

Virtualisation avec *openVZ*

Journées thématiques SIARS
DR12 CNRS

septembre 2009

M. Libes – T. Dostes

Plan

- Vserver / openVZ ?
- Fonctionnalités de openVZ
- Installation de openVZ
 - Paquetage des distributions, Patch noyau, proxmox
- Récupération d'OS template, ou création
- Opérations de base
 - Création, destruction, lancement, arrêt, migration
- Configuration de openVZ
- Les interfaces réseau
- Gérer les ressources des VM

Rappel rapide techniques de virtualisation

Les différentes Techniques de virtualisations:

➤ **Machine Virtuelle** (VmWare GSX, Virtual PC) :

Solution très comparable à **un émulateur**, et parfois même confondue ;

➤ **Para-virtualisation** (Xen, Vmware ESX) :

Un hyperviseur **est un noyau hôte allégé** et optimisé pour ne faire tourner que des noyaux d'OS invités modifiés pour tourner sur cette architecture spécifique. Les applications en espace utilisateur des OS invités tournent ainsi sur une pile de deux noyaux optimisés, les OS invités ayant conscience d'être virtualisés.

➤ **Isolateur** (Linux-Vserver, chroot, OpenVZ) :

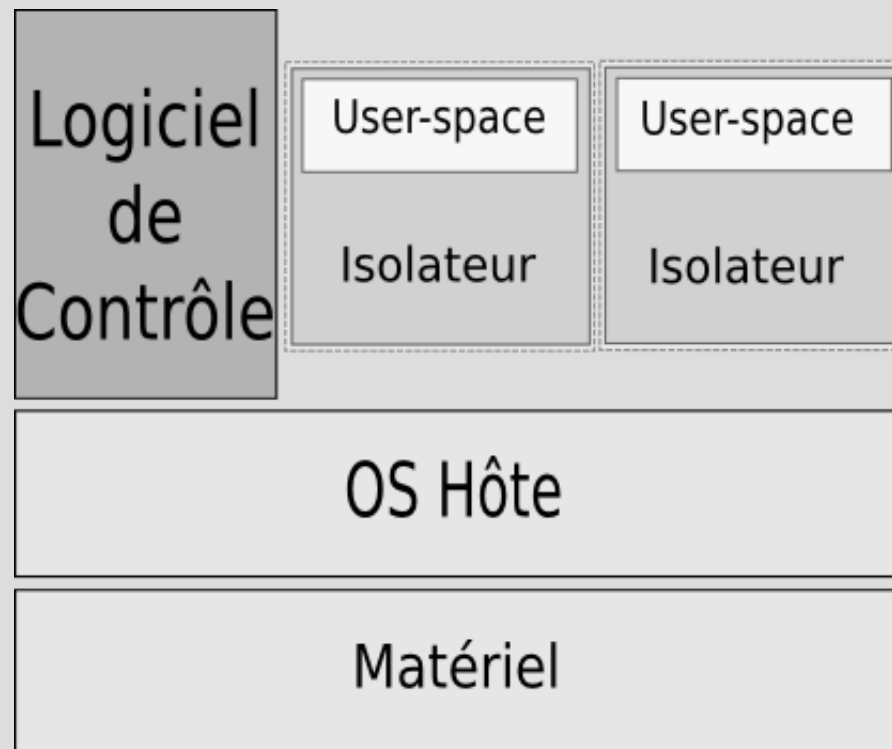
Un isolateur est un logiciel permettant d'isoler l'exécution des applications dans **des contextes ou zones d'exécution**.

openVZ : Principe de base

- OpenVZ (comme Vserver) = Virtualisation au niveau Kernel : Il s'agit d'un partitionnement logique au niveau des ressources systèmes : 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 comme dans d'autres systèmes 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.



openVZ : Principe de base

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

openVZ : Principe de base

- openVZ 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 (*vzctl*) pour administrer les machines virtuelles.
- Concrètement, un container openVZ 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 CT0, celui dédié à la machine hôte.
- => les machines virtuelles n'ont PAS de noyau

openVZ 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

openVZ avantages

- C'est la performance qui rend les VPS openVZ attractifs.
- Cela peut simplement être vu comme un chroot du file system amélioré par une isolation des processus
- Avantages = **performances natives** (pas de perte mesurable). A part la gestion du contexte, un processus dans un VPS 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 VPS et la mémoire demandée à l'hôte est celle réellement utilisée par les processus du VPS).
- La possibilité de mutualisation est donc ici très importante ; il est possible de déployer plusieurs dizaines de VPS sur un serveur physique correctement taillé.

openVZ avantages

- désormais supporté au niveau des distributions Debian par des paquets appropriés
- Virtualisation légère :
 - Performance native, aucun ralentissement mesurable.
 - Très économe en ressource : plusieurs VPS déployés sur une machine
 - Les règles IPTables (firewall interne) peuvent être manipulées dans un serveur virtuel : sécurité renforcée
- Différentes distributions peuvent fonctionner en parallèle sur un même noeud hôte HN
 - (Debian, CentoS, Fedora, Ubuntu...)

