



- [Accueil](#)
- [A propos](#)
- [Nuage de Tags](#)
- [Contribuer](#)
- [Who's who](#)

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source

28 juil 2008

## XEN et l'optimisation d'espace disque

Catégorie : [Administration système](#) Tags : [GLMF](#)



Retrouvez cet article dans : [Linux Magazine 89](#)

Comment minimiser le stockage utilisé par les systèmes invités ?

Xen, le moniteur de machines para-virtuelles du monde libre permet la cohabitation de plusieurs instances de systèmes d'exploitation dans une même machine physique. Grâce aux techniques de copie-sur-écriture (Copy-On-Write ou COW), il est possible de partager les exécutables d'un système d'exploitation par plusieurs systèmes invités. Le COW permet aussi de replacer une machine virtuelle dans un état connu, de manière quasi instantanée.

Après quelques rappels sur Xen [1] et une introduction rapide des techniques de copie-sur-écriture [2], nous proposons de démarrer plusieurs machines virtuelles à partir d'une copie unique d'un système RHEL5. La structure des médias RHEL5 ayant complètement changée, nous détaillerons quelques différences notables avec les versions précédentes de cette distribution. Nous montrerons aussi comment il est possible d'exécuter une opération de type « undo » permettant de revenir rapidement à un état connu d'un système invité.

## Xen : quelques rappels

Xen est un Moniteur de Machines Virtuelles (VMM) très largement décrit dans GNU/Linux Magazine 85 et 87. Il est comparable à des produits propriétaires comme l'ESX de VMware ou l'Integrity VM de HP. Toutefois, il est différent dans le sens où, sur des processeurs n'ayant pas de technologie spécifique de virtualisation, les machines virtuelles invitées ont besoin de modifications dans leur noyau. C'est pour cela qu'on emploie le qualificatif de « paravirtualisation », quand on parle de Xen.

Xen se démarque aussi des autres produits de virtualisation par sa terminologie. Aux synonymes « machine virtuelle » et « système invité » s'ajoute le terme « domaine », qui n'est autre qu'une

machine virtuelle instanciée. Le démarrage d'un système Xen est automatiquement suivi de la création d'un premier domaine appelé « domaine0 ». Du point de vue de l'utilisateur, ce domaine est similaire à un système Linux habituel. Pourtant, il est particulier, car il contrôlera les autres domaines. Cette fonction de contrôle est assurée par le démon ~~xend~~ qui peut communiquer avec les autres domaines via des structures internes à Xen.

Les autres domaines, aussi appelés « domaines utilisateurs » ou domaineU sont instanciés (créés) par le domaine0 à partir d'un fichier de configuration fournissant de multiples informations. Par exemple, on y trouve le nom du domaine, la configuration réseau, l'endroit et le type du Virtual Block Device (VBD) contenant le système d'exploitation. Les VBD présentables aux domainesU peuvent être des fichiers, des disques, des partitions physiques locales ou distantes (SAN), des volumes logiques ou encore, à partir de Xen 3.0.3, des images au format QCoW (Qemu Copy-on-Write).

Dans la partie pratique de cet article, nous préparerons des volumes logiques situés sur un disque physique local, avant de les présenter aux différents domaines. Cette préparation se fera en étant connecté au domaine0.

## Gestionnaires de volumes logiques

La gestion des espaces de stockage utilisant les volumes logiques (Linux Magazine 53) est une méthode logicielle offrant, entre autres, des fonctionnalités de haute disponibilité (RAID). Le gestionnaire de volumes privilégié dans Linux est LVM (Logical Volume Manager) principalement développé par Sistina [5]. Il en existe d'autres comme EVMS [6], développé par IBM.

Brièvement, les gestionnaires de volumes logiques permettent de regrouper des disques, des partitions physiques ou des unités logiques (LUN). Dans ces groupes, on aura la possibilité de créer des espaces de stockage appelés « volumes ». Ces volumes pourront être étalés (ou non) sur plusieurs disques physiques (striping) ou encore placés en mode miroir. Ensuite, il suffira de transformer ces volumes en systèmes de fichiers (ext2/3, reiserfs, JFS ou encore XFS) avant de les monter dans l'arborescence.

