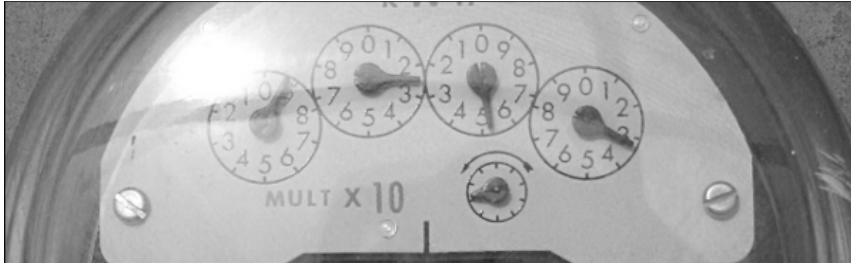


janvier 10th, 2010

OpenVZ (Partie 2)



Nous allons maintenant voir les paramétrage de OpenVZ de façon plus poussée, au niveau de la gestion des ressources. Nous allons en effet voir comment nous pouvons partager le processeur avec les VE et comment gérer les ressources systèmes. Nous parlerons aussi des quotas disques, mais nous reviendrons sur les redimensionnement de disques LVM qu'un peu plus tard dans un autre billet.

La gestion du processeur

Il existe 4 manières de paramétrer le partage de votre CPU:

- `cpus`: cela vous permet de choisir combien de cores vous autorisez votre conteneur à utiliser.
- `cpulimits`: il s'agit du temps processeur que l'on veut allouer à notre conteneur. Il s'exprime en pourcentage et chaque core d'un processeur représente 100%. Ainsi, la charge potentielle totale d'un processeur quad-core serait donc de 400%.
- `cpuunits`: semblable à `cpulimit`, sauf que le nombre ne s'exprime plus en pourcentage, mais avec un nombre entier. Le total disponible correspondra au total de vos « `cpuunits` » alloués. Si vous allouez 100 à une VE, puis 200 à une autre, alors le premier environnement utilisera $100/(100+200)$ des ressources CPU, soit 1/3. Le second aura 2/3 des ressources.
- `ve0cpuunits`: même concept que `cpuunits`, sauf qu'il s'agit du temps cpu alloué au à la machine hôte. Il faut essayer de lui réserver entre 5% et 8% au minimum.

Vous comprendrez que le plus simple reste d'allouer un certain nombre de core aux VE, ou bien de déterminer un pourcentage de partage. Par exemple, disons que nous voulons attribuer 3 cores à une VE de CTID 100, il suffirait alors de taper:

```
# vzctl set 100 --cpus 3 --save
```

De la même façon, mettons que nous voulons attribuer la moitié des ressources d'un CPU quad-core, nous attribuerons alors l'équivalent de 2 CPU, soit 200% de temps CPU:

```
# vzctl set 100 --cpulimits 200 --save
```

Maintenant nous pourrions mixer. On attribue 2 cores pour une application multi-threadé, mais n'utilisons que 10% des ressources totale du CPU quad core (soit... 40% des 400%):

```
# vzctl set 100 --cpus 2 --cpulimits 40 --save
```

Oui, l'intérêt n'est pas évident, mais c'est juste un exemple. ^_^

La gestion du disque

Parmi les réglages disponibles pour la gestion des disques virtuels, seuls 3 me semblent intéressants: l'espace disque, le nombre d'inodes, et le temps de quotas.

Pour déterminer un espace disque dans l'environnement virtuel, il faut utiliser l'option « `diskspace` ». La structure est la suivant: « `vzctl set CTID --diskspace limite_basse:limite_haute` ».

```
# vzctl set 100 --diskspace 29G:30G --save
```

Dans notre exemple, 30 Go est une limite implacable que l'environnement virtuel ne pourra pas dépasser. 29 Go est en revanche est une limite temporaire, qui ne peut être excédée qu'un certain temps seulement. Au delà de ce temps, nous ne pourrions plus écrire de données supplémentaires sur le disque. Ce temps de tolérance est de 10 minutes par défaut, mais il est possible de le changer. Par exemple, réduisons-le à 5 minutes (300 secondes):

```
# vzctl set 100 --quotatime 300 --save
```

