Formation virtualisation

- Vserver
- Xen

JT-SIARS - DR 12 CNRS

Avril 2008

M. Libes & T. Dostes

Plan virtualisation avec Vserver

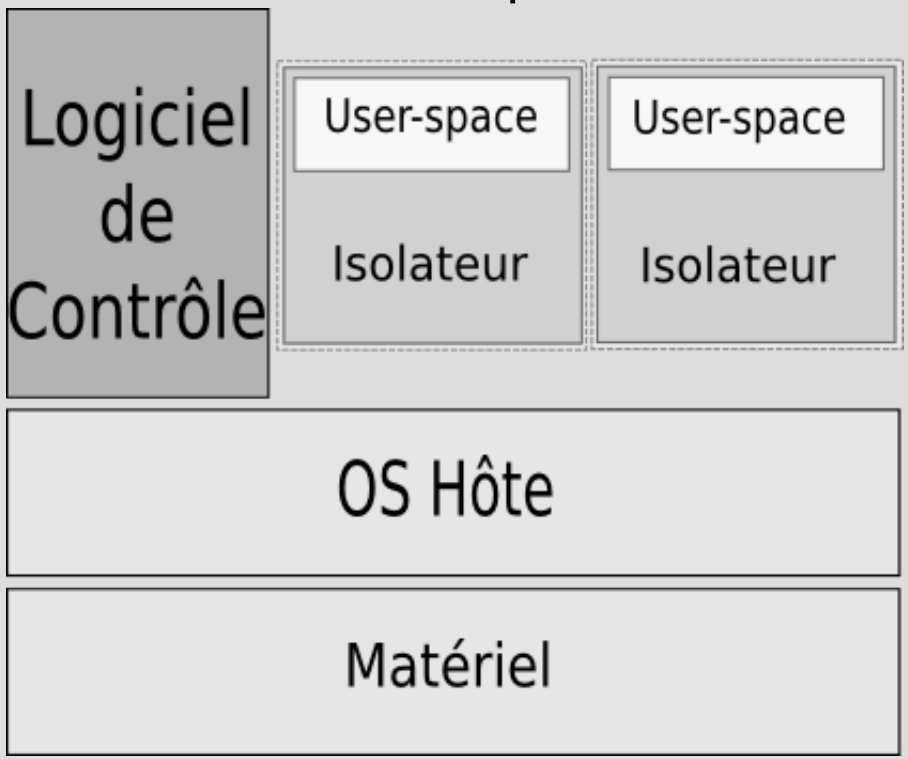
- Principes de la virtualisation avec Vserver
- Mise en oeuvre - Installation
  - avec les paquetages Debian
  - compilation des sources à la main
- Création d'une machine virtuelle
  - from scratch
  - par copie ou duplication de VM
- les fichiers de configuration
- Utilisation : lancement, arrêt
  - les commandes principales
- les « capabilities », context, Limits
- Les vserver-tools

Vserver : Principe de base

- Vserver = Virtualisation au niveau Kernel : Il s'agit d'un partitionnement logique au niveau des ressources processus, réseau et « file system »
- C'est le noyau du système d'exploitation qui fait une ***isolation*** entre des machines logiques, tout comme il isole déjà les processus entre eux.
  - isolation des « processus » et
  - mise en cage (chroot) des différents « file system » des machines virtuelles.
- Dans ce cas, il n'y a pas d'« émulation » à proprement parler comment dans d'autres système de virtualisation.

Virtualisation par Isolation

- Un isolateur est un logiciel permettant d'isoler l'exécution des applications dans des contextes ou zones d'exécution. L'isolateur permet ainsi de faire tourner plusieurs fois la même application (à base d'un ou plusieurs logiciels) prévue pour ne tourner qu'à une seule instance par machine.



Vserver : Principe de base

- Linux-VServer est un mécanisme d'isolation des ressources système sur le PC.
  - file system, CPU, adresse réseau et mémoire sont exécuté dans un contexte système particulier
- Chaque machine virtuelle sous le seul contrôle du noyau de la machine hôte a accès des ressources isolées dans un contexte
- chaque processus de chaque machine virtuelle ne peut pas faire de déni de service hors de son système de partition

Vserver : Principe de base

- Un Vserver repose sur une modification du noyau Linux sous la forme d'un patch. Le code n'étant pas actuellement intégré dans le noyau officiel.
- Ce patch est accompagné d'utilitaires indispensables `(dpkg -L util-vserver)` pour administrer les machines virtuelles.
  - Debian intègre depuis la version Sarge les utilitaires de gestion des vservers. (intégration simple).
- Concrètement, un vserver fonctionne par un système de contexte supplémentaire ajouté à chaque processus. C'est un système de *virtualisation léger et peu intrusif*.
- La machine physique démarre le noyau Linux. Tous les processus lancés par ce noyau (à partir d'init) le sont avec un contexte 0, celui dédié à la machine hôte.
- => les machines virtuelles n'ont PAS de noyau

Vserver avantages

- Les Virtual serveurs partagent les mêmes appels systèmes (un seul noyau) et il n'y a donc aucune charge supplémentaire due a une émulation
- Les process d'une machine virtuelle sont des processus normaux d'un seul et même noyau. Les E/S sont donc plus efficaces que sur un système qui tournerait à travers une émulation
- installation aisée : pas besoin de clônage ou image disque de machine. Il suffit de copier un file system pour installer une machine virtuelle
- l'adressage réseau des machines virtuelles est basé sur une isolation. Il n'y a pas de surcharge

Vserver avantages

- C'est la performance qui rend les vservers attractifs.
  - il ne s'agit pas d'un PC virtuel, mais plutôt de "serveurs Linux" virtuels.
- Cela peut simplement être vu comme un chroot amélioré.
- Avantages = performances natives (pas de perte mesurable). A part la gestion du contexte, un processus dans un vserver a les mêmes caractéristiques qu'un processus d'une machine Linux standard.
- Consommation mémoire légère (la mémoire est mutualisée entre le serveur hôte et les vservers et la mémoire demandée à l'hôte est celle réellement utilisée par les processus du vserver).
- La possibilité de mutualisation est donc ici très importante ; il est possible de déployer plusieurs dizaines de vservers sur un serveur actuel correctement taillé.

Vserver avantages

- Virtualisation légère :
- Performance native, aucun ralentissement mesurable.
- Très économe en ressource :
  - machine PIII avec 256 Mo de RAM
  - 10 vservers déployés
- Les règles IPtables (firewall interne) ne peuvent être manipulées dans un serveur virtuel : sécurité renforcée
- Différentes distributions peuvent fonctionner en parallèle
  - (Debian, Fedora, Ubuntu...)