La fonctionnalité qui nous intéresse dans ces gestionnaires de volumes est celle des volumes logiques virtuels encore appelés « clones persistants » ou « snapshots ». Le snapshot d'un volume existant n'est autre qu'une « photo » d'un volume logique. Le terme « photo » est approprié puisqu'il est possible de voir et même d'accéder à tous les fichiers à l'instant de la prise de la photo, et à cet instant uniquement, même si les fichiers ont été modifiés depuis cette prise. Les snapshots sont souvent utilisés pour effectuer des sauvegardes de systèmes en état de marche. Cela permet d'être sûr de sauvegarder des données statiques.

Comme les snapshots ne sont pas une copie de données, mais une liste de pointeurs et de blocs de stockage, ils ne prennent pas de place significative. Toutefois, les gestionnaires de volumes associent à ces listes de pointeurs et blocs, un espace physique prévu pour contenir les modifications éventuelles des fichiers référencés par le snapshot suivant la technique de copie sur écriture.

## Copie-sur-écriture

La copie-sur-écriture (copy-on-write ou COW) est un terme générique utilisé dans différents domaines comme la gestion de la mémoire virtuelle des systèmes d'exploitation. Succinctement,

un processus souhaitant modifier une ressource sans que les autres processus ayant aussi accès à cette ressource ne voient cette modification, en fera une copie privée. Si la ressource devant être modifiée n'est accessible qu'en lecture seule, le processus la recopiera dans une zone lecture-écriture afin de la modifier à son gré.

C'est une variante de ce dernier cas de figure qui nous intéresse. En effet, nous allons créer des snapshots d'un système d'exploitation complet et les présenter aux différentes VM. Toute modification de ces snapshots par la VM sera écrite dans la zone associée et réservée à cet effet. Il existe d'autres techniques de copie-sur-écriture que les snapshots de volumes logiques, comme par exemple les disques image clairsemés de type QCoW (Qemu Copy on Write sparse disk image). Les disques QCoW ont un format spécifique et sont supportés comme VBD à partir de Xen 3.0.3. Les utilitaires ~~img2qcow~~, ~~qcow-create~~ et ~~qcow2raw~~ sont fournis pour les gérer. Bien qu'on appelle ces espaces de stockage des « disques », ce sont en réalité des fichiers situés sur un système de fichier de la machine hôte et présentés comme disques aux machines virtuelles. Outre leur caractéristique de copie-sur-écriture, ces fichiers sont « troués », clairsemés de zéros dont l'espace sur le système de fichier du système hôte n'est pas comptabilisé comme utilisé. Ainsi, seul l'espace réellement utilisé apparaît à l'utilisateur. Ce type de VBD est une alternative aux snapshots des volumes logiques.

## RHEL5 : nouveau format

De la même manière que la distribution SLES10 est la première distribution de Novell incluant Xen, RedHat, pour la première fois, propose ce produit de virtualisation dans RHEL5. Ainsi, à partir d'un même média (CD ou DVD) de RHEL5, il sera possible de générer soit un serveur classique sans spécificité particulière, soit un moniteur de machines virtuelles Xen, soit un système invité Xen. La matrice des différentes possibilités est plus complexe, puisque cette nouvelle version inclut aussi des fonctions de mise en grappe (cluster), avec ou sans l'utilisation de système multi-chemin d'accès au stockage dénommé « ClusterStorage ».

Afin de permettre à l'utilisateur de sélectionner simplement les combinaisons de son choix, RedHat a complètement modifié la structure des répertoires sur les médias de cette nouvelle distribution. Très tôt dans le processus d'installation, l'utilisateur doit entrer une clé d'activation (figure 1) qui conditionnera l'affichage des différentes catégories de produits à installer.



Figure 1 : Clé d'activation RHEL5 (bêta) permettant de générer un VMM Xen

Si, durant la génération d'un système hôte, cette clé d'activation contient le caractère majuscule « V » comme indiqué sur la figure 1, on trouvera plus tard la catégorie de paquets « Virtualisation » aux côtés des catégories « système de base », « langages », « applications »... En sélectionnant les paquetages de cette nouvelle catégorie, on générera un moniteur Xen (figure 2).

