

Virtualisation Linux : la solution OpenVZ

Agence universitaire de la Francophonie

26 mars 2009

Plan

1 Introduction

2 Premiers pas

1 Introduction

2 Premiers pas

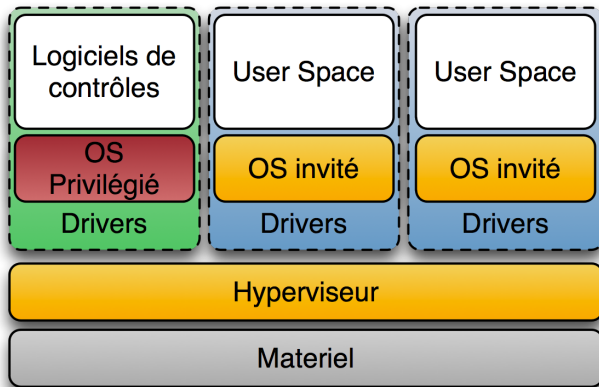
Généralités sur la virtualisation (1/2)

Différents types de virtualisation

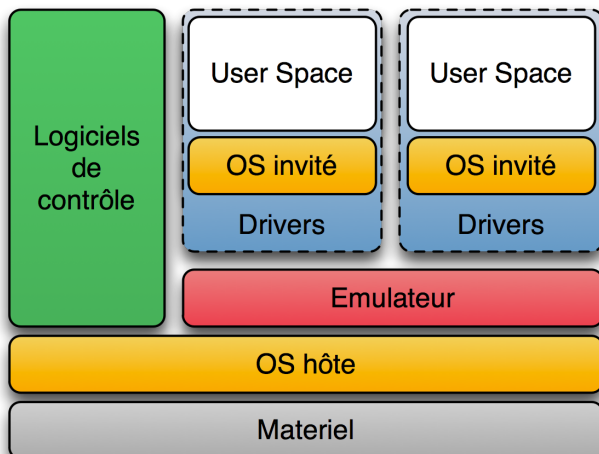
- Paravirtualisation (hyperviseur) : Xen, VMware ESX, Microsoft
- Emulation (machine virtuelle) : QEMU/KVM, VirtualBox, VMware, Bochs, Parallels
- Isolation : FreeBSD Jail, Solