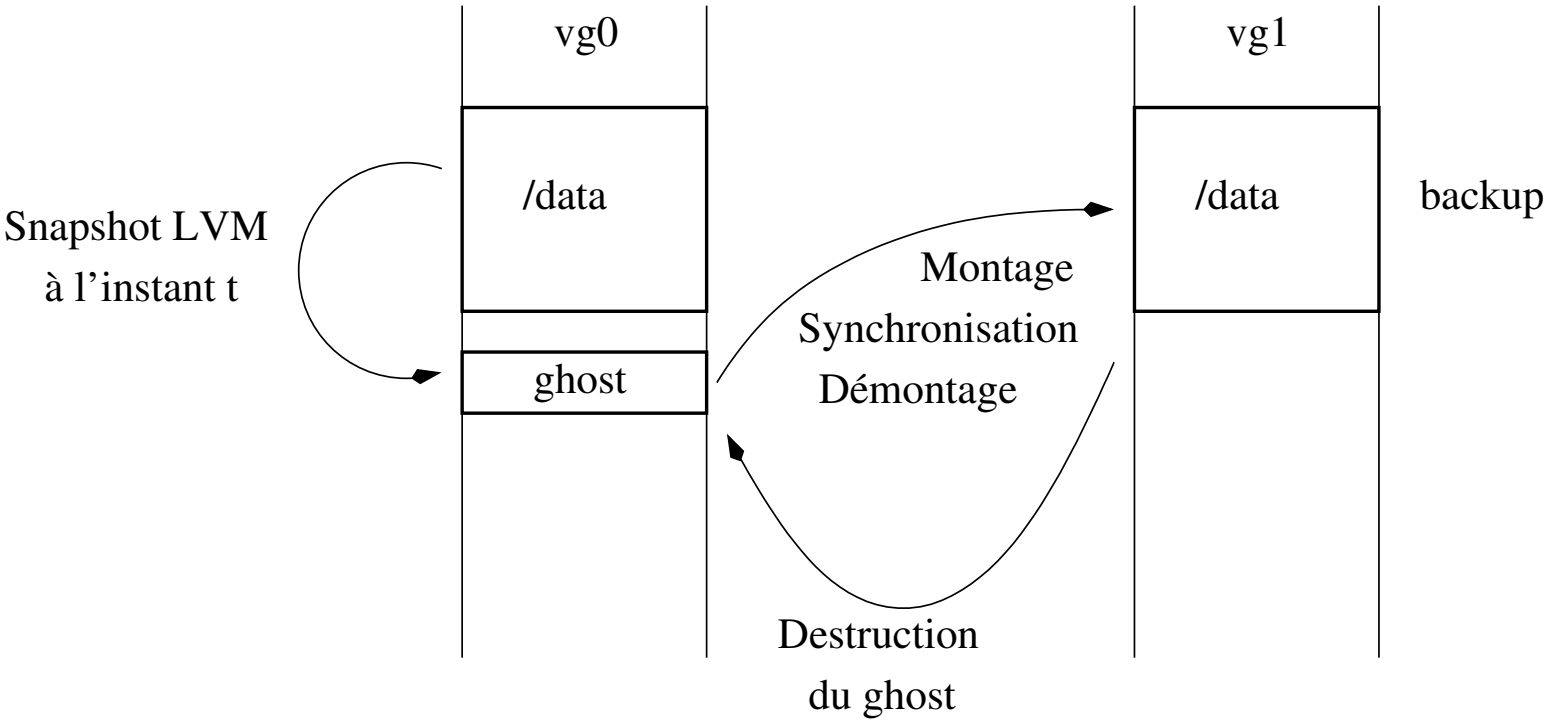
- Hypothèse : deux Volumes LVM accessibles sur le dom0 : vg0 et vg1.
- La machine dom1 est sur une partition de 6Go sur vg0
- Création d'une partition de taille identique sur vg1

```
lvcreate -L 6G -n dom1-disk vg1
mkfs.ext3 /dev/vg1/dom1-disk
```

- Création d'une partition snapshot sur vg0 de `dom1-disk` de 500M
- Copie block à block des données d'une partition sur l'autre à CHAUD sans interruption de service !