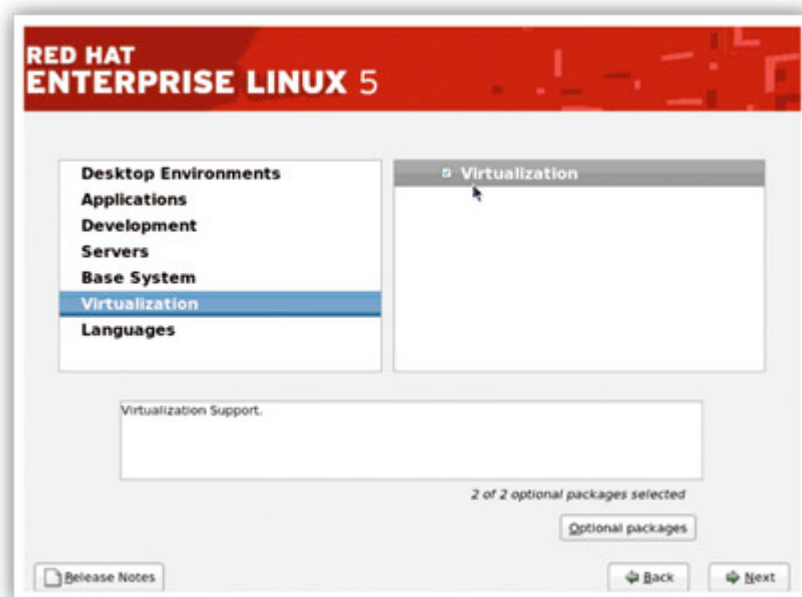


Figure 2 : Catégories contextuelles de paquetages dans RHEL5

Hormis le système de clé d'activation comportant ces caractères significatifs, le reste du processus de génération d'un système hôte de type RHEL5 est globalement semblable à celui de RHEL4. Prenons quand même soin d'activer une méthode de partage de fichier à distance comme le serveur NFS ou un serveur FTP, car nous aurons besoin de cette fonctionnalité pour la génération des systèmes invités (figure 3).

# Partage des binaires : méthodologie

Considérant que notre serveur Xen est prêt à héberger des systèmes invités, nous devons maintenant identifier en son sein, un espace de stockage permettant de contenir la distribution d'un système invité. Bien qu'il soit possible d'utiliser les fichiers et exécutables du domaine0 pour constituer nos futures machines virtuelles, nous ne choisirons pas cette option. Une telle d'architecture est pourtant valide, puisque utilisée dans le CD de démonstration fourni par XenSource [3]. Ce live CD met en commun les fichiers du domaine0 et des domaines utilisateurs grâce au système de fichier unionfs [4], qui permet la copie-sur-écriture. Ainsi, il est possible de démarrer un nombre considérable de domaines en plus du domaine0, à partir d'un CD de seulement 650 MO.