Vserver : inconvénients

- Ne tourne que sous Linux et nécessite un patch du kernel
- Pas supporté au niveau des distributions
- Kernel unique pour machine hôte et machine virtuelle
  - hôte et machines virtuelles partage le même noyau et donc les bugs potentiels et failles de sécurité
- Le réseau est basé sur une isolation :
  - plusieurs adresses sur une même carte :
  - cela empêche chaque machine virtuelle d'avoir son propre routage , ou son propre pare-feu (pas d'iptables sur machines virtuelles)
- La couche réseau n'est pas entièrement virtualisée:
  - Pas de Ipv6, ni d'interface localhost, pas de multicast

Vserver : inconvénients

- certains appels systèmes propres au hardware ne peuvent pas être virtualisés : horloge, /proc /sys
- pas de possibilité de dédier de la bande passante en E/S (#QoS) aux différentes machines virtuelles
- Certains applicatifs sont gênés et peuvent ne pas fonctionner correctement, soit parce qu'ils utilisent des « capabilities » interdites (bind), soit à cause de la couche réseau.
- Pas de NFS noyau

Vserver : inconvénients

- Vserver a l'inconvénient de ses avantages, c'est à dire sa légèreté...
  - le noyau est partagé par toutes les instances de serveurs, et tous les drivers et couches de communication ne sont pas virtualisés ; les conséquences sont multiples :
- Certains programmes nécessitant des privilèges élevés, ou manipulant directement le matériel ne pourront fonctionner correctement.
  - Par exemple, le déploiement d'un serveur NFS kernel ne peut se faire qu'au niveau de l'hôte principal. Pas plus que démarrer un serveur X window n'est possible simplement...
- La mutualisation de la couche réseau peut aussi entraîner des problèmes de routage complexes et empêche IPv6 de fonctionner.

Vserver et réseau ?

- Les processus sont les seuls éléments *virtualisés*.
- Il n'y a pas de virtualisation des couches réseaux ou de stockage.
- A l'intérieur d'un vserver, il est impossible de:
  - manipuler les adresses IP de l'interface,
  - de manipuler iptables ,
  - ou encore d'utiliser mknod.
- Comme un vserver hérite d'une tâche unique, il est possible au lancement du serveur virtuel, de le limiter en nombre de processus, de lui donner une priorité (nice)...
- Cela est implémenté en interne via la commande ***ulimit***. Il est facile de voir l'état des serveurs sur la machine hôte via la commande ***vserver-stat***

Recommandations : sécurité

- La sécurité du serveur maître est primordiale : Si ce serveur se fait attaquer, tous les serveurs virtuels sont compromis.
- Ne mettre que le strict minimum dans le serveur maître :
  - SSH, outils de sécurité et de surveillance, utilitaires de la solution de virtualisation. Pas de démons réseau...
- Dans les serveurs virtuels, ne mettre que les services nécessaires
- Ne pas hésiter à mettre un firewall applicatif entre serveurs virtuels et serveur maître. Un serveur virtuel compromis restera cloisonné, et l'attaque en restera là
- Si la machine physique est en panne, tous les serveurs virtuels sont en panne : Il est important de redonder les services critiques
- Il faut surveiller l'état de charge de la machine

Vserver : mise en oeuvre

Le principe de base est de :

1. patcher un kernel linux (vanilla kernel) avec les patch de vserver
    - NB: certaines distributions (debian) proposent d'utiliser un kernel déjà patché ... et de démarrer sur ce nouveau noyau ==> plus facile
2. installer le paquetage « util-vserver » permettant de créer et lancer, configurer adminsitrer de nouvelles machines virtuelles
3. il faut récupérer ou se créer un root file system pour la nouvelle machine virtuelle..
    - on démarre avec un File System et un ensemble de paquetages réduits au strict minimum

Vserver : mise en oeuvre

- Préparation de la machine hôte Debian
- Principe de base : la machine hôte n'a pas vocation à faire fonctionner autre chose que les
  - services de base (ssh, MTA, supervision) et
  - héberger les Vserveurs
- Allouer un minimum d'espace disque
- Installer uniquement le strict nécessaire

Installation paquetages Vserver Debian

- $ apt-get install linux-image-vserver-686 util-vserver vserver-debiantools

```
Les paquets supplémentaires suivants seront installés :
  debootstrap libbeecrypt6 linux-image-2.6-vserver-686
  linux-image-2.6.18-6-vserver-686

Paquets suggérés :

  linux-doc-2.6.18 kernel-patch-vserver vlan yum

Les NOUVEAUX paquets suivants seront installés :
  debootstrap libbeecrypt6 linux-image-2.6-vserver-686
  linux-image-2.6.18-6-vserver-686 linux-image-vserver-686 util-
vserver
  vserver-debiantools

0 mis à jour, 7 nouvellement installés, 0 à enlever et 2 non mis
à jour.....
```

Vserver : un noyau et des modules

- $ dpkg -L linux-image-2.6-vserver-686
- un noyau de boot avec initrd
  - `/boot/config-2.6.18-6-vserver-686`
  - `/boot/vmlinuz-2.6.18-6-vserver-686`
  - `/boot/System.map-2.6.18-6-vserver-686`
- un ensemble de modules du kernel
  - `/lib/modules`
  - `/lib/modules/2.6.18-6-vserver-686`
  - `/lib/modules/2.6.18-6-vserver-686/modules.dep`
  - `/lib/modules/2.6.18-6-vserver-686/modules.alias`
  - `/lib/modules/2.6.18-6-vserver-686/modules.ieee1394map`
  - `/lib/modules/2.6.18-6-vserver-686/modules.pcimap`

Vserver : les utilitaires standards

- dpkg -L util-vserver
- fichier de config
  - `/etc/vservers`
    - `ls -l /etc/vservers/.defaults/`**`vdirbase`**
    - `/etc/vservers/.defaults/vdirbase -> /var/lib/vservers`
- un ensemble de binaires utilitaires
  - `/usr/sbin/vserver-stat`
  - `/usr/sbin/vserver-info`
  - `/usr/sbin/vserver`
  - `/usr/sbin/vtop`
  - `/usr/sbin/vps`

Installation paquetages Vserver

- On se retrouve avec les principaux répertoires suivants :
  - ***/etc/vservers :*** le répertoire des fichiers de configuration des vservers
  - ***/etc/vservers.conf*** : le fichier de configuration basique de l'outil util-vserver
  - ***/var/lib/vservers*** : le répertoire contenant les « file system » des vservers
  - ***/usr/sbin/vserver*** : l'executable principal pour interagir, démarrer, stopper, construire, détruire... entrer dans un vserver
  - ***/usr/lib/util-vserver*** : les scripts, fonctions principales.

