http://daffyduke.lautre.net/spip/?OpenVZ-evaluation

OpenVZ evaluation

- 1- Blog-Notes - Au boulot -

Date de mise en ligne : lundi 12 septembre 2011

Rappel sur les différentes techniques de Virtualisation

Machines Virtuelles (VM)

Les machines virtuelles émulent les ressources hardware du serveur réel. Cette émulation entraîne un surcoût de ressources (VMM, instructions CPU privilégiées supplémentaires etc...) mais permet d'acueillir des OS invités sans les modifier puisqu'ils ne se rendent pas compte qu'ils fonctionnent sur un environnement émulé.

Solutions : VMware, QEMU, Microsoft Virtual Server

Paravirtualisation

Cette technique requiert une VMM (voir mon draft sur KVM pour plus de détails) et nécessite de porter les guests OS. Cela engendre un surcoût d'administration mais c'est fiable, sécurisé et cela offre les meillleures performances à l'heure actuelle.

Solutions : XEN, UML

OS Level Virtualisation

Cette dernière technique consiste à considérer un système comme une application à part entière. Le principe est de créer plusieurs instances d'un même guest OS sur un seul et unique OS hyperviseur. Ce dernier est un OS modifié en charge d'administrer, d'isoler et de sécuriser les guests. La contrainte est que les guests OS doivent être identiques, à savoir Linux/Linux dans notre cas.

Solutions : OpenVZ, Virtuozzo, Linux-VServer, Solaris Zones, FreeBSD Jails

La technique de virtualisation la plus performante est XEN. Néanmoins, il semblerait qu'OpenVZ entraîne une perte de performance de 1 à 3%, ce qui est acceptable. Nous allons donc étudier cette solution selon les critères qui nous semblent pertinents (stabilité, performance, administration, industrialisation etc...)

OpenVZ

Vocabulaire

– VE (Virtual Environment) : Un VE peut être un guest OS ou le serveur lui-même (hyperviseur). C'est un système d'exploitation à part entière.
– VE0 : Environnement Virtuel maître (hyperviseur)

Le VE0 est utilisé pour administrer les VEs (processus, fichiers etc...), gérer le hardware ou encore upgrader le

kernel. Le VE0 peut accéder aux VEs, la réciproque n'est pas vraie.

Installation

Création du kernel hyperviseur

Il est nécessaire d'appliquer un patch au kernel hyperviseur. Pour le moment seul un patch 2.6.18 est stable, 2.6.20 et 2.6.22 sont encore en développement.

La conf grub est standard.

Une fois le kernel compilé, installé et le bootloader modifié on reboot.

Ensuite, les paramètres sysctl à appliquer sont les suivants :

```
 net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0
# Source route verification
net.ipv4.conf.all.rp_filter = 1
# pas obligatoire
kernel.sysrq = 1
net.ipv4.tcp_ecn = 0
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

On load les modules qui vont bien :

```
 [root@test_xen /etc/vz/conf]# uname -r
2.6.18-028stab039-openvz
[root@test_xen /etc/vz/conf]# modprobe vzdev
[root@test_xen /etc/vz/conf]# modprobe vzethdev
[root@test_xen /etc/vz/conf]# modprobe vznetdev
[root@test_xen /etc/vz/conf]# modprobe vzdquota
[root@test_xen /lib/modules/2.6.18-028stab039-openvz]# lsmod | grep
vz
vznetdev              18048  1
vzethdev              12296  0
vzmon                  43272  3 vznetdev,vzethdev
vzdquota              41748  1 [permanent]
vzdev                  4868  4 vznetdev,vzethdev,vzmon,vzdquota
[root@test_xen /lib/modules/2.6.18-028stab039-openvz]#
```

Outils indispensables

Les outils suivants sont nécessaires à l'utilisation d'OpenVZ.

– vzctl : Permet l'administration des VEs (création/destruction, démarrage/arrêt, paramétrage etc...)
– vzquota : Permet la gestion des quotas par VE

Configuration

Tout est ici :

```
 [root@test_xen /etc/vz/conf]# ll /etc/vz/
total 20
drwxr-xr-x  2 root root 4096 Oct 24 17:33 conf/
drwxr-xr-x  2 root root 4096 Aug 14 11:12 cron/
drwxr-xr-x  3 root root 4096 Aug 14 11:12 dists/
drwxr-xr-x  2 root root 4096 Aug 14 10:14 names/
-rw-r--r--  1 root root  850 Oct 24 17:22 vz.conf
[root@test_xen /etc/vz/conf]#
```

Modification des variables suivantes :