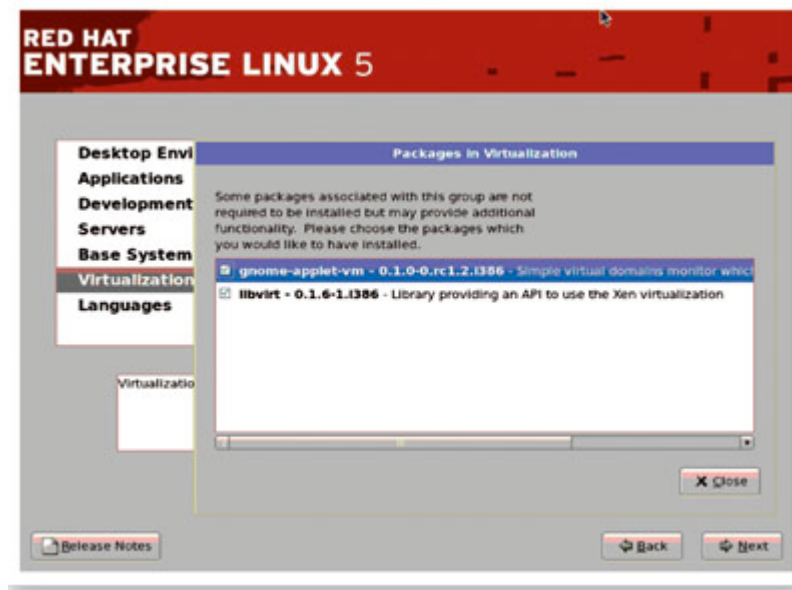


Figure 3 : Paquetages de virtualisation optionnels dans RHEL5

Afin de simplifier notre exercice, nous avons choisi de séparer complètement les espaces de fichiers du domaine0, et ceux de ses systèmes invités. Nous allons créer un volume logique de type LVM2 dans lequel nous générerons un système invité. Ensuite, nous créerons des snapshots de ce volume que nous présenterons en mode d'écriture, aux différentes machines virtuelles. Le choix du gestionnaire de volumes (volume manager) LVM2 est naturel, puisqu'il est installé par défaut dans les systèmes RHEL, qu'il dispose de la fonctionnalité des snapshots et qu'il est aussi utilisé par défaut lors de la génération du serveur (Xen ou autre). Rien n'empêche l'utilisation d'un autre gestionnaire de volumes comme EVMS, si ce n'est qu'il nécessite un petit travail de rapatriement de paquetages non disponibles en standard dans RHEL5.

## Création d'un volume logique pour le système invité de référence

Notre serveur Xen possède plusieurs disques physiques et la commande `fdisk -l` indique que le disque de 18 GO ~~de/sd~~ est libre et donc utilisable dans sa totalité. Initialisons-le pour ensuite le

placer dans un groupe de type LVM2 :

```
Xen# pvcreate /dev/sdd
Physical volume « /dev/sdd » successfully created
```

Ensuite, nous créons un groupe de volumes appelé `vg` et dans lequel nous plaçons le disque précédemment formaté :

```
Xen# vgcreate vg /dev/sdd
Volume group "vg" successfully created
```

La dernière étape consiste à constituer un volume logique assez large dans ce groupe de volumes, pour héberger un serveur RHEL5. Nous l'appellerons `vm` et lui affecterons une taille de 5 GO :

```
Xen# lvcreate -L5120 -n vm vg
Logical volume "vm" created
```

Grâce aux trois commandes précédentes, nous disposons d'un espace de stockage de 5 GO de type LVM2, permettant d'héberger un système d'exploitation de type RHEL5. Après avoir généré un tel système dans cet espace, nous le prendrons en « photo » (snapshot) et l'utiliserons comme système de base pour plusieurs machines virtuelles.

## Génération du système invité de référence

À partir de RHEL5-beta1 (septembre 2006), le processus de génération d'un système Xen invité diffère sensiblement de celui décrit dans la documentation de Xen 3.0 et téléchargeable chez XenSource. Cette documentation n'est en effet valable que pour RHEL4 et les versions de développement de RHEL5 d'avant septembre 2006. De ce fait, nous souhaitons détailler cette étape importante.

Tout d'abord, nous nous placerons dans un contexte défavorable où la distribution est présente sous la forme de cinq fichiers `iso` stockés dans le répertoire `/var/ftp/kits`. Nous devons les « concaténer » dans un répertoire unique. En effet, il est nécessaire de disposer de la distribution complète et « éclatée », en un lieu unique de l'arborescence.

Le petit script suivant permet de regrouper le contenu des cinq disques en un endroit unique.

Notez que le `disc1` est différent des autres, car il comprend le répertoire `images`, qui contient les amorces de démarrage et les fichiers de description de la distribution.

```
RNAME=/var/ftp/kits
DNAME=RHEL5-Server-20060927.0-i386
mkdir $RNAME/$DNAME
for i in 1 2 3 4 5 ; do
    mkdir $RNAME/disc$I
    mount -o loop=/dev/loop$i $RNAME/$DNAME-disc${i}-ftp.iso $RNAME/disc$I
    [ $i == 1 ] && cp -ax $RNAME/disc1/images $RNAME/$DNAME
    cp -ax disc${i}/{Cluster,ClusterStorage,VT,Server} $RNAME/$DNAME
    umount $RNAME/disc$I
done
```

Le cœur du script précédent est la commande `cp -ax` qui copie les répertoires `Cluster`, `ClusterStorage`, `VT` et `Server` de chacun des fichiers `iso` montés en « loopback » vers le répertoire final. Le sous-répertoire `Server` contient les paquetages principaux de la distribution RHEL5. Les sous-répertoires `Cluster`, `ClusterStorage` et `VT` contiennent les paquetages correspondant aux options C, S, V pouvant être placées dans la boîte de dialogue de la clé d'activation.