Intérêts openVZ

- Les VPS « Virtual Private Servers » se comportent comme des serveurs à part entière, mais « isolés » du système hôte
 - chaque VPS a ses propres processus, utilisateurs, système de fichiers, shell
 - chaque VPS a sa propre adresse IP, ses propres « ports », un filtrage IP et ses règles de routage
 - chaque VPS possède sa propre configuration et librairies système... plusieurs VPS peuvent cohabiter et chacune est isolée des autres ainsi que du système hôte
- On peut ainsi faire tourner des dizaines de serveurs virtuels sur une même machine → Administration facilitée, économie d'énergie
 - Fournir des serveurs à des étudiants
 - Tester des applications en toute sécurité
 - Consolider des serveurs

Vserver / openVZ

Vserver

- Documentation disparate
- Galère pour récupérer des des OS template de VE
- Configuration fastidieuse (capabilities,)
- Pas de boucle locale lo0
- Pas de pare-feu possible sur le nœud

OpenVZ

- + fonctionnalités
 - Iptables
 - NFS serveur
- Meilleure virtualisation des interfaces réseau
- Meilleure gestion des ressources
- Documentation rédigée et complète
- Gestion des OS template pour VE

Vserver / openVZ

Technique	OS	Licence	Fonctionnalité Système					Fonctionnalité Réseau			
			Isolation FS	Quotas Disque	Memoire Limit	Quotas CPU	Migration à chaud	Isolation Réseau	Firewall (netfilter)	loopback	IPv6
Linux-Vserver	Linux	GPL V2							*		
OpenVZ	Linux	GPL V2									

* : le filtrage réseau est effectué uniquement sur la machine hôte.

Quelques termes openVZ

- **HN** : « hardware node », la machine hôte
- **Container** : une machine virtuelle isolée
- CT : container = 1 VPS
 - CT0 : le container principal...le HN
 - CTID : identifiant de container
- **VPS** : « virtual private system »
- **VE** : Virtual Environment
 - VEID : identifiant de VE
- VM : « virtual machine » plutôt utilisé pour de la virtualisation via un émulateur

Fonctionnalités openVZ

Fonctionnalités : Virtualisation système

- Chaque VPS est un système indépendant
- Une part infime de CPU (1-2%) est consommée par la virtualisation (même noyau)
- Principales fonctionnalités :
 - Les VPS sont des systèmes Linux normaux (file systeme, scripts, programmes) : aucune spécificité openVZ
 - Les VPS sont totalement isolées les unes des autres (file system, processes, Inter Process Communication (IPC), `sysctl` variables)
 - Les processus appartenant à un VPS sont schedulés par rapport à tout le CPU disponible ; Les VPSs n'ont pas leur propre CPU , ils partagent le CPU de la machine hôte et peuvent donc utiliser toute la CPU disponible

Fonctionnalités : Virtualisation réseau

La virtualisation réseau de openVZ isole les VPS entre elles

- Chaque VPS a sa propre adresse réseau, les adresses multiples par VPS sont permises
- Le trafic réseau de chaque VPS est isolé. Les VPS sont protégées les une des autres (pas de snooping possible)
- On peut protéger les VPS par des iptables à l'intérieur de chaque VPS
- On peut manipuler les règles de routage allouer des MTU différents par VPS etc...

Fonctionnalités : gestion des ressources

La gestion des ressources physique est un élément capital de la virtualisation puisque plusieurs VPS peuvent entrer en concurrence pour la RAM, le CPU, l'espace disque

- OpenVZ permet de contrôler les ressources disponibles et allouées à chaque VPS.

La gestion des ressources permet:

- de partager les ressources physique du noeud hôte entre les VPSs;
- fournir une garantie de performance et protéger les VPS contre les DoS
- Toutes les ressources peuvent être modifiées à chaud sans rebooter le noeud hôte.
 - Si on veut augmenter le taux de RAM pour un VPS, on modifie le parametre dans le fichier de conf. , ce qui est impossible à faire avec les technique de virtulisation par émulateur ou hyperviseur

Fonctionnalités : gestion des ressources

- **Quotas disques à 2 niveaux:** au niveau de la machine hôte et au niveau des VPS
 - 1er niveau de quotas: l'administrateur de la machine hôte peut placer des quotas disques pour chaque VPS : en espace disque (blocs) ou nombre de fichiers (inodes)
 - 2ème niveau : l'administrateur d'une VPS peut utiliser les quotas Unix traditionnel pour limiter l'espace par utilisateur ou groupe
 - Paramètres modifiables à chaud: si on veut donner plus d'espace disque à un CT, on modifie dans le fichier de configuration.. pas besoin de rebooter la machine hôte ou de retailler des partitions