Mise en production Xen sous Debian

Sauvegarde d'un domU



Mise en production Xen sous Debian

Sauvegarde d'un domU - Cas Debian

```
listpartition='dom1-disk'
for partition in $listpartition
do
   mkdir -p /tmp/ghost
   mkdir -p /tmp/backup

   lvcreate --size 500M --snapshot --name ghost /dev/vg0/$partition

   mount /dev/vg0/ghost       /tmp/ghost
   mount /dev/vg1/$partition  /tmp/backup

   time rsync -av --delete  /tmp/ghost/ /tmp/backup/

   umount   /tmp/ghost
   umount   /tmp/backup

   lvremove --force /dev/vg0/ghost
done
```

Mise en production Xen sous Debian

Sauvegarde d'un domU - Au LEGI

Sauvegarde des domU au LEGI

- Les domU sont sauvés sur un vg0 d'une autre machine dom0 via rsync sur ssh (même script plus évolué).
- En règle général, 3 copies de chaque domU sur d'autres dom0 (reprise rapide en cas de plantage matériel).
- Synchronisation croisées toutes les nuits entre les dom0.
- Copie sur SAN pour certains domU.

Mise en production Xen sous Debian

Sauvegarde d'un domU - Cas d'une partition image (jamais testé)

Création de partition loop

```
kpartx -av /srv/xen/vm/dom1-disk.img
add map loop1p1 (253:2): 0 9687132 linear /dev/loop1 63
add map loop1p2 (253:3): 0 546210 linear /dev/loop1 9687195
```

Montage des partitions

```
mount /dev/mapper/loop1p1 /mnt
ls /mnt
bin boot cdrom dev etc home ...
```

Destruction des partitions loop

```
umount /mnt
kpartx -dv /srv/xen/vm/dom1-disk.img
del devmap : loop1p2
del devmap : loop1p1
loop deleted : /dev/loop1
```

Exercice

Marier kpartx avec les snapshot LVM et de faire pareil que précédemment (un brin plus complexe).

Mise en production Xen sous Debian

Politique de Sauvegarde

Politique de Sauvegarde

- Pas de différence majeure entre un dom0, un domU et une quelconque machine
- Il faut utiliser des outils de sauvegardes qui permettent de remonter dans le temps
- Ces outils sont complémentaires des miroirs réalisés précédemment
- `rsnapshot`, `rdiff-backup`, `dirvish`, `backuppc`, `bacula`, `amanda`...

Mise en production Xen sous Debian

Politique de Sauvegarde

backuppc

- Très facile à configurer via interface web ou via fichier
- Très bonne interface pour récupérer des fichiers (un peu complexe pour l'utilisateur lambda)
- Capitalise les fichiers communs entre les machines
- Réparti les sauvegardes au cours de la journée
- Lent (est-ce du à mon SAN ?)
- Ne marche pas chez moi sur les grosses partitions (10To par exemple)
- N'aime pas sauver des dossiers vides (par exemple, ne pas mettre /opt si vide)

Mise en production Xen sous Debian

Politique de Sauvegarde

rdiff-backup

- Rapide une fois le premier transfert réalisé
- Fonctionne sur des gros volumes
- Comme avec rsync, on construit ses scripts de sauvegarde
- Très facile à lancer
- Récupération des données sans documentation pas triviale
- Semble repartir à zéro chez moi en cas de plantage (ne repars pas de l'endroit ou il a planté comme rsync)

Mise en production Xen sous Debian

Attachement d'une baie SAN

Baie SAN AoE

- Voir exposé AoE lors de la journée Josy sur le stockage à l'automne 2010
- Création des périphérique SAN dans le dom0 (`modprobe aoe`)
- Gestion des LVM dans le dom0 (`pvscan, vgscan, lvscan`)
- Export des volumes LVM dans les domU (`/dev/sanX/diskY`)

Mise en production Xen sous Debian

Attachement d'une baie SAN

Avantages / Inconvénients

- Très simple
- Le domU ne voit rien du SAN, que des disques locaux
- Migration des données d'un disque local vers un SAN aisé lorsque la taille devient importante
- L'augmentation de taille des LVM n'est pas vu par le domU sans reboot de celui-ci (normalement possible avec les dernières version de Xen)
- Migration de la gestion des gros volumes SAN dans des containers de type LXC ?
- Migration possible de domU d'un dom0 vers un autre si ce domU est entièrement sur le SAN (peu utile en pratique)

Conclusion et Perspectives

XEN, c'est bien, mangez en !

- Peu de serveur physique, gain de place
- Permet de s'approcher du modèle : un serveur = un service
- Charge globale des serveurs physiques = 0 ! Ce sont les IO disques qui me semblent être le facteur limitant.
- On l'a souvent donné comme mort et il est toujours là
- Les fichiers de configuration sont stables d'une version à l'autre
- Le prochain noyau Linux 3.0 intègre en natif le dom0 et le domU pour la première fois (pérénité du projet)
- La plateforme XCP va être porté sur une base DEBIAN (Ubuntu)
- Il reste plein de chose à dire...