Ainsi, ces quelques lignes de shell nous permettront de générer un système invité de référence, à partir de la distribution éclatée dans le répertoire `/var/ftp/kits/RHEL5-Server-20060912.2-i386`. Cette génération pourra être effectuée soit via la ligne de commande et le script `xenguest-install`, soit grâce à l'assistant de création de l'interface graphique `virt-manager` [7] intégrée à Gnome (figure 4, page suivante).



Figure 4 : Interface graphique de gestion des VM intégrée à Gnome et assistant de création

Ce gestionnaire graphique de machines virtuelles, spécifique à RedHat n'étant pas présent dans notre version encore en développement de RHEL5, nous avons dû le télécharger [7], ainsi que ses nombreuses dépendances. Il est toutefois prévu de l'intégrer dans la version finale.

L'assistant graphique de création des machines virtuelles utilise, au final, le script interactif `xenguest-install` qui accepte aussi toutes les options nécessaires à la création d'une nouvelle VM. C'est d'ailleurs lui que nous allons utiliser pour créer notre premier système invité, grâce à la commande :

```
Xen# /usr/sbin/xenguest-install --name=vm --file=/dev/vg/vm \
--ram=512 --location=ftp://servername/kits/$DNAME --nographics
```

Cette commande créera une machine virtuelle nommée `vm`, possédant 512 MO de mémoire à la « mise sous tension ». Xen présentera le volume logique `/dev/vg/vm` au système invité afin qu'il puisse y placer son système d'exploitation en utilisant le protocole `ftp` à partir du serveur `servername`. Bien que dans notre cas, `servername` corresponde au domaine0 de notre serveur Xen, il est nécessaire de fournir le nom complet ou son adresse IP et, en aucun cas, `localhost`. En effet, le paramètre `--location` sera transmis au système invité qui devra contacter via le réseau `servername` pour télécharger et installer le système d'exploitation. Comme indiqué plus haut, on aura pris soin de démarrer le service `ftp` sur `servername`. Si l'on souhaite utiliser le protocole `nfs` pour cette génération, la syntaxe est `--location='nfs:servername:repertoire/de/la/distro'`. L'envoi de la commande `xenguest-install` précédente génère le fichier de configuration `/etc/xen/vm`, démarre le domaine `vm` et présente à l'utilisateur la console `/dev/console`. Il s'ensuit une installation classique de RHEL5 au sein de ce domaine. Il est possible à tout moment de quitter cette console pour revenir à l'invite du domaine0 grâce à la séquence d'échappement `ctrl-]`, sans compromettre l'installation. Pour retourner à la console du domaine `vm`, il suffit d'envoyer la commande `xm console vm`.

Une fois la machine virtuelle générée et personnalisée à souhait, nous devons l'arrêter pour en faire plusieurs snapshots. Cet arrêt peut se faire dans le système invité (`init 0`) ou à partir du domaine0 grâce à la commande `xm shutdown vm`. Il est possible d'être plus violent avec `xm`

~~destroy vm.~~

## Création des copies du volume de référence

Les systèmes invités que nous allons créer n'utiliseront pas directement le volume logique du domaine ~~vm~~, mais ses snapshots. De ce fait, ~~vm~~ ne sera pas modifié. Nous souhaitons quand même éviter tout risque de modification ou de corruption en le plaçant dans un mode de lecture seule :

```
Xen# lvchange pr /dev/vg/vm
```

Notons que pour revenir à un mode lecture-écriture, il suffit d'envoyer la commande ~~lvchange p~~ ~~rw~~ /dev/vg/vm.