Fonctionnalités : gestion des ressources

- Objectif : Aucune VPS ne doit pouvoir saturer le système hôte à elle seule
- Partage de CPU équitable
 - L'administrateur OpenVZ peut paramétrer des valeurs différentes de `cpuunits` pour différents CT.
 - On peut aussi définir des limites de CPU time et limiter un CT pour qu'il ne dépasse pas 10% de CPU disponible
- Partage E/S équitable :
 - au niveau des CT on peut assigner des priorités d'I/O. Le scheduler attribue une bande passante en I/O à chaque VPS selon les priorités configurées

Fonctionnalités : gestion des ressources

- Migration à chaud
 - *http://wiki.openvz.org/Checkpointing_and_live_migration*
- Des capacités de checkpointing permettent de sauvegarder l'état à chaud d'un VPS.
- Un snapshot d'un CT est créé et son état est sauvé sur disque Ce fichier peut alors être transféré sur une autre machine et restauré en état de marche sans délai (quelques secondes).
- L'état complet du container est sauvé y compris les connexions réseau ouvertes.
- Cela permet de migrer/transférer un VPS d'un serveur physique à l'autre sans arrêt
 - D'un point de vue de l'utilisateur la migration est totalement transparente

Fonctionnalités : les « BeanCounters »

- Les « beancounters » sont un ensemble de paramètres attribué à chaque CT imposant des limites et des garanties pour que chaque VPS n'utilise pas seule toute les ressources de la machine hôte.
- Une vingtaine de paramètres pour limiter les opérations de chaque CT
 - Modifiables dans le fichiers de configuration
 - */etc/vz/conf/<ctid>.conf*
 - Visibles à chaud dans */proc/user_beancounters*
 - 5 valeurs associées à chaque « beancounter », limite hard et soft, l'administrateur est averti en cas de dépassement
- http://wiki.openvz.org/User_beancounter

Fonctionnalités : Les OS templates

- Un OS template est un « file system » d'une distribution Linux qui sert à « peupler » les VPS
- Un OS template comporte les librairies système, les scripts pour booter la machine, applications et utilitaires basiques
 - désormais disponibles sous forme de paquetages
 - Ou bien que l'on peut se créer soi même
- Différents OS templates sont disponibles et donc différentes distributions Linux (VPS) peuvent coexister sur le même nœud physique (HN)

Installation de openVZ

Installation des paquetages openVZ

Pour fonctionner OpenVZ a besoin :

- d'un noyau "patché"...
- d'utilitaires de contrôles des machines virtuelles (lancement, arrêt..) et,
- d'un « OS Template » : le système de fichiers du VPS

Installation des paquetages openVZ

- Paquetages nécessaires sous Debian
 - Un noyau patché
 - Au minimum l'utilitaire vzctl,
 - Plus .. vzdump, vzquota

```
com10: ~# uname -a
Linux com10 2.6.18-openvz-18-53.5d3-686 #1 SMP Sun Jan 11 01:09:09
CET 2009 i686 GNU/Linux
```

```
com10: ~# dpkg -l | grep vz
```

```
ii  linux-image-2.6.18-openvz-18-53.5d3-686 ovz004.1d3      Linux
kernel binary image for version 2.6.18
ii  vzctl          3.0.22-14          server virtualization solution -
control tools
ii  vzdump        1.11               OpenVZ backup scripts
ii  vzquota       3.0.11-1           server virtualization solution -
```

Installation du kernel openvz

- Rappel : il suffit de démarrer sur un kernel patché « openVZ » pour avoir les capacités d'isolation de VPS → installer un noyau patché
 - <http://wiki.openvz.org/Category:Installation>
- i) Par le mécanisme de gestion de paquetage des distributions [apt-get Debian, yum Centos)
- ii) Via proxmox (bare metal installation)
- iii) Via le CD d'install Debian ou Centos
- iv) En patchant et compilant un nouveau noyau

Installation du kernel openvz sur Debian

- Les paquets openvz sont directement disponibles à partir de Debian lenny

- On peut les voir en activant la completion bash

- `. /etc/bash_completion`

- `# apt-get install linux-image`

```
# apt-get install linux-image-2.6.26-1-openvz-686
```

- L'installation modifie le fichier de boot grub menu.lst
- rajouter le dépôt openVZ dans le fichier `/etc/apt/source.list` pour obtenir les paquets spécifiques du site openVZ
 - `deb http://download.openvz/debian-systems etch openvz`

Installation du kernel openvz sur CentOS

- Récupérer un fichier des dépôts openvz et les faire prendre en compte par yum

```
# cd /etc/yum/repos.d
# wget http://download.openvz.org/openvz.repo
# rpm --import http://download.openvz.org/RPM-GPG-
Key-OpenVZ
```