Vserver : les utilitaires Debian

- $ dpkg -L vserver-debiantools
  - **## duplique un vserver existant**
    - /usr/sbin/**dupvserver**
  - **## crée nouveau vserver**
    - /usr/sbin/**newvserver**
  - **## créer un nouveau vserver par téléchargement via ftp et http**
    - /usr/sbin/newnfsvserver
  - /usr/sbin/stripserver

modification du fichier de boot

- on trouve un nouveau noyau de boot dans grub ou lilo

```
title      Debian GNU/Linux, kernel 2.6.18-6-vserver-686
root       (hd0,4)
kernel     /boot/vmlinuz-2.6.18-6-vserver-686 root=/dev/hda5 ro
initrd     /boot/initrd.img-2.6.18-6-vserver-686
savedefault
```

- On reboote sur le nouveau noyau « virtualisé »

```
$ uname -a
Linux pc-momo 2.6.18-6-vserver-686 #1 SMP Sun Feb 10 22:30:33
   UTC 2008 i686 GNU/Linux
```

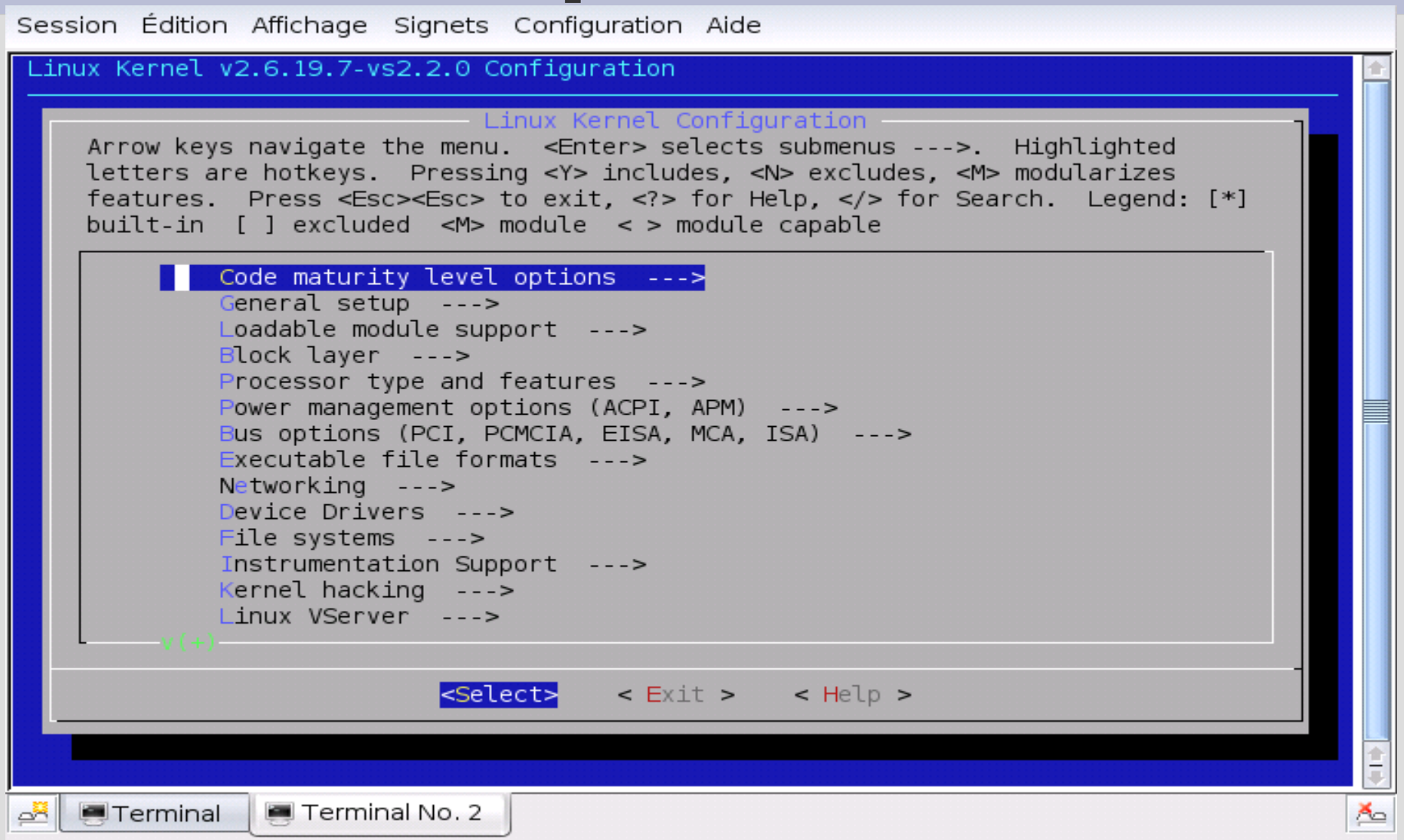
Installation : compilation noyau

- Téléchargement des sources du kernel Linux dans /usr/src :
  - `wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.19.7.tar.gz`
- Décompression de l'archive :
  - `# cd /usr/src/ ; tar xvzf linux-2.6.19.7.tar.gz`
- Téléchargement du patch Vserver *(relatif à la même version du kernel !)* :
  - `wget http://ftp.linux-vserver.org/pub/kernel/vs2.2/patch-2.6.19.7-vs2.2.0.diff`
- Créer un lien pour patcher le noyau :
  - `cp -la linux-2.6.19.7 linux-2.6.19.7-vs2.2.0`