En examinant la taille disponible sur notre groupe de volumes, nous constatons qu'il reste un peu plus de 8 GO :

```
Xen# vdisplay /dev/vg | grep Free
Free PE / Size      2293 / 8.96 GB
```

Si nous affectons un espace de 1 GO par snapshot pour stocker les différences par rapport au volume de référence, nous pouvons donc créer 8 snapshots, que nous dénommerons ~~snap\$i-vm~~ (i allant de 1 à 8). Ces snapshots constitueront les VBD des futurs domaines ~~vm1, vm2, ..., vm8~~. La petite boucle suivante effectue la création des huit « photos » avec un tampon associé de 1 GO :

```
Xen# for i in 1 2 3 4 5 6 7 8 ; do
    lvcreate -s -L1024M -n snap$i-vm /dev/vg/vm
done
```

Les huit machines virtuelles devant être identiques, la création d'un fichier de configuration générique unique est possible. Nous placerons le numéro d'ordre de chaque machine dans la variable python « ~~vmid~~ » du fichier ~~/etc/xen/xmdefconfig~~. La commande ~~xm~~ utilisée pour créer les domaines est prévue pour le consulter sans option supplémentaire. Ce fichier comporte une première partie de description des variables et de leurs valeurs limites, ainsi que des règles d'usage. Ensuite, on trouve la description proprement dite des machines virtuelles :

```
Xen# cat /etc/xen/xmdefconfig
# -*- mode: python; -*-
def vmid_check(var, val):
    val = int(val)
    if val <= 0:
        raise ValueError
    if val > 8:
        raise ValueError
    return val
xm_vars.var('vmid',
            use="Virtual machine id. Integer greater than 0 and less than 9.",
            check=vmid_check)
# Check the defined variables have valid values..
xm_vars.check()
#-----
disk = [ 'phy:/dev/vg/snap%d-vm,xvda,w' % vmid ]
vif = [ 'mac=00:16:3e:6b:54:e%d, bridge=xenbr0' % vmid ]
bootloader="/usr/bin/pygrub"
memory = "512"
name = "vm%d" % vmid
uuid = "12289a7e-b698-35bf-41a4-883efd880a4%d" % vmid
```



```
vcpus = 2
```

La variable `disk` décrit le `VDB` présenté à la VM (`/dev/vg/snap%d-vm`), le nom qu'il aura au sein de cette VM (`xvda`) et son mode (`write`). Le `%d` sera remplacé par la variable `vmid` qui varie de 1 à 8. Ensuite, nous trouvons la variable `vif` décrivant une interface réseau virtuelle. Cette interface virtuelle, connectée au pont `xenbr0` permettra l'accès au réseau extérieur. La variable `bootloader` contient le chemin complet d'un script python mimant `grub` (figure 5) et qui propose à l'utilisateur la liste des noyaux utilisables au sein de notre VM.