- Mettre à jour le cache yum avec le nouveau dépôt, et installer le kernel openvz, et rebooter

```
# yum update
# yum install ovzkernel vzctl
```

- <http://wiki.openvz.org/Yum>
- http://wiki.openvz.org/Quick_installation

Installation des utilitaires openvz

- *(On continue sous debian lenny)*
- Pour contrôler toutes les opérations sur les VPS
 - apt - get install **vzctl**
- Pour sauvegarder et restaurer les VPS
 - apt - get install **vzdump**
- Pour gérer les quotas dans les VPS
 - apt - get install **vzquota**
- Pour surveiller les processus sur toutes les VPS
 - apt - get install **vzprocps**

Redémarrer sur le nouveau noyau virtualisé

- Après avoir récupéré et installé le kernel patché openVZ, il suffit de rebooter la machine hôte...
 - vous devriez voir le nouveau noyau à démarrer dans le menu de boot
 - Sinon vérifier */boot/grub/menu.lst*

```
com10: ~# uname -a
```

```
Linux com10 2.6.18-openvz-18-53.5d3-686 #1  
SMP Sun Jan 11 01:09:09 CET 2009 i686  
GNU/Linux
```

LES OS TEMPLATES

Les OS template

- Les «OS template » sont les systèmes de fichiers « modèles » (sous forme de .tar.gz) qui vont servir à créer les machines virtuelles.
- OpenVZ permet de faire coexister différentes distributions sur le même serveur hôte.
- Un certain nombre de OS template sont préétablis et disponibles
 - <http://wiki.openvz.org/Download/template/precreated>
- On peut également créer ses propres templates personnalisés pour différents OS (CentoS, Debian..)
 - <http://wiki.openvz.org/Category:Templates>

Obtenir et installer les OS template

- les OS template Debian sont proposés directement sous forme de paquetages Debian
 - *apt-get install vzctl-ostmpl-debian-5.0-i386-minimal*
 - *apt-get install vzctl-ostmpl-debian-4.0-i386-minimal*
- D'autres OS template préétablis sont proposés sur
 - <http://wiki.openvz.org/Download/template/precreated>
 - `wget`
`http://download.openvz.org/template/precreated/centos-5-x86.tar.gz`
- Il suffit de déposer ces OS template dans le répertoire prévu à cet effet (sur la machine hôte)
 - */var/lib/vz/template/cache/*

Création de OS template

- Si besoin on peut se créer ses propre OS template
 - Qui ne sont que un tar.gz bien configuré d'un file system
- <http://wiki.openvz.org/Category:Templates>
- http://wiki.openvz.org/Debian_template_creation

Opérations de base sur les VPS openvz

- Création,
- destruction,
- lancement,
- arrêt,
- migration

Créer une machine virtuelle (VM)

- Après avoir téléchargé les OS template...
- ✓ **vzctl create 2143**
 - ✓ **--ostemplate debian-5.0-i386-minimal**
 - ✓ **--hostname ldap2**
 - ✓ **--name ldap2**
 - ✓ **--ipadd 139.124.2.143**
 - ✓ **--nameserver 192.168.1.2**
 - ✓ **--save**
- Où 2143 est l'ID (identifiant du container)

Démarrer/ entrer / arrêter les VPS

- `vzctl start <ctid>`
 - `$ vzctl start 2143`
- `vzctl stop <ctid>`
 - `$ vzctl stop 2143`
- `vzctl enter <ctid>`
 - `$ vzctl enter 2143`
- Pour démarrer le VPS au boot du HN (machine hôte)
 - `vzctl set 101 -- onboot yes -- save`

Paramètres du noyau pour openVZ

- Un certain nombre de paramètres du kernel doivent être positionnés pour faire fonctionner openVZ
- Les paramètres du kernel sont dans ***/etc/sysctl.conf***
- Modifiable par la commande « ***sysctl*** »

– A minima activer le ip_forward du kernel du HN

```
$ sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.proxy_arp = 0
net.ipv4.conf.all.rp_filter = 1
kernel.sysrq = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
kernel.ve_allow_kthreads = 1
```

Paramètres du noyau pour openVZ

- Rappel : **sysctl** est utilisé pour modifier les paramètres du noyau en cours d'exécution. Les paramètres utilisables sont ceux listés dans le répertoire */proc/sys*.