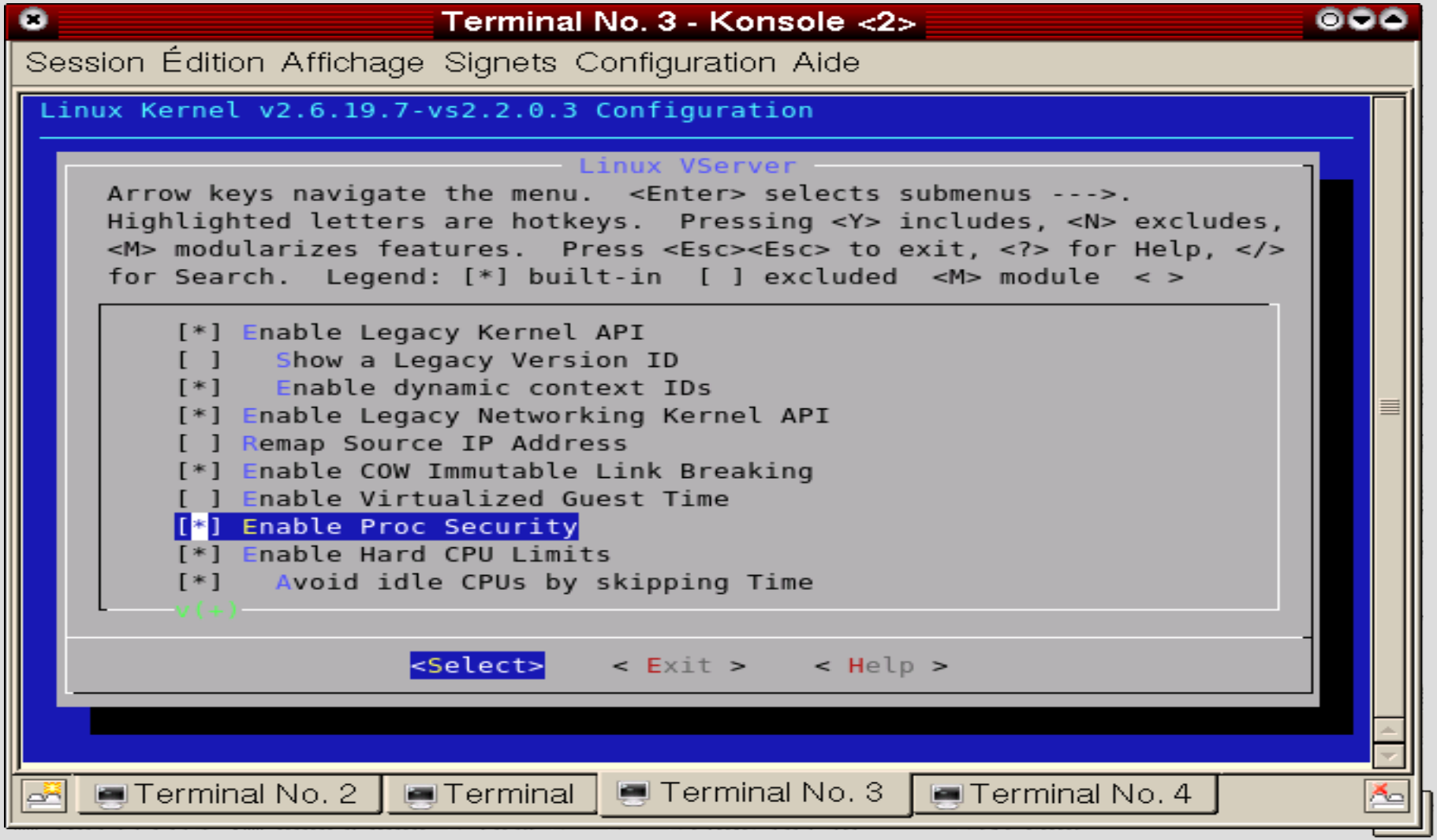
Installation : compilation noyau

- patcher les sources du noyau :
  - # mv patch-2.6.19-7-vs2.0.diff /usr/src/linux-2.6.19.7
  - # cd linux-2.6.19.7
  - # bzcat patch-2.6.19.7-vs2.2.0.diff.bz2 | patch -p1
- Configuration du noyau *[après le patch une section speciale apparaît pour vserver]*
  - # make menuconfig
- autres paquetages nécessaires avant la compilation :
  - # apt-get install bzip2 libncurses5-dev ***fakeroot kernel-package***

compilation du noyau patché



compilation du noyau patché



compilation du noyau patché

- compilation
  - make && make modules_install

- $ cp .config /boot/config-2.6.19.7-vs2.0
- $ cp System.map /boot/System.map-2.6.19.7-vs2.0
- $ cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.19.7-vs2.0
- $ mkinitrd -o /boot/initrd.img-2.6.19.7-vs2.0 2.6.19.7-vs2.0

compilation du noyau patché

- Il faut mettre à jour le menu grub. Editer le fichier /boot/grub/menu.lst et ajouter les lignes suivantes avant les autres entrées. Assurez vous que la ligne "default" est mise à 0

```
title       Debian GNU/Linux, kernel 2.6.19.7-vs2.2.0.3-vserver
root        (hd0,4)
kernel      /boot/vmlinuz-2.6.19.7-vs2.2.0.3-vserver root=/dev/hda5 ro
initrd      /boot/initrd.img-2.6.19.7-vs2.2.0.3-vserver
savedefault
```

note: Sous debian *update-grub* fait tout le boulot de modification du fichier */boot/grub/menu.lst* automatiquement à partir du moment où le noyau est présent dans /boot

compilation du noyau patché

- ***la même chose en plus facile à la mode Debian :***
- cd /usr/src/linux
  - `$ fakeroot make-kpkg --initrd --append-to-version -vserver --revision 1 kernel-image`
- cela produit un paquetage .deb dans /usr/src
  - `linux-image-2.6.19.7-vs2.2.0.3-vserver_1_i386.deb`
- que l'on peut installer comme un paquet Debian
  - `$ dpkg -i linux-image-2.6.19.7-vs2.2.0-vserver_1_i386.deb`
- Ce paquet va modifier tout seul le menu dans le fichier /boot/grub/menu.lst, placer le initrd et l'image du kernel (vmlinuz) dans le répertoire /boot, mais aussi créer le repertoire /lib/modules/2.6.19.7-vs2.2.0-vserver/ contenant tout les modules utiles.

compilation du noyau patché

- **dpkg -i linux-image-2.6.19.7-vs2.2.0.3-vserver_1_i386.deb**

```
Sélection du paquet linux-image-2.6.19.7-vs2.2.0.3-vserver
   précédemment désélectionné.
(Lecture de la base de données... 143566 fichiers et répertoires déjà
   installés.)
Dépaquetage de linux-image-2.6.19.7-vs2.2.0.3-vserver (à partir de
   linux-image-2.6.19.7-vs2.2.0.3-vserver_1_i386.deb) ...
Paramétrage de linux-image-2.6.19.7-vs2.2.0.3-vserver (1) ...
Running depmod.
Finding valid ramdisk creators.
Using mkinitramfs-kpkg to build the ramdisk.
Running postinst hook script /sbin/update-grub.
You shouldn't call /sbin/update-grub. Please call /usr/sbin/update-
   grub instead!
Searching for GRUB installation directory ... found: /boot/grub
Found kernel: /boot/vmlinuz-2.6.19.7-vs2.2.0.3-vserver
Found kernel: /boot/vmlinuz-2.6.19.7-vs2.0.3
Found kernel: /boot/vmlinuz-2.6.18-6-vserver-686
Found kernel: /boot/vmlinuz-2.6.18-5-686
Updating /boot/grub/menu.lst ... done
```

compilation util-vserver

- $ wget http://ftp.linux-vserver.org/pub/utils/util-vserver/util-vserver-0.30.215.tar.bz2
- $ bzcat util-vserver-0.30.215.tar.bz2 | tar xvf -
  - $ ./configure --prefix=/opt/util-vserver --with-crypto-api=none --with-vrootdir=/home/VSERVER –sysconfdir=/etc
  - make && make install

$ /opt/util-vserver/sbin/vserver-info

```
            prefix: /opt/util-vserver
     sysconf-Directory: /etc
         cfg-Directory: /etc/vservers
      initrd-Directory: $(sysconfdir)/init.d
    pkgstate-Directory: ${prefix}/var/run/vservers
       vserver-Rootdir: /home/VSERVER
```

-

création d'une machine virtuelle

- La création de nouveaux vservers peut se faire via la commande
  - ***$ vserver build***
- qui est capable de gérer un certain nombre de distributions Linux... dont Debian (utilise la commande *deboostrap* pour installer un système de base). Quelques corrections sont ensuite apportées au serveur fraîchement installé (essentiellement un /dev quasiment vide), afin de fonctionner au mieux dans un vserver.
- Une fois cette commande terminée, un vserver de base est installé. Il peut être sauvegardé pour ensuite servir de patron (sous la forme d'un fichier tar.gz, qu'il suffit de décompresser ensuite, au besoin).
- Cela permet des déploiements « minute ».

création d'une machine virtuelle

- La racine d'un vserver se trouve dans un sous-répertoire du serveur maître.. pointé par */etc/vserver.default/vdirbase*
- **vserver** vserver-momo **build**
  - ***--rootdir*** /etc/vserver/.default/vdirbase
  - ***--hostname*** vserver-momo
  - ***--interface*** eth0:192.168.0.5/255.255.255.0
  - ***-m*** *debootstrap* **--**
    - ***-d*** etch
    - **-m** http://ftp.fr.debian.org/debian/