```
 [root@test_xen /etc/vz/conf]# grep -i CFG
/usr/share/vzpkg/functions
VECFGDIR=/etc/vz/dists/scripts/
VZCFG=/etc/vz/vz.conf
...
```

Par défaut, les arborescences des VEs, les templates de configurations etc... se situent ici :

```
 [root@test_xen /etc/vz]# ll /vz/
total 20
drwxr-xr-x  2 root root 4096 Aug 14  2007 dump/
drwxr-xr-x  2 root root 4096 Oct 24  2007 lock/
drwxr-xr-x  5 root root 4096 Oct 24  2007 private/
drwxr-xr-x  5 root root 4096 Oct 24  2007 root/
lrwxrwxrwx  1 root root  20 Aug 16  2007 template ->
/mnt/openvz/template/
drwxr-xr-x  2 root root 4096 Oct 24  2007 vztmp/
```

J'ai crée ici 2 templates (lfslc1 et lfslc4) :

```
 [root@test_xen /etc/vz]# ll /vz/template/cache/
total 446636
-rw-r--r--  1 root root  46399236 Aug 16  2007
fedora-core-5-i386-minimal.tar.gz
-rw-r--r--  1 root root 145292526 Aug 16  2007 lfslc1.tar.gz
-rw-r--r--  1 root root 265195047 Aug 16  2007 lfslc4.tar.gz
[root@test_xen /etc/vz]#
```

Administration

Création d'un VE

```
 [root@test_xen /etc/vz]# vzctl create 113 --ostemplate lfslc4
--config vps.basic
Unable to get full ostemplate name for lfslc4
Creating VE private area (lfslc4)
Warning: configuration file for distribution lfslc4 not found
default used
Performing postcreate actions
VE private area was created
[root@test_xen /etc/vz]#
```

Cette commande effectue principalement les actions suivantes :

– Lecture des fichiers de configuration par défaut :

```
 [root@test_xen /etc/vz]# ll /etc/vz/vz.conf
-rw-r--r--  1 root root 850 Oct 24 17:22 /etc/vz/vz.conf
[root@test_xen /etc/vz]# ll /etc/vz/conf/ve-vps.basic.conf-sample
-rw-r--r--  1 root root 1558 Aug 14 11:12
/etc/vz/conf/ve-vps.basic.conf-sample
```

– Vérifie l'existence du template spécifié pour la création :

```
 [root@test_xen /etc/vz]# ll /vz/template/cache/
total 446636
-rw-r--r--  1 root root  46399236 Aug 16 12:01
fedora-core-5-i386-minimal.tar.gz
-rw-r--r--  1 root root 145292526 Aug 16 12:21 lfslc1.tar.gz
-rw-r--r--  1 root root 265195047 Aug 16 12:14 lfslc4.tar.gz
[root@test_xen /etc/vz]#
```

– Création du fichier de configuration correspondant au VE 113 :

```
 [root@test_xen /etc/vz]# ll /etc/vz/conf/113.conf
-rw-r--r--  1 root root 1558 Oct 29  2007 /etc/vz/conf/113.conf
```

– Vérifie que la partition cible dispose de l'espace disque nécessaire à la création du VE 113 puis crée le VE :

```
 [root@test_xen /etc/vz]# ll /vz/private/
total 20
drwxr-xr-x  21 root root 4096 Oct 24 17:29 110/
drwxr-xr-x  21 root root 4096 Aug 14 03:58 111.tmp/
drwxr-xr-x  21 root root 4096 Mar 26  2007 112/
drwxr-xr-x  21 root root 4096 Mar 26  2007 113/
drwxr-xr-x  21 root root 4096 Oct 24 17:07 2/
[root@test_xen /etc/vz]#
```

Destruction d'un VE

```
 [root@test_xen /etc/vz]# vzctl destroy 113
Destroying VE private area: /vz/private/113
VE private area was destroyed
```

```
 [root@test_xen /etc/vz]# ll /vz/private/
total 16
drwxr-xr-x  21 root root 4096 Oct 24 17:29 110/
drwxr-xr-x  21 root root 4096 Aug 14 03:58 111.tmp/
drwxr-xr-x  21 root root 4096 Mar 26  2007 112/
drwxr-xr-x  21 root root 4096 Oct 24 17:07 2/
[root@test_xen /etc/vz]#
```

Démarrage d'un VE