- `$ sysctl -w net.ipv4.ip_forward = 1`

- ou bien

- `echo 1 >`

- `/proc/sys/net/ipv4/ip_forward`

- `cat /proc/sys/net/ipv4/ip_forward`

Détruire un VPS

- D'abord arrêter le VPS
 - \$ vzctl **stop** <ctid>
- Puis le détruire
 - \$ vzctl **destroy** <ctid>
 - *Revient à détruire le fichier de config et le file system*

Lister les VPS

- Lister les VPS en cours d'exécution sur un HN
 - *vzlist*

```
com10: ~# vzlist
```

CTID	NPROC	STATUS	IP_ADDR	HOSTNAME
2131	11	running	139.124.2.131	cup
2140	24	running	139.124.2.140	mapserver
2142	41	running	139.124.2.142	www2
2143	23	running	139.124.2.143	ldap2
2144	32	running	139.124.2.144	pcsic144

Interagir avec les VPS

- Lister les processus d'un VPS, depuis la machine hôte
 - **vzctl exec** <CTID> ps ax
 - **vztop**
- Entrer, se logger dans un VPS
 - **vzctl enter** <CTID>
 - \$ vzctl enter 2143

Sauvegarde et restauration des VPS

- Sauvegarde :
 - Par défaut dans `/var/lib/vz/dump`
 - Sinon dans le répertoire de `-dumpdir`
 - *(pourquoi pas sur un NAS monté en NFS !!)*
 - `# vzdump --compress --dumpdir /mnt/vzdump/ --stop 2143 --mailto root@mondomaine.fr`
- Restauration d'un VPS sauvegardé
 - `#vzdump --restore /mnt/vzdump/vzdump-2143.tgz 2143`

Sauvegarde et restauration des VPS : les logs

VMID	NAME	STATUS	TIME	SIZE	FILENAME
2144	pcsic144	OK	00:06:39	633MB	/mnt/vzdump/vzdump-2144.tgz
TOTAL			00:06:39	633MB	

Detailed backup logs:

```
vzdump --compress --dumpdir /mnt/vzdump/ --stop 2144 --mailto  
administrateur@com.univmed.fr
```

```
2144: Sep 13 02:00:02 INFO: Starting Backup of VM 2144 (openvz)  
2144: Sep 13 02:00:02 INFO: status = CTID 2144 exist mounted running  
2144: Sep 13 02:00:02 INFO: starting first sync /var/lib/vz/private/2144 to  
/var/tmp/vzdumtmp21495  
2144: Sep 13 02:02:19 INFO: Number of files: 34321  
2144: Sep 13 02:02:19 INFO: Number of files transferred: 29768  
2144: Sep 13 02:02:19 INFO: Total file size: 1030888289 bytes  
2144: Sep 13 02:02:19 INFO: Total transferred file size: 1027698936 bytes
```

Migration des VPS d'un HN à l'autre

- Une des Fonctionnalité la plus puissante pour assurer une redondance de service ou une « presque » haute disponibilité
 - Migrer un VPS à chaud d'un serveur hôte à un autre sans interruption de disponibilité
- La migration fonctionne avec rsync et ssh
- Prévoir une génération de clés ssh sur les 2 noeuds hôtes
 - `$ ssh-keygen -t dsa -b 2048`
- Mettre la clé publique de HN1 dans HN2 et réciproquement pour identifier les 2 serveurs
 - `$ scp .ssh/id_dsa.pub HN2: /root/.ssh/authorized_keys`
 - Et inversement

Migration des VPS d'un HN à l'autre

- Migrer le VPS 2143 vers le HN 192.168.1.10
 - `$ vz migrate -r no --online -v 192.168.1.10 2143`
- Argument **-r yes|no** permet de détruire ou non le VPS « source » après la migration
 - Le laisser sur place accélérera le transfert dans l'autre sens si besoin
- **--online** : migration à chaud sans interruption

Quelques autres commandes intéressantes

- Génère un fichier de config en **partageant** les ressources systèmes physiques réelles en autant de VPS demandées

– \$ **vzsplit**

```
com10: ~# vzsplit  
Enter the number of  
containers: 4
```

- Connaître dans quel VPS tourne le process pid

– \$ **vzpid**

```
com10: ~# vzpid 27628  
Pid    VEID    Name  
27628  2140    apache
```

Quelques autres commandes d'exploitation interessantes

- Comparatif de l'état de la mémoire utilisée par les différents VPS
 - *vzmemcheck*

```
com10:~# vzmemcheck -v
```

```
Output values in %
```

```
veid      LowMem LowMem   RAM MemSwap MemSwap  Alloc  Alloc  
Alloc
```

```
          util commit  util  util commit  util commit limit  
2144      0.65 21.20  4.36  2.25 17.17  8.06 17.17 18.70  
2131      0.25  5.22  0.34  0.18  3.06  0.84  3.82  7.38  
2142      0.81 21.20  5.55  2.86 17.17  7.41 17.17 18.70  
2140      0.48  7.95  2.36  1.21  3.30  2.30  4.06  7.63  
2143      0.39  5.22  0.78  0.40  3.06  2.39  3.82  7.38
```

```
-----  
Summary:   2.58 60.79 13.40  6.90 43.76 21.00 46.04 59.79
```

Quelques autres commandes intéressantes

- Comparatif de l'état d'utilisation du CPU par les différents VPS
 - *Vzcpucheck*