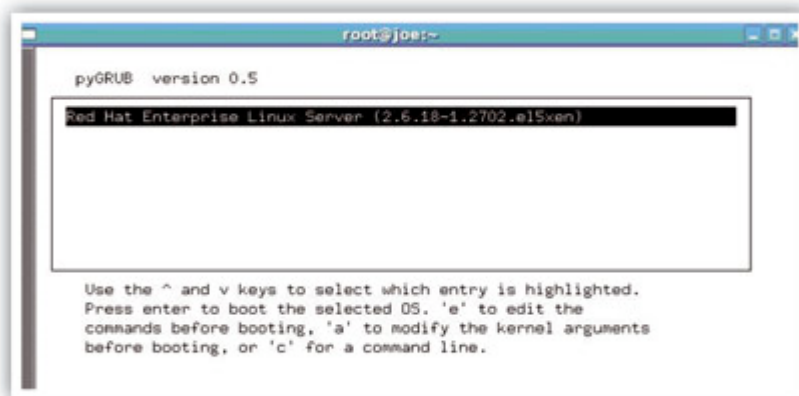


Figure 5 : Le bootloader pyGRUB

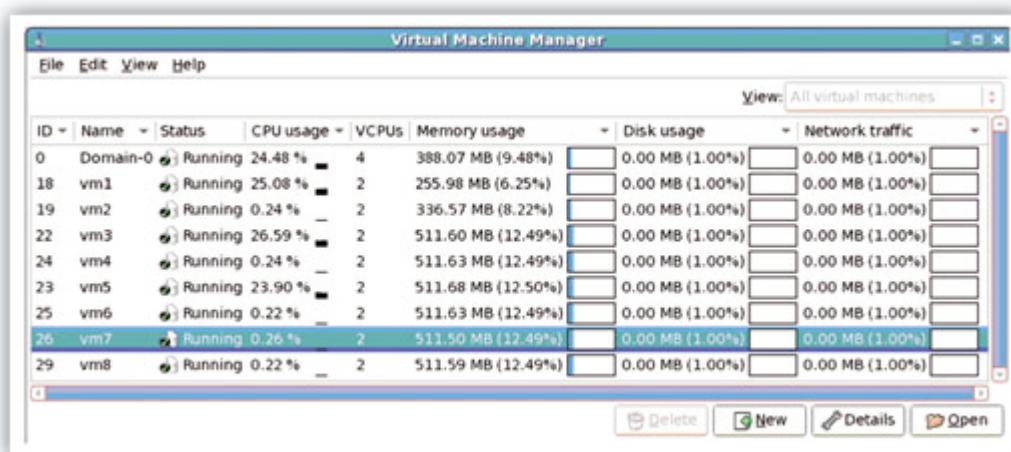
Pour créer un domaine et visualiser le démarrage de la VM associée, on enverra la commande :

```
Xen# xm create -c vmid=1
```

L'option `-e` de cette commande indique que l'on souhaite se connecter à la console du domaine. Pour la quitter, il faudra envoyer la séquence `ctrl-]`. Il est aussi possible de démarrer les huit VM simultanément grâce à la petite boucle :

```
for i in 1 2 3 4 5 6 7 8 ; do
xm create vmid=$i
done
```

La visualisation en temps réel de l'évolution de ces domaines en mode texte se fait avec `xentop`, ou alors en mode graphique grâce au `virt-manager` (figure 6).



*Figure 6 : Affichage principal du gestionnaire graphique de machines virtuelles*

## Remise en l'état

Lors de la création des snapshots, nous avons associé une zone de 1 GO pour stocker les copies en écriture de la VM, qui, encore une fois, ne peuvent pas se faire sur le volume logique original. Par exemple, si nous créons un fichier ~~/tmp/toto.txt~~, il sera physiquement stocké dans cette zone de 1 GO.

Les commandes ~~lvdisplay~~ et ~~lvs~~ nous renseignent sur le taux de remplissage de cette zone, respectivement sous les dénominations « allocated to snapshot » et « snap% ». Il est fortement conseillé de surveiller de près ce taux de remplissage car, en cas de remplissage complet de cette zone, LVM2 place automatiquement le snapshot en lecture-seule, ce qui rend le système d'exploitation dans un état difficilement gérable.

Pour « purger » cette zone et donc se replacer dans l'état initial du système invité, nous allons arrêter le domaine, effacer le snapshot et le recréer. Toutes ces commandes pouvant être effectuées à partir du domaine0 :

```
Xen# xm shutdown vm8
Xen# lvremove -f /dev/vg/snap8-vm
Xen# lvcreate -s -L1024M -n snap8-vm /dev/vg/vm
Xen# xm create vmid=8
```

Ainsi, nous venons d'effectuer pas à pas une opération de type « undo » en utilisant une fonctionnalité du gestionnaire de volumes.

## Conclusion

L'apparition d'une nouvelle technologie suscite toujours de nouveaux besoins. Dans le cas de la virtualisation, l'utilisateur final constatera rapidement des limitations au niveau du stockage. Son souhait le plus vif sera donc de mutualiser ce genre de ressource, au même titre que la puissance CPU, les entrées/sorties et la mémoire. Xen, associé aux solutions de Copie en Écriture récentes et présentes dans la plupart des distributions Linux, permet d'exaucer ce vœu. Une conséquence heureuse de cette mutualisation est la possibilité de disposer d'une manière simple pour replacer des machines virtuelles dans un état antérieur connu. Espérons que cette fonctionnalité fera le bonheur des équipes de test des logiciels du monde libre et même des autres...