```
 [root@test_xen /vz]# vzctl start 113
Warning: configuration file for distribution lfslc4 not found
default used
Starting VE ...
VE is mounted
Setting CPU units: 1000
Configure meminfo: 49152
VE start in progress...
[root@test_xen /vz]#
```

Arrêt d'un VE

```
 [root@test_xen /vz]# vzctl stop 113
Stopping VE ...
VE was stopped
VE is unmounted
[root@test_xen /vz]# ll root/113/
total 0
[root@test_xen /vz]#
```

Liste des VEs

```
 [root@test_xen /vz]# vzlist
      VEID      NPROC STATUS  IP_ADDR         HOSTNAME

      113         12 running               -
```

Entrer/Sortir d'un VE

```
 [root@test_xen /etc/vz/conf]# vzctl enter 113
entered into VE 113
[root@113 /]# exit
logout
exited from VE 113
[root@test_xen /etc/vz/conf]#
```

Configuration du réseau

OpenVZ propose 2 types d'interfaces réseaux : venetet veth. Nous utiliseront vethpour la souplesse d'implémentation.

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

Le bonding est configuré de la sorte :

```
 [root@test_xen /etc/vz/conf]# cat /etc/bonding.cfg

bond0="eth0 eth1"
[root@test_xen /etc/vz/conf]# cat /etc/modprobe.conf
alias eth0 bnx2
alias eth1 bnx2
alias bond0 bonding
options ip_conntrack ip_conntrack_enable_ve0=1
options bonding max_bonds=2 miimon=100 mode=active-backup
[root@test_xen /etc/vz/conf]#
```

```
 [root@test_xen /etc/vz/conf]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=test_xen
GATEWAY=192.168.17.1
GATEWAYDEV=bond0
[root@test_xen /etc/vz/conf]# cat
/etc/sysconfig/network-scripts/ifcfg-bond0
ONBOOT=yes
TYPE=bonding
DEVICE=bond0
IP=192.168.17.52
BROADCAST=192.168.17.255
NETMASK=255.255.255.0
MII_TX_CHECKSUM=on
MII_SPEED=100
MII_DUPLEX=full
[root@test_xen /etc/vz/conf]# cat
/etc/sysconfig/network-scripts/ifcfg-eth0
ONBOOT=yes
TYPE=native
DEVICE=eth0
MII_TX_CHECKSUM=on
MII_SPEED=100
MII_DUPLEX=full
[root@test_xen /etc/vz/conf]# cat
/etc/sysconfig/network-scripts/ifcfg-eth1
ONBOOT=yes
TYPE=native
DEVICE=eth1
MII_TX_CHECKSUM=on
MII_SPEED=100
MII_DUPLEX=full
[root@test_xen /etc/vz/conf]#
```

Configuration d'une interface réseau virtuelle

```
 vzctl set 112 --ipadd 192.168.17.53
```

```
 [root@lfslc4 /]# cat /etc/sysconfig/network-scripts/ifcfg-venet0:0
DEVICE=venet0:0
ONBOOT=yes
IPADDR=192.168.17.53
NETMASK=255.255.255.255
```

Dans ce mode, nous utilisons du source routing et il n'est pas utile de spécifier une adresse MAC pour les VE.

```
 [root@test_xen /etc/vz/conf]# ip r
192.168.17.53 dev venet0  scope link  src 192.168.17.52
192.168.17.54 dev venet0  scope link  src 192.168.17.52
192.168.17.0/24 dev bond0  proto kernel  scope link  src
192.168.17.52
default via 192.168.17.1 dev bond0  metric 1
[root@lfslc4 /]# ip r
192.168.17.0/24 dev venet0  proto kernel  scope link  src
192.168.17.53
default via 192.168.17.1 dev venet0  metric 1
```

Configuration d'un interface ethernet virtuelle

Le VE doit être démarré :

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# vzlist
      VEID      NPROC STATUS  IP_ADDR        HOSTNAME

      112        14 running               -

[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]#
```

Le module kernel suivant doit être chargé :

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# lsmod | grep
vzethdev
vzethdev              12296  0
```

On configure le réseau du VE depuis VE0 en sauvegardant la conficguration :

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# vzctl set 112
--netif_add
eth0,00:0C:29:86:7E:54,veth112.0,00:19:BB:C8:0E:5A --save
```