Unités de CPU calculées
Selon un algorithme
openvz

```
com10:~# vzcpucheck -v
VEID  CPUUNITS
-----
0      1000
2140   1000
2142   50000
2143   1000
2131   1000
2144   1000
Current CPU utilization: 55000
Power of the node: 319338
```

Quelques autres commandes intéressantes

- Calcul des ressources consommées par un VPS

- **\$ vzcalc**

```
com10:~# vzcalc -v
```

```
Usage: vzcalc [-v] <veid>
```

```
com10: ~# vzcalc -v 2140
```

Resource	Current (%)	Promised (%)
Max (%)		
Low Mem	0.49	7.95
7.95		
Total RAM	2.36	n/a
n/a		
Mem + Swap	1.21	3.30
n/a		
All Loc Mem	2.30	4.06

Rajouter des capacités aux VPS

- Possibilité de changer l'heure aux VPS

```
$ vzctl set 1002 -- capability sys_time: on -- save
```

- Puis pour changer le fuseau horaire, lancer l'utilitaire

```
$ tzconfig (etch) ou
```

```
$ dpkg-reconfigure tzdata (lenny)
```

- Faire un VPS comme client NFS

- sur la machine Hôte, il faut le module NFS chargé... \$ modprobe nfs

- Vérifier que le support NFS est activé dans le kernel :

```
– vzctl exec <ctid> cat /proc/filesystems
```

- Lancer le VPS avec le support NFS

```
– vzctl set 101 -- features "nfs: on" -- save
```

Configuration de openVZ

Configuration openVZ

- Le fichier de configuration générale de openVZ :
/etc/vz/vz.conf

```
## Logging parameters
```

```
LOGGING=yes
```

```
LOGFILE=/var/log/vzctl.log
```

```
LOG_LEVEL=0
```

```
VERBOSE=0
```

```
## Disk quota parameters
```

```
DISK_QUOTA=yes
```

```
VZFASTBOOT=no
```

```
## Template parameters
```

```
TEMPLATE=/var/lib/vz/template
```

```
## Defaults for containers
```

```
VE_ROOT=/var/lib/vz/root/$VEID
```

```
VE_PRIVATE=/var/lib/vz/private/$VEID
```

```
CONFIGFILE="vps.basic"
```

```
DEF_OSTEMPLATE=""
```

```
## IPv4 iptables kernel modules
```

```
IPTABLES="ipt_REJECT ipt_tos ipt_limit ipt_multiport
```

```
iptables_filter iptable_mangle
```

Configuration openVZ

- Les fichiers de configuration des VPS se trouvent dans
 - */etc/vz/conf/<ctid>.conf*
- La configuration est lue au démarrage du CT
- Mais l'utilitaire « *vzctl set* » permet de changer des paramètres à chaud

```
ll /etc/vz/conf
total 60
-rw-r--r-- 1 root root 1763 avr  1 15:22 2022.conf
-rw-r--r-- 1 root root 1754 jui 17 16:09 2103.conf
-rw-r--r-- 1 root root 1754 avr  1 15:28 2103.conf . migrated
-rw-r--r-- 1 root root 2104 jui 17 16:14 2105.conf
-rw-r--r-- 1 root root 2104 avr  1 15:44 2105.conf . migrated
-rw-r--r-- 1 root root 1724 avr  1 15:32 2130.conf . destroyed
-rw-r--r-- 1 root root 1757 avr 10 16:32 2131.conf . destroyed
```

Configuration openVZ

- « *vzctl* » permet de changer des paramètres à chaud
 - *set* CTID [parameters] [--save]
 - Si « *-save* » est donné le parametre est sauvé dans le fichier de configuration */etc/vz/conf/<ctid>.conf*
 - sinon appliqué à chaud sur le VPS

```
$ vzctl set 2140 --i padd 139.124.2.140 --name server  
139.124.2.103 --hostname mapserver --onboot yes --save  
$ vzctl set 2145 --searchdomain com.univ-mrs.fr --save
```

Configuration openVZ

- */etc/vz/conf/<ctid>.conf*
 - *Toute la config des VPS*
 - *Et les beancounters*
- ```
Disk quota parameters (in form of softlimit:hardlimit)
DI SKSPACE=" 1048576: 1153024"
DI SKI NODES=" 200000: 220000"
QUOTATI ME=" 0"

CPU fair scheduler parameter
CPUUNI TS=" 1000"

VE_ROOT="/ var / l i b / v z / r o o t / $VEI D"
VE_PRI VATE="/ var / l i b / v z / p r i v a t e / $VEI D"
OSTEMPLATE=" debi an- 4. 0- i 386- m i n i m a l "
ORI GI N_ SAMPLE=" vps. basi c"
I P_ ADDRESS=" 139. 124. 2. 113"
NAMESERVER=" 139. 124. 2. 113"
NETI F=" "
HOSTNAME=" web"
CAPABI LI TY=" SYS_ TI ME: on "
```