- *[NB: Attention à la syntaxe des arguments ;-)]*
- *$ vserver build build --help*

création d'une machine virtuelle à partir d'un modèle

- Création d'un « patron » (template)
  - *cd /var/lib/vservers/pcsic25*
  - *tar czvf ../vserver-de-base.tgz ./**
- utilisation de ce « file system » patron pour créer une macine virtuelle nouvelle
- *vserver pcsic26 build -n pcsic26 --hostname pcsic26.com.univ-mrs.fr --interface eth0:139.124.2.26/24* ***-m template*** *--* ***-t ./vserver-de-base.tgz***

création d'une machine virtuelle (debian)

- ***newvserver***
- $ newvserver
- --vsroot /var/lib/vservers/
- --hostname NOM \
- --domain com.univ-mrs.fr
- --ip ip/masque \
- --dist etch
- --mirror http://ftp.fr.debian.org/debian/
- --interface eth0

- `--vsroot : chemin réel sur le système hôte ou le système de fichier dela VM sera installé`
- `--hostname : nom d'hôte du futur serveur virtuel`
- `--domain : nom de domaine dns, utile pour la résolution de nom sur le réseau`
- `--ip : adresse ip du futur serveur virtuel`
- `--dist : nom de la distribution debian installé sur le Vserver`
- `--mirror : mirroir de la distribution à installer`
- `--interface : interface réseau de la machine physique`

utilisation : Lancement d'un vserver

- Tous les processus démarrés par init sont dans le même contexte xid 0, et héritent de ses caractéristiques.
- Le contexte s'applique aux processus et aux adresses IP, qui sont ainsi isolés de tout ce qui n'appartient pas au contexte local.
- Les processus d'un vserver ne sont que des ***processus standards, juste isolés et bridés*** sur les appels systèmes qu'ils peuvent utiliser. Les disques ainsi que la mémoire sont également partagés, ce qui permet une forte montée en charge, mais peut aussi compromettre la stabilité de l'intégralité du système, si on n'y prend garde.
- Le vserver se base sur la gestion de “capabilities” du noyau Linux, qui permet de descendre les privilèges d'une tâche (et des tâches héritées), et de limiter les appels systèmes possibles. Et ainsi sécuriser l'ensemble.

Utilisation des VM: lancement d'un vserver

- Le lancement d'un vserver se fait via la commande
  - *$* ***vserver*** *<nom_du_vserver>* ***start*** .
- Cette commande va lire un ensemble de fichiers de configuration correspondants au serveur virtuel à démarrer.
- Elle crée un ***contexte*** (un numéro dit xid), qui est unique à la machine physique.
- Ensuite, le processus change de racine (chroot).
- Les adresses IPv4 du vserver sont créées en tant qu'adresses *secondaires* de l'interface réseau de la machine physique, puis liées au contexte actuel par des règles de routage internes au noyau Linux.
- Ensuite la commande vserver se remplace par un processus unique (le programme init , qui est le processus standard de démarrage des UNIX).

Administration des VM sur la machine hôte

- Paquetage indispensable « util-vserver »
- Lancer une machine virtuelle
  - `vserver <nom_VM> **start**`
- Arreter une machine virtuelle
  - `vserver <nom_VM> **stop**`
- Rentrer dans une machine virtuelle à partir de la machine hôte
  - `vserver <nom_VM> **enter**`
- changer de contexte (rentrer dans contexte d'un vserver)
  - **`chcontext --xid <numero_contexte> /bin/bash`**

Démarrage automatique des Vservers

- Lors du démarrage du serveur le script /etc/init.d/vservers-default vas être lancé pour initialiser et démarrer les vserveurs voulus.
- Les vservers démarrés par ce script sont les vserveurs possédant « default » dans le fichier
  - /etc/vservers/<nom_vserver>/apps/init/***mark***
- Pour lancer le Vserver pcsic21 au démarrage de la machine hôte :
  - `# echo "default" > /etc/vservers/pcsic21/apps/init/mark`

Utilisation des VM sur la machine hôte

- Installer des paquetages supplémentaires dans une machine virtuelle
  - rentrer dans la machine virtuelle
    - `vserver <nomVM> enter`
      - `apt-get install openssh-server`
  - Depuis la machine hôte sans rentrer dans la machine virtuelle
  - **`vapt-get`** `<nomVM>` **`--`** `install openssh-server`

Utilisation des VM : Détruire une machine virtuelle

- Détruire une machine virtuelle
  - **$ vserver <nom_VM> delete**
- $ vserver vserver-momo4 delete

```
Are you sure you want to delete the vserver vserver-momo4
  (y/N) y
```

```
# ll /home/VSERVER/
drwxr-xr-x 20 root root 4096 2008-03-24 17:53 vserver-momo
drwxr-xr-x 20 root root 4096 2008-04-13 16:39 vserver-momo3
# ll /etc/vservers/
-rw-r--r-- 1 root root   99 2006-12-09 11:46 newvserver-vars
drwxr-xr-x 6 root root 4096 2008-03-24 17:49 vserver-momo
drwxr-xr-x 6 root root 4096 2008-04-13 16:36 vserver-momo3
```

Administration des VM sur la machine hôte

- vserver-stat
  - montre les machines virtuelles lancées

```
$ vserver-stat
CTX   PROC    VSZ    RSS  userTIME   sysTIME    UPTIME NAME
0      104   1.8G 624.9M   8m19s54   2m22s90   2h02m31 root server
49153   11   136M  36.1M   0m12s44   0m01s58  20m50s62 vserver-momo
45000    2   3.8M   1.4M   0m00s28   0m00s26  17h23m40 vserver-momo2
49153   12 194.2M    50M   0m02s24   0m01s10  17h44m21 vserver-momo4
49156    8 127.2M  33.4M   0m02s11   0m01s70  17h25m23 vserver-momo3
```

- vtop
- vps ax , vpstree
  - montre tous les processus y compris ceux des machines virtuelles avec leur numéro de contexte

Administration des VM sur la machine hôte

- ***$ vserver-info*** : indique tous les chemins des VM

```
Versions:
   Kernel: 2.6.19.7-vs2.2.0.3-vserver
   util-vserver: 0.30.212; Dec  9 2006, 12:26:5
Paths:
   prefix: /usr
   sysconf-Directory: /etc
   cfg-Directory: /etc/vservers
   initrd-Directory: $(sysconfdir)/init.d
   pkgstate-Directory: /var/run/vservers
   vserver-Rootdir: /var/lib/vservers
```

copie – déplacement de machine virtuelle

- ***vserver-copy***
  - copier le système de fichier d'une machine virtuelle ainsi que toute sa configuration, d'une machine hôte à une autre machine hôte

    sur une autre machine hôte en réseau
  - `vserver-copy [options] vserver host:[newname]`
  - `$ vserver-copy -v  -s VMsic20 pcsic10:VMsic23`
  - NB: nécessite un fichier de configuration spécifique à la machine à copier /etc/vservers/VMsic23.conf

duplication de machine virtuelle (debian)

- dans le paquetage vserver-debiantools duplication d'une machine virtuelle sur un même hôte en changeant l'adresse IP
- $ ***dupvserver*** --from vserver-momo --to vserver-momo5 --ip 192.168.0.8

```
Changed the following files:

/etc/vservers/.defaults/vdirbase/vserver-
  momo5/etc/hostname

/etc/vservers/.defaults/vdirbase/vserver-momo5/etc/hosts

/etc/vservers/.defaults/vdirbase/vserver-
  momo5/etc/ssh/sshd_config
```