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# grep ifname
/etc/vz/conf/112.conf
NETIF="ifname=eth0,mac=00:0C:29:86:7E:54,host_ifname=veth112.0,host_mac=00:19:BB:C8:0E:5A"
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]#
```

On configure les interfaces réseaux du VE0 de la sorte :

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# grep conf.eth0
/etc/sysctl.conf
net.ipv4.conf.eth0.forwarding=1
net.ipv4.conf.eth0.proxy_arp=1
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# sysctl -p
```

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# ifconfig
veth112.0 0
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# echo 1 >
/proc/sys/net/ipv4/conf/veth112.0/forwarding
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# echo 1 >
/proc/sys/net/ipv4/conf/veth112.0/proxy_arp
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]#
```

Le routage :

```
 [root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# ip route add
192.168.17.53 dev veth112.0
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]# ip r
192.168.17.53 dev veth112.0  scope link
192.168.17.0/24 dev bond0  proto kernel  scope link  src
192.168.17.52
default via 192.168.17.1 dev bond0  metric 1
[root@test_xen /proc/sys/net/ipv4/conf/veth112.0]#
```

Puis le VE :

```
 [root@112 /proc/sys/net/ipv4/conf/eth0]# ifconfig eth0 0
[root@112 /proc/sys/net/ipv4/conf/eth0]# ip addr add 192.168.17.53
dev eth0
[root@112 /proc/sys/net/ipv4/conf/eth0]# ip route add default dev
eth0
```

Depuis VE0 vers VE et inversement, les transferts réseaux ne sont pas fiables, il y a des pertes de paquets à cause d'un mauvaise propagation/réponses ARP. Ce phénomène ne se produit pas quedans le cas VE0 <-Â» VE.

Ajout des interfaces ethernet virtuelle dans un bridge

Création du bridge :

```
 [root@test_xen ~]# brctl addbr vzbr
```

Ajout des interfaces VE dans le bridge :

```
 [root@test_xen ~]# brctl addif vzbr veth112.0
[root@test_xen ~]# brctl addif vzbr veth113.0
```

Configuration du bridge :

```
 [root@test_xen ~]# ifconfig vzbr 0
[root@test_xen ~]# echo 1 >
/proc/sys/net/ipv4/conf/vzbr/forwarding
[root@test_xen ~]# echo 1 >
/proc/sys/net/ipv4/conf/vzbr/proxy_arp
```

Ajout des informations de routage :

```
 [root@test_xen ~]#ip route add 192.168.17.54 dev vzbr
[root@test_xen ~]#ip route add 192.168.17.55 dev vzbr
```

Configuration bridge + vlan

On configure les vlans sur le VE0. (VLANID + TYPE)

```
 [root@test_xen /proc/sys/net/ipv4/conf]# vzlist
      VEID      NPROC STATUS  IP_ADDR        HOSTNAME

      112         12 running               -

[root@test_xen /proc/sys/net/ipv4/conf]#
```

```
 [root@test_xen /proc/sys/net/ipv4/conf]# vzctl set 112 --netif_add
eth2,00:0C:29:86:7E:56,veth112.2,00:19:BB:C8:0E:5A --save
[root@test_xen /proc/sys/net/ipv4/conf]# vzctl set 112 --netif_add
eth1,00:0C:29:86:7E:55,veth112.1,00:19:BB:C8:0E:5A --save
```

```
 [root@test_xen /proc/sys/net/ipv4/conf]# ifconfig veth112.0 0
[root@test_xen /proc/sys/net/ipv4/conf]# ifconfig veth112.1 0
[root@test_xen /proc/sys/net/ipv4/conf]# ifconfig veth112.2 0

[root@test_xen /proc/sys/net/ipv4/conf]# brctl addbr vzbr104
[root@test_xen /proc/sys/net/ipv4/conf]# brctl addbr vzbr136
[root@test_xen /proc/sys/net/ipv4/conf]# brctl addbr vzbr137

[root@test_xen /proc/sys/net/ipv4/conf]# ifconfig vzbr104 0
[root@test_xen /proc/sys/net/ipv4/conf]# ifconfig vzbr136 0
[root@test_xen /proc/sys/net/ipv4/conf]# ifconfig vzbr137 0

[root@test_xen /etc/vz/conf]# brctl addif vzbr104 veth113.0
[root@test_xen /etc/vz/conf]# brctl addif vzbr136 veth113.1
[root@test_xen /etc/vz/conf]# brctl addif vzbr137 veth113.2
```