# Configuration openVZ

- Les files system des VPS

- /var/lib/vz/

- *dump* pour les sauvegardes
    - *private* pour les files systems des vps
    - *template* pour les templates

```
ll /var/lib/vz
```

```
total 20
```

```
drwxr-xr-x 2 root root 4096 avr 27 14:32 dump
```

```
drwxr-xr-x 2 root root 4096 sep 4 00:29 lock
```

```
drwxr-xr-x 6 root root 4096 jui 17 16:14 private
```

```
drwxr-xr-x 6 root root 4096 jui 17 16:14 root
```

```
drwxr-xr-x 3 root root 4096 mar 27 11:45 template
```

# Configuration des VPS : vérification

- Vérification de la configuration des VPS par rapport aux ressources physiques réelle existantes
  - Recherche les config dans lesquelles les ressources allouées à un CT dépassent les capacités réelles du système Hôte
  - ***vzcfgvalidate***

```
com10: ~# vzcfgvalidate -i /etc/vz/conf/2144.conf
Validation completed: success
```

# Virtualisation du réseau

# La virtualisation du réseau sous openVZ

- L'interface réseau de OpenVZ est virtualisée. L'interface réseau du HN dédiée aux machines virtuelles s'appelle « **venet** » (virtual network device)
- L'adresse réseau de l'interface venet dans un VPS est paramétrée par la commande:

```
- vzctl set <CTID> -- i padd <P1>[, <P2>, ...] [-- save]
```

- Cette interface réseau se comporte comme une connexion point à point entre le HN et ses VPS, venet n'a pas d'adresse MAC

```
venet0 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
UP BROADCAST POINTOPOINT RUNNING NOARP MTU:1500 Metric:1
RX packets:319210 errors:0 dropped:0 overruns:0 frame:0
TX packets:326010 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:190337564 (181.5 MiB) TX bytes:85793785 (81.8 MiB)
```

```
venet0:0 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:139.124.2.144 P-t-P:139.124.2.144 Bcast:0.0.0.0 Mask:255.255.255.255
UP BROADCAST POINTOPOINT RUNNING NOARP MTU:1500 Metric:1
```

# Interface réseau veth0

- L'autre type d'interface réseau de openVZ est veth0 virtual Ethernet device
- Celle ci se comporte comme une véritable interface ethernet traditionnelle dans un VPS , avec une vraie adresse MAC, une gateway, une table de routage etc...
  - vzctl set <CTID> --netif\_add <ifname>[, <mac>, <host\_ifname>, <host\_mac>, <bridge>]
  - vzctl set 101 --netif\_add eth0 -save
  - vzctl set 101 --netif\_add eth0, 00: 12: 34: 56: 78: 9A, veth101. 0, 00: 12: 34: 56: 78: 9B -- save
- À utiliser lorsqu'on veut faire un « bridge » (niveau 2) entre veth et eth
- Une interface Virtual Ethernet comporte 2 interfaces Ethernet – la première dans le HN CT0 et l'autre dans le CT du VPS. Ces interfaces sont connectées l'une à l'autre comme sur un bridge.. si un paquet parvient à une interface l'autre le recoit également.

# Differences entre venet et veth

- [http://wiki.openvz.org/Differences\\_between\\_venet\\_and\\_veth](http://wiki.openvz.org/Differences_between_venet_and_veth)

| Feature                | veth    | venet   |
|------------------------|---------|---------|
| MAC address            | Yes     | No      |
| Broadcasts inside CT   | Yes     | No      |
| Traffic sniffing       | Yes     | No      |
| Network security       | Low [1] | High    |
| Can be used in bridges | Yes     | No      |
| Performance            | Fast    | Fastest |

- veth permet les broadcasts dans les VPS : à utiliser pour des serveurs ayant besoin de broadcast DHCP, SAMBA etc..
- La sécurité des veth est plus faible.. A déconseiller dans des environnement non secure si on fait du « hosting »
- L'interface venet n'est paramétrée que par le sys admin du HN
- Avec les interfaces veth le parametrage est fait dans le VPS par le sysadmin du VPS .. *CT's user can actually ruin your ethernet network with such direct access to ethernet layer.*

# Gestion des Ressources

p46

# Gestion des ressources

Paramètre de contrôle des ressources

On peut contrôler :

La Gestion des Quotas Disques

Le Partage du CPU

La Gestion des paramètres systèmes

Gestion de la configuration des ressources des VPS

# Parametres beancounters

|      |                                                                                                                                                        |                                                                                                         |                                 |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------|
| DISK | <p>Parametres de gestion des quotas disque dans les VPS</p> <p>On peut activer ou désactiver cette gestion des quotas dans les fichiers de config.</p> | <p>DISK_QUOTA<br/>,<br/>DISKSPACE,<br/>DISKINODES<br/><br/>,<br/>QUOTATIME,<br/>QUOTAUGIDL<br/>IMIT</p> | <p>Managing<br/>Disk quotas</p> |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------|