### Références

- [1] <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>
- [2] <http://fr.wikipedia.org/wiki/Copy-On-Write>
- [3] <http://www.xensource.com/>
- [4] <http://www.filesystems.org/project-unionfs.html>
- [5] <http://www.sistina.com/lvm/>
- [6] <http://evms.sourceforge.net>
- [7] <http://virt-manager.et.redhat.com/download.html>

Retrouvez cet article dans : [Linux Magazine 89](#)

Posté par ([La rédaction](#)) | Signature : François Donzé | Article paru dans



## Laissez une réponse

Vous devez avoir ouvert une [session](#) pour écrire un commentaire.

« [Précédent](#) [Aller au contenu](#) »

[Identifiez-vous](#)

[Inscription](#)

[S'abonner à UNIX Garden](#)

## • Articles de 1ère page

- [Pear et les librairies PHP](#)
- [FreeDOS](#)
- [Mise en œuvre d'une passerelle Internet sous Linux](#)
- [Les pseudo-classes en CSS](#)
- [GNU/Linux Magazine N°108 - Septembre 2008 - Chez votre marchand de journaux](#)
- [Linux Pratique N°49 - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Donner du style à son texte : utiliser les lettrines](#)
- [Installer un serveur Syslog](#)
- [Quatre serveurs FTP hyper sécurisés avec vsftpd](#)
- [Le point sur... Les formats d'images sur le web](#)



Actuellement en kiosque :

## • Il y a actuellement

•

**718** articles/billets en ligne.

Recherche

## • Catégories

- - [Administration réseau](#)
  - [Administration système](#)
  - [Agenda-Interview](#)
  - [Audio-vidéo](#)
  - [Bureautique](#)
  - [Comprendre](#)
  - [Distribution](#)
  - [Embarqué](#)
  - [Environnement de bureau](#)
  - [Graphisme](#)
  - [Jeux](#)
  - [Matériel](#)
  - [News](#)
  - [Programmation](#)
  - [Réfléchir](#)
  - [Sécurité](#)
  - [Utilitaires](#)
  - [Web](#)

## • Archives

- - [août 2008](#)
  - [juillet 2008](#)

- [juin 2008](#)
- [mai 2008](#)
- [avril 2008](#)
- [mars 2008](#)
- [février 2008](#)
- [janvier 2008](#)
- [décembre 2007](#)
- [novembre 2007](#)
- [février 2007](#)

## • [GNU/Linux Magazine](#)

- ◦ [Ce billet est hors ligne montrer/masquer GNU/Linux Magazine 108 - Septembre 2008 - Chez votre marchand de journaux](#)
- [Edito : GNU/Linux Magazine 108](#)
- [GNU/Linux Magazine HS 38 - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : GNU/Linux Magazine HS 38](#)
- [GNU/Linux Magazine 107 - Juillet/Août 2008 - Chez votre marchand de journaux](#)

## • [GNU/Linux Pratique](#)

- ◦ [Linux Pratique N°49 -Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : Linux Pratique N°49](#)
- [À télécharger : Les fichiers du Cahier Web de Linux Pratique n°49](#)
- [Linux Pratique Essentiel N°3 - Août/Septembre 2008 - Chez votre marchand de journaux](#)
- [Edito : Linux Pratique Essentiel N°3](#)

## • [MISC Magazine](#)

- ◦ [Misc 38 : Codes Malicieux, quoi de neuf ? - Juillet/Août 2008 - Chez votre marchand de journaux](#)
- [Edito : Misc 38](#)
- [Références de l'article « Détection de malware par analyse système » d'Arnaud Pilon paru dans MISC 38](#)
- [Références de l'article « La sécurité des communications vocales \(3\) : techniques numériques » d'Éric Filiol paru dans MISC 38](#)
- [Misc 37 : Déni de service - Mai/Juin 2008 - Chez votre marchand de journaux](#)

© 2007 - 2008 [UNIX Garden](#). Tous droits réservés .