Installation d'un serveur SSH dans une VM

- Installer un serveur openssh
  - rentrer dans la machine virtuelle
    - `vserver <nomVM> enter`
    - `apt-get install openssh-server`

- Il faut que le daemon SSHD n'écoute QUE sur l'interface réseau de la machine virtuelle sur lequel il se trouve
  - Dans/etc/ssh/sshd_config
    - ListenAddress 192.168.0.1

Les fichiers de configuration des Vservers

- La méthode actuelle pour la configuration des vservers se fait :
- sous la forme d'un mini file system : ensemble de répertoires comprenant des fichiers de configuration..
  - /etc/vservers/<nom_du_vserver>/
- on trouvera une doc ici http://www.nongnu.org/util-vserver/doc/conf/configuration.html
- L'ancienne manière se fait sous la forme d'un fichier de configuration unique par vserveur
  - /etc/vservers/<nom_du_vserver.conf>

Le fichier de configuration général du service vserver

- Répertoire général de configuration ***/etc/vservers.conf***
- # Configuration file for the vservers service

BACKGROUND=yes
start the vservers on tty9, in background *so the rest of the*
boot process end early

BACKGROUND=no
VSERVERS_ROOT=/var/lib/vservers

les fichiers de configuration des VM

- Répertoire général de configuration ***/etc/vservers***.
  - Un répertoire de configuration par machine virtuelle http://www.nongnu.org/util-vserver/doc/conf/configuration.ht
  - ***ls -l /etc/vservers***
  - `drwxr-xr-x 6 root root 4096 2008-03-24 17:49` ***vserver-momo***
- Répertoire caché .default
  - ***$ ls -l /etc/vservers/.defaults/***

```
drwxr-xr-x 4 root root 4096 2008-04-15 18:43 apps
lrwxrwxrwx 1 root root   19 2008-04-15 18:43 cachebase ->
   /var/cache/vservers
drwxr-xr-x 2 root root 4096 2006-12-09 20:27 files
lrwxrwxrwx 1 root root   21 2008-04-15 18:43 run.rev ->
   /var/run/vservers.rev
lrwxrwxrwx 1 root root   13 2008-04-15 19:19 vdirbase ->
   /var/lib/vserver ... /là/ou/vous/voulez ... /home/VSERVER
```

les fichiers de configuration des VM

- /etc/vservers/pcsic21.conf : le fichier de configuration basique du vserver pcsic21.
- /etc/vservers/pcsic21/***context*** : le fichier contenant le xid (context id) du vserver pcsic21.
- /etc/vservers/pcsic21/***flags*** : permet de définir un certain nombre de restrictions à appliquer à pcsic21
- /etc/vservers/pcsic21/***bcapabilities*** : permet de donner un certain nombre de permission via des CAP_*
- /etc/vservers/pcsic21/***ccapabilities*** : permet d'activer la prise en charge de la limitation des ressources
- /etc/vserver/pcsic21/***rlimits***/ : contient les fichiers qui restreignent l'accès aux ressources
- /etc/vservers/pcsic21/***dlimits/0/*** : contient les fichiers qui restreignent l'accès aux disques
-

les fichiers de configuration des VM

- répertoires et fichiers de configuration pour une machine virtuelle
  - $ ls -l /etc/vservers/vserver-momo/

```
drwxr-xr-x 4 root root 4096 2008-03-24 17:49 apps
lrwxrwxrwx 1 root root   46 2008-03-24 17:49 cache ->
  /etc/vservers/.defaults/cachebase/vserver-momo
-rw-r--r-- 1 root root  112 2008-03-24 17:49 fstab
drwxr-xr-x 3 root root 4096 2008-03-24 17:49 interfaces
-rw-r--r-- 1 root root   13 2008-03-24 17:49 name
lrwxrwxrwx 1 root root   30 2008-03-24 17:49 run ->
  /var/run/vservers/vserver-mono
drwxr-xr-x 2 root root 4096 2008-03-24 17:49 uts
lrwxrwxrwx 1 root root   26 2008-03-24 17:49 vdir ->
  /home/VSERVER/vserver-momo
```

les fichiers de configuration des VM

- $ more /etc/vservers/vserver-momo/interfaces/0/ip
  - `192.168.0.5`
- $ more /etc/vservers/vserver-momo/name
  - `vserver-momo`
- $ more /etc/vservers/vserver-momo/uts/nodename
  - vserver-momo
-

les fichiers de configuration des VM : numéro de contexte

- Numéro de Contexte des VM :
  - on peut initialiser le compteur de contextes

echo 2000 > /etc/vservers/.defaults/context.next

  - on peut forcer le *numéro* de contexte d'une VM
    - dans le fichier `/etc/vservers/<nomVM>/context`
  - dans le cas où le noyau n'a pas été compilé avec l'option donnant des numéros de contexte dynamique
    - CONFIG_VSERVER_DYNAMIC_IDS=n

$ echo 1001 > /etc/vservers/vserver-momo2/***context***

les fichiers de configuration des VM
/etc/vservers/<VMconfig.conf> 1/3

- Pour ajuster le fichier de configuration de votre Vserver
  - **/etc/vservers/vserver-momo2.conf**

```
# Le nom du vserver.
     S_HOSTNAME="vserver-momo2"
# Sur quelle interface l'allias IP sera créé.
     IPROOTDEV="eth0"
# IP du vserver (IPROOT écrase l'option IPROOTDEV). Pour une
    passerelle :
# IPROOT="eth0:192.168.0.2/255.255.255.0
eth1:172.16.141.12/255.255.255.128"
#
# Pour avoir aussi une boucle locale. IPROOT="192.168.0.2
    127.0.0.1"
     IPROOT="192.168.0.2"
```

les fichiers de configuration des VM

/etc/vservers/<VMconfig.conf> 2/3

```
# Le masque et broadcast, si on ne la pas définit dans IPROOT.
IPROOTMASK=255.255.255.0
IPROOTBCAST="192.168.0.255"

# Démarrage ou non de ce vserver au lancement du daemon vserver.

ONBOOT="yes"

# Modifie le niveau de priorité pour tous les processus du vserver.
# de -20 (très haut) à 19 (très bas)
S_NICE=""
```

les fichiers de configuration des VM

/etc/vservers/<VMconfig.conf> 3/3

```
# Régle les états du contexte de sécurité.
#lock: Empêche le vserver d'être placé dans un nouveau contexte.
# nproc: Limite le nombre de processus dans le vserver selon l'ulimit cela
    devient une limite par- vserver)
S_FLAGS="lock nproc fakeinit hideinfo ulimit sched_hard
    sched_prio virt_mem virt_uptime virt_cpu virt_load hide_mount
    fork_rss "

# Limitation à 1024 processus pour l'ensemble du vserver.
ULIMIT="-H -n 1024"


# Capacités à donner au vserver (le moins possible).
# On donnera par exemple CAP_SYS_RESOURCE à un vserver hébergeant
    un serveur DNS Bind9.


S_CAPS="CAP_NET_RAW  CAP_SYS_ADMIN
    CAP_NET_BIND_SERVICE CAP_NET_BROADCAST"
```