Nous pouvons consulter l'état du disque de la VM avec `vzquota`:

```
# vzquota stat 123
resource      usage      softlimit   hardlimit   grace
1k-blocks     327664     1048576     1153434
```

inodes	18857	200000	220000
--------	-------	--------	--------

Le dernier réglage qui pourrait être intéressant serait le nombre d'inodes. D'après wikipedia, à chaque fichier est associé un inode qui connaît son propriétaire, son groupe, les permissions et le type du fichier. Donc plus vous aurez d'inodes, plus vous pourrez manipuler de fichiers. En pratique vous ne deviez pas avoir besoin d'y toucher. Personnellement, j'ai dû y revenir lorsque j'effectuais des tests disque. :-p

```
# vzctl set 100 --diskinodes $((200000)):$((220000)) --save
```

Gestion de la mémoire vive

Elle est facile celle-là: je vous l'ai déjà donné dans le guide d'installation d'openVZ! :-)

```
# vzctl set 100 --privmmpages 280M:300M --save
```

Gestion des autres ressources

Enfin viennent les ressources systèmes, dont je ne connais rien du tout! Nous pouvons trouver un fichier de stat dans `/proc/beancounters`. La documentation d'OpenVZ recommande juste d'observer la valeur des éléments donc le « compteur d'échec » (failcnt). Si un élément venait à être supérieur à 0, il faudrait alors l'augmenter.

```
cat /proc/user_beancounters
```

1	Version: 2.5					
2	uid resource	held	maxheld	barrier	limit	failcnt
3	100: kmemsize	836919	1005343	2752512	2936012	0
4	lockedpages	0	0	32	32	0
5	privmmpages	4587	7289	49152	53575	0
6	shmpages	39	39	8192	8192	0
7	dummy	0	0	0	0	0
8	numproc	20	26	65	65	0
9	physpages	2267	2399	0	2147483647	0
10	vmguarpages	0	0	6144	2147483647	0
11	oomguarpages	2267	2399	6144	2147483647	0
12	numtcpsock	3	3	80	80	0
13	numflock	3	4	100	110	4
14	numpty	1	1	16	16	0
15	numsiginfo	0	1	256	256	0
16	tcpsndbuf	0	0	319488	524288	0
17	tcprcvbuf	0	0	319488	524288	0
18	othersockbuf	6684	7888	132096	336896	0
19	dgramrcvbuf	0	8372	132096	132096	0
20	numothersock	8	10	80	80	0
21	dcachesize	87672	92168	1048576	1097728	0
22	numfile	238	306	2048	2048	0
23	dummy	0	0	0	0	0
24	dummy	0	0	0	0	0
25	dummy	0	0	0	0	0
26	numiptent	10	16	128	128	0

Dans cet exemple, nous voyons que le paramètre `numflock` à eu des échecs de fonctionnement. Nous pouvons augmenter ses limites en utilisant « `vzctlset CTID -nom_du_parametre valeur_souple:valeur_dure` » :

```
# vzctl 100 set --numflock $((150)):$((155)) --save
```

Partage des ressources équitables

Il nous est aussi possible de partager équitablement les ressources entre les environnements dès leur création. Pour cela, nous pouvons utiliser la commande « `vzsplit` » suivie par l'argument `-n` et un nombre. Ce nombre correspond à la portion du système que nous désirerions utiliser (1/5ème dans cet exemple).

```
# cd /etc/vz/conf/
# vzsplit -n 5 -f vps.testsplit
Config /etc/vz/conf/ve-vps.testsplit.conf-sample was created

# vzcfgvalidate /etc/vz/conf/ve-vps.testsplit.conf-sample
Validation completed: success
```

```
# vzctl create 101 --ostemplate debian-5.0-i386-minimal --config vps.testsplit
Creating VE private area (debian-5.0-i386-minimal)
Performing postcreate actions
VE private area was created
# vzctl start 101
Starting VE ...
VE is mounted
Setting CPU units: 26812
Configure meminfo: 153965
VE start in progress...
# vzcalc 101
Resource      Current(%)  Promised(%)  Max(%)
Memory        0.99        21.00        25.30
```

Le mot de la fin

Et voilà! Rien de bien compliqué, mais il est intéressant de savoir comment changer les valeurs utiles pour l'administration de tout le jours. La prochaine partie consistera à la redimensionnement d'une partition LVM avec un conteneur OpenVZ dedans.

Sources

- > wiki.openvz.org/User_Guide/Managing_Resources
- > wiki.openvz.org/Resource_shortage
- > download.openvz.org/doc/OpenVZ-Users-Guide.pdf

Tags: *openvz, virtualization*

Posted in [openvz](#), [virtualisation](#) | [No Comments](#) »

[Commentaires \(0\)](#)