```
 [root@test_xen /proc/sys/net/ipv4/conf]# ip route add 192.168.17.54
dev vzbr104
[root@test_xen /proc/sys/net/ipv4/conf]# ip route add
192.168.136.54 dev vzbr136
[root@test_xen /proc/sys/net/ipv4/conf]# ip route add
192.168.137.54 dev vzbr137
```

La configuration des interfaces dans le VE est standard. Si l'on ne souhaite pas avoir d'IPs dans le domaine VE0 associés à chacun des VLANS, il suffit de les supprimer puis de remettre en place le routage.

```
 [root@test_xen /etc/vz/conf]# ifconfig bond0.136
bond0.136      Link encap:Ethernet  HWaddr 00:19:BB:C8:0E:5A
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:432 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:41564 (40.5 Kb)  TX bytes:40628 (39.6 Kb)

[root@test_xen /etc/vz/conf]# ip r
192.168.17.54 dev vzbr104  scope link
192.168.17.55 dev vzbr104  scope link
192.168.137.55 dev vzbr137  scope link
192.168.137.54 dev vzbr137  scope link
192.168.136.55 dev vzbr136  scope link
192.168.136.54 dev vzbr136  scope link
192.168.17.0/24 dev bond0  proto kernel  scope link  src
192.168.17.52
192.168.137.0/24 dev bond0.137  proto kernel  scope link  src
192.168.137.52
192.168.136.0/24 dev bond0.136  scope link
default via 192.168.17.1 dev bond0  metric 1
[root@test_xen /etc/vz/conf]#
```

Une autre configuration est possible, cette fois on associe les IPs du VE0 au bridge puis on ajoute les interfaces des VE et du VE0 dans le bridge. Cette configuration est la meilleure possible, elle fonctionne dans tous les cas, les VE0 peuvent communiquer avec les VE sans problème.

```
 [root@test_xen ~]# brctl show
bridge name    bridge id              STP enabled    interfaces
br104          8000.0019bbc80e5a      no              bond0
                                                        veth112.0
                                                        veth113.0
br136          8000.0019bbc80e5a      no              bond0.136
                                                        veth112.1
                                                        veth113.1
br137          8000.0019bbc80e5a      no              bond0.137
                                                        veth112.2
                                                        veth113.2
[root@test_xen ~]#
```

Aucune règle spécifique de routage n'est nécessaire :

```
 [root@test_xen ~]# ip r
192.168.17.0/24 dev br104  proto kernel  scope link  src
192.168.17.52
192.168.137.0/24 dev br137  proto kernel  scope link  src
192.168.137.52
192.168.136.0/24 dev br136  proto kernel  scope link  src
192.168.136.52
default via 192.168.17.1 dev br104  metric 1
[root@test_xen ~]#
```

Les IPs sont montées sur les bridge :

```
 [root@test_xen ~]# ifconfig br136
br136          Link encap:Ethernet  HWaddr 00:19:BB:C8:0E:5A
          inet addr:192.168.136.52  Bcast:192.168.136.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0
          TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5656 (5.5 Kb)  TX bytes:6790 (6.6 Kb)

[root@test_xen ~]# ifconfig br137
br137          Link encap:Ethernet  HWaddr 00:19:BB:C8:0E:5A
          inet addr:192.168.137.52  Bcast:192.168.137.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2660 (2.5 Kb)  TX bytes:3150 (3.0 Kb)

[root@test_xen ~]# ifconfig br104
br104          Link encap:Ethernet  HWaddr 00:19:BB:C8:0E:5A
          inet addr:192.168.17.52  Bcast:192.168.17.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:908960 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23524518 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:0
          RX bytes:45173730 (43.0 Mb)  TX bytes:1445451892 (1378.4
Mb)

[root@test_xen ~]#
```

Il est aussi possible de ne pas monter d'IPs sur les bridge.

Configuration de keepalived

Les VEs sont des 'jails' améliorées, par conséquence il n'est pas possible d'accéder aux modules kernel depuis ces instances. La seule solution est de créer un keepalived depuis le VE0 puis de load-balancer sur les instances VE. La configuration réseau à retenir est d'associer les IPs du VE0 directement au bridge comme précisé ci-dessus.

Configuration d'IPtables

IPtables fonctionne parfaitement et indépendamment sur chacun des VE0 et VEs.