Les contextes xid

- c'est grâce aux « contextes » (xid) que l'on met en oeuvre la notion ***d'isolation*** des vserver
  - jeter un coup d'oeil aux processus tournant sur un autre vserver est impossible s'il n'a pas le même xid.
- les numéros de contexte attribués aux vserveurs peuvent être soit ***dynamiques*** soit ***fixes***
- Il est plutôt déconseillé d'utiliser les xid dynamiques, puisque à chaque redémarrage du vserver, il nous sera difficile de savoir quel est son xid.
- Pour définir un numéro de contexte à un vserveur, il faut le placer dans le fichier de configuration:
  - ***/etc/vservers/<nom_du_vserver>/context***
- Ce fichier contient le xid du vserver (identificateur de contexte) . Ce xid est très important puisque c'est ce numéro qui est utilisé pour marquer les dossiers, processus d'un vserver

Les contextes xid

- Le fichier de configuration pour définir le numéro de contexte n'est pas obligatoire
  - s'il n'est pas défini, alors le système attribuera un xid dynamique au vserver
- Dans l'optique d'isoler un vserver de tout autre contexte. Il existe une règle pour demander à l'hôte d'attribuer un xid en statique.
  - Si la valeur stockée dans le fichier /etc/vservers/<nom-vserver>***/context*** est :
    - supérieure à 49151, le xid est attribué de manière dynamique
    - comprise entre 3 et 49150, le xid vaut la valeur indiquée (statique)
- Les xid de 0 et 1 sont reservés,
  - 0 pour la machine hôte (le « root suprême »),
  - le 1 pour le contexte Spectateur (qui voit tous les processus de l'hôte),

Les contextes xid

- Le contexte 0 est celui de la machine hôte. C'est le seul contexte habilité à créer ou accéder à un autre contexte
- Le contexte 1 permet de voir tous les contextes sans interagir avec les process (pas de kill)
  - ***vps ax | grep apach***
  - ***$ chcontext --xid 1*** ps wwaux | grep apac

```
root      3535  0.0  0.8  21076  8932 ?        Rs   19:11   0:00
   /usr/sbin/apache2 -k start
www-data  3550  0.0  0.4  21076  4760 ?        S    19:11   0:00
   /usr/sbin/apache2 -k start
www-data  3551  0.0  0.4  21076  4760 ?        S    19:11   0:00
   /usr/sbin/apache2 -k start
```

- Rentrer dans un nouveau contexte
  - **chcontext --xid 49152 /bin/bash**

contexte et isolation de /proc

/proc

Système de fichiers virtuel permettant l' Accès aux informations du noyau

- Nécessaire dans un vserver : /proc/uptime, /proc/cmdline liste des processus, type de cpu, mémoire utilisée, points de montage, ...

- Les processus des autres contextes n'apparaissent pas

Certaines entrées sont « cachées » à l'aide d'attributs supplémentaires

contexte et isolation de /proc

- problème possible au lancement d'un vserver

```
$ vserver-stat
WARNING: can not access /proc/uptime. Usually, this is
   caused by procfs-security. Please read the FAQ for
   more details http://linux-vserver.org/Proc-Security
    open("/proc/uptime"): No such file or directory
```

- `$ showattr /proc/uptime`
  - `AwH-ui- /proc/uptime`
- pour résoudre le probleme
  - lancer le script ***/usr/lib/util-vserver/vprocunhide*** qui rétablit les bonnes permissions
  - ou $ setattr --~hide /proc/uptime
- `$showattr /proc/uptime`
  - `Awh-ui- /proc/uptime`

Isolation des file system des machines virtuelles

- Les file system des vservers sont isolés par un simple chroot... qui n'est pas infranchissable
  - nouvel attribut de fichier ***barrier*** (cet attribut empêche le franchissement de la directory marquée par le flag B)

- $ showattr -d /home/VSERVER/ /var/lib/vservers

```
---Bui- /home/VSERVER/
---bui- /var/lib/vservers
```

- Pour mettre l'attribut « barrier
  - `$ setattr --barrier /var/lib/vservers/`
- Pour enlever l'attribut « barrier » (~ pour enlever)
  - `$ setattr --~barrier /var/lib/vservers/`

Les capabilities

- Les capabilities systemes sont les droits ou permissions que le systeme donne aux processus
  - http://www.lids.org/lids-howto/node34.html
  - http://linux-vserver.org/Capabilities_and_Flags
  - Par exemple si un processus essaye de paramétrer l'horloge, le kernel Linux testera si le processus possede le bit CAP_SYS_TIME
- Pour voir les capabilities par défaut

```
$ reducecap --show
```

Capability	Effective	Permitted	Inheritable
CAP_CHOWN	X	X	
CAP_KILL	X	X	
CAP_SETGID	X	X	
CAP_SETUID	X	X	
CAP_QUOTACTL	X	X	

Les capabilities : liste

CAP_CHOWN Change le propriétaire et le groupe du fichier

CAP_KILL Envoie un signal au processus avec un ID utilisateur différent, réel ou effectif

CAP_SETGID Permet le passage de setgid, setgroups,et de gids forgés sur les sockets

CAP_SETUID Permet le passage de set*uid et d'uids forgés sur les sockets

CAP_SETPCAP Transfert/déplace toute capacité de l'ensemble permis vers/ depuis n'importe quel pid

CAP_LINUX_IMMUTABLE Permet la modification des attributs de fichier S_IMMUTABLE et S_APPEND

CAP_NET_BROADCAST Permet la diffusion et l'écoute en multicast

CAP_NET_ADMIN Permet la configuration de l'interface,le pare-feux IP, le masquerading, l'accounting, le débugage de sockets, les tables de routage, la liaison sur n'importe quelle adresse,

Les capabilities : liste

CAP_NET_RAW	Permet l'utilisation des sockets RAW et PACKET
CAP_SYS_MODULE	Ajoute et retire des modules noyau
CAP_SYS_CHROOT	Permet chroot
CAP_SYS_PTRACE	Permet ptrace() sur n'importe quel processus
CAP_SYS_ADMIN	quasiment toutes les capabilities
CAP_SYS_BOOT	Permet le reboot
CAP_SYS_NICE	Permet l'élévation des privilèges et la modification de la priorité des autres processus, en modifiant l'ordonnanceur
CAP_SYS_RESOURCE	Passe outre les limites de ressource, de quota, l'espace réservé sur un système de fichier...
CAP_MKNOD	Permet les aspects privilégiés de mknod

Les ccap : capabilities par contexte

- Les « system capabilities » standard ne permettent pas de gérer finement les droits des process.