# Parametres beancounters

|        |                                                              |                                                                                                                                                                                    |                                |
|--------|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| CPU    | Parametres définissant le temps CPU alloué aux VPS           | VE0CPUUNITS, CPUUNITS                                                                                                                                                              | gestion du partage de CPU      |
| System | Parametres de gestion mémoire<br><br>TCP sockets, IP packets | avnumproc, numproc, numtcpsock, numothersock, vmguarpages, kmemsize, tcpsndbuf, tcprcvbuf, othersockbuf, dgramrcvbuf, oomguarpages, lockedpages, shmpages, privvmpages, physpages, | Gestion des paramètres système |

# Paramètres de Gestion des quotas disque

- **disk\_quota** : « on » ou « off » pour activer ou désactiver la gestion des quotas
- **Diskspace** : l'espace disque alloué aux VPS en blocks de 1024
  - `# vzctl set 101 --diskspace 1000000:1100000 --save`
- **Diskinodes** : Nombre total d'inodes (fichiers, répertoire, and symbolic links) qu'un VPS peut contenir
  - `# vzctl set 101 --diskinodes 90000:91000 --save`
- **Quotatime** : la durée en secondes pendant laquelle on permet un dépassement de quotas. Le VPS peut dépasser temporairement son quota pendant « quotatime » secondes, avant qu'il ne soit bloqué
  - `# vzctl set 101 --quotatime 600 --`

# Vérification des quotas

- Vérification de l'état des quotas
  - Vzquota, vzcfgvalidate
  - Valide les paramètres du fichier de config
    - vzcfgvalidate /etc/vz/conf/123.conf
  - Affiche les quotas disques en cours

```
com10: ~# vzquota stat 2140
 resource usage soft limit
hard limit grace
 1k- blocks 754540 10048576
10153024
 i nodes 18898 200000
220000
```

# Gestion des ressources

- En cas de comportement anormal d'un VPS et des services qu'il contient, il faut vérifier les ressources disponibles allouées aux VPS.. une limite est peut être atteinte!
- La premier chose a faire est de consulter les « beancounters »
  - *\$ cat /proc/user\_beancounters*
- La dernier colonne montre un compteur d'échecs (failure) lorsque une ressource a atteint ses limites
  - On vérifie quelle ressources et on peut augmenter l'allocation de cette ressource (sans faire n'importe quoi..)

# Gestion des ressources

- Pour modifier un paramètre « bean counter »
- Regarder ses limites soft (barrier) et hard (limit) et les augmenter
  - *vzctl set 123 --kmemsize \$((2752512\*2)):\$((2936012\*2)) --save*
- Vzctl provoque la modification à chaud
- Avec le flag `--save` on applique les nouveaux paramètres dans le fichier de configuration du VPS
- Pour vérifier la nouvelle configuration
  - *# vzcfgvalidate /etc/vz/conf/<ctid>.conf*

# Gestion des ressources : quota disques

- Exemple pour les quotas disque per VPS
- Les beancounters sollicités sont
  - Diskspace, diskinodes

```
com10: ~# grep -i disk /etc/vz/conf/2131.conf
Disk quota parameters (in form of
softlimit:hardlimit)
DISKSPACE="33045480:36350032"
DISKINODES="200000:220000"
```

- Pour augmenter les limites allouées d'un facteur 2
- `vzctl set 123 --diskspace $(( 33045480*2 )):$((36350032*2 )) --save`

# Gestion des ressources : partage CPU

- 2 paramètres contrôlent l'allocation de CPU
  - `cpuunits` et `cpulimit` (la limite à ne pas dépasser en %)
  - `vzctl set 101 --cpuunits 1000 -save`
  - `vzctl set 102 --cpuunits 2000 -save`
  - `vzctl set 103 --cpuunits 3000 --save`
- Le pourcentage de CPU réel allouée est calculé selon le rapport `cpunit alloué / total`
  - Total `cpuunit` alloué  $1000+2000+3000 = 6000$
  - Le VPS 101 obtient  $1000/6000$  (1/6eme du temp) (16%)
  - Le VPS 102 obtient  $2000/6000$  or 1/3 du temps CPU
  - etc
- Bien lire la doc : [http://wiki.openvz.org/Resource\\_shortage](http://wiki.openvz.org/Resource_shortage)

# Gestion des ressources : partage CPU

- Cpulimit : permet de définir une limite en % de CPU allouée a un VPS
  - `vzctl set 101 --cpulimit 25 --save`
- Indique que le VPS 101 n'aura jamais plus de 25% de CPU alloué (meme si le proc ne fait rien)
- 100% = 1 proc entier pour des dual core ou quad core il faut multiplier par 2 ou 4
-

TP  
à vous de jouer....