- Les Ccap permettent de rajouter ou enlever des capacités par contexte (context capabilities)
  - http://linux-vserver.org/Capabilities_and_Flags

- Les « context capabilities » sont spécifiques à Linux Vserver
-
- Elles sont appliquées à tous les processus à l'intérieur d'un contexte
  - exemple : RAW_ICMP est une permission plus « fine » que CAP_NET_ADMIN qui permet le « ping » mais pas le traceroute

Les ccap : capabilities par contexte

- Comment enlever/rajouter une context capability ?
- par fichier de configuration /etc/vservers/viu/ccapabilities
  - permet d'activer un certain nombre de paramètres, comme rlimit. Il existe théoriquement 8 ccaps qui sont :
  - *utsname, raw_icmp, syslog, secure_mount, secure_remount, binary_mount, quota_ctl, rlimit*
- par la commande *vattribute*
  - **$ vattribute --set --xid 49152 --ccap ~RAW_ICMP**

```
$ vserver vserver-momo3 enter

ping 192.168.0.1
ping: icmp open socket: Operation not permitted
```

Les ccap : capabilities par contexte

0	0x00000001	SET_UTSNAME	Allow setdomainname(2) and sethostname(2)
1	0x00000002	SET_RLIMIT	Allow setrlimit(2)
8	0x00000100	RAW_ICMP	Allow usage of raw ICMP sockets
12	0x00001000	SYSLOG	Allow syslog(2)
16	0x00010000	SECURE_MOUNT	Allow secure mount(2)
17	0x00020000	SECURE_REMOUNT	Allow secure remount
18	0x00040000	BINARY_MOUNT	Allow binary/network mounts
20	0x00100000	QUOTA_CTL	Allow quota ioctls
21	0x00200000	ADMIN_MAPPER	Allow access to device mapper
22	0x00400000	ADMIN_CLOOP	Allow access to loop devices

Limiter les ressources à des processus

- On limite les ressources dans les répertoires et fichiers de configuration des vserveurs
  - ***/etc/vservers/<nomduvserver>/rlimits/***
- Ce répertoire peut contenir plusieurs fichiers *correspondant à des ressources*, comme *as, rss, memlock, nproc, data*... où on définit la limite de la ressource en question pour le vserver.
- Exemple pour limiter la taille des pages résidentes à 100000
  - `echo 100000 > /etc/vservers/viu/rss`
- (la colonne de gauche définit pour chacune des limites l'option à mettre dans la commande de création de vserver dans le cas où l'on veut tout définir d'un coup. Malheureusement, cette commande ne met l'architecture rlimits correctement en place.)

Limiter les ressources à des processus

Limit	*ProcFS*	*config*	*Unit*	*Description*
-t	CPU	cpu	s	*Quantité de temps cpu en seconde*
-f	FSIZE	fsize	kb	*Taille des fichiers crées par le shell*
-d	DATA	data	kb	*Taille d'un segment de donnée d'un processus*
-s	STACK	stack	kb	*Taille de la pile*
-c	CORE	core	kb	*Taille des fichiers crées par le noyau*
-m	RSS	rss	page	*resident set size*
-u	NPROC	nproc	int	*nombre de processus*
-n	NOFILE	nofile	int	*nombre de file handles*
-l	MEMLOCK	memlock	page	*Pages lockées en mémoire*
-v	AS/VM	as	page	*Page de mémoire virtuelle*
-?	LOCKS	locks	int	*Verrous du système de fichier*
-p	MSGQUEUE	MSGQ	512b	*message queue size*
	VLimit			
--	NSOCK (16)	SOCK	int	nombre de sockets
--	OPENFD (17)	OFD	int	nombre de file descriptors

Limiter les ressources disques à des processus

- On limite les ressources disques des processus dans les répertoires et fichiers de configuration des vserveurs
  - **/etc/vservers/viu/dlimits/0/**
- Le répertoire est composé de 7 éléments :
  - le ***répertoire*** /etc/vservers/viu/dlimits/***0/***
  - le fichier /etc/vservers/viu/dlimits/0/directory
  - le fichier /etc/vservers/viu/dlimits/0/space_total
  - le fichier /etc/vservers/viu/dlimits/0/space_used
  - le fichier /etc/vservers/viu/dlimits/0/inodes_total
  - le fichier /etc/vservers/viu/dlimits/0/inodes_used
  - le fichier /etc/vservers/viu/dlimits/0/reserved

Limiter les ressources disques à des processus

- exemple:
  - cat /etc/vservers/viu/dlimit/0/directory
    - /etc/vservers/viu
  - echo 5000000 > /etc/vservers/viu/dlimit/0/space_total
  - echo 100000 > /etc/vservers/viu/dlimit/0/inodes_total

utilitaires avancés : les vserver-tools

Pourquoi VS-Tools ?

- http://www.jres.org/tuto/tuto6/index
- http://www.jres.org/_media/tuto/tuto6/tutojres_.pdf?id=tuto%3Atuto6

- Util-Vserver : le paquetage officiel indispensable de Linux-Vserver
  - Fournit l'interface entre l'utilisateur, les machines virtuelles, et le noyau de l'hôte
  - Manque de fonctionnalités, mais sa conception permet d'en ajouter sans être modifié
- Vserver-Debiantools: «la touche Debian» Intéressant, mais encore insuffisant

utilitaires avancés : les vserver-tools

- VS-Tools: «le complément»

  Création des Vserveurs simplifiée à l'extrême

  - arguments indispensables: un nom, et une adresse IPv4
  - Surveillance et supervision
  - Déplacements et consolidation des Vserveurs
  - Prise en charge automatique du routage et des VLANs
  - Gestion du firewall
  - Gestion (à chaud) des ressources

vs-tools : installation

- Installation de «vs-tools» en premier sur la machine hôte
  - Créer une paire de clés SSH
  - Télécharger et installer les scripts
  - wget https://listes.univ-reims.fr/sympa/d_read/vs-tools/Source
  - cd /usr/src ; tar -xzf Latest.tgz ;
  - cd vs-tools ; ***sh install.sh***

- Configurer les différents fichiers de config de «vs-tools»
- Enfin de déployer l'ensemble sur tous les hôtes du pool
  - cd /usr/src/vs-tools ; sh update-hosts.sh

vs-tools : installation

- Paramétrer les différents fichiers de configuration dans /etc/vs-tools/
- backup.conf
- monitor.conf
- slaves.conf
- create.conf
- networks.conf
- util-vserver.conf
- firewall.conf
- vs-tools.conf

vs-tools : exploitation

- Généralités
- Toutes les commandes sont préfixées: «vs-»
- Et documentées: «--help»
- Certaines sont dédiées au «maître»
- vs-scan, vs-backup, vs-move, vs-get, vs-put, vs-remove
- L'activité de la librairie est journalisée via Syslog
- Elle peut être invoquée «manuellement»
- vs-functions <nom fonction> <arguments>

références

- site officiel
  http://linux-vserver.org/Welcome_to_Linux-VServer.org

- Vserver sur Debian
- http://linux-vserver.org/Installation_on_Debian
- http://linux-vserver.org/Installation_on_Linux_2.6
- http://fr.wikibooks.org/wiki/Vserver
- Compilation du noyau
- http://www.howtoforge.com/linux_vserver_debian

références

- Les capabilities (droits des processus) :
  - http://www.lids.org/lids-howto/node34.html
  - http://linux-vserver.org/Capabilities_and_Flags
-
- seminaires en france sur la virtualisation
  - http://www.jres.org/tuto/tuto6/index
  - http://www.resinfo.org/spip.php?article3
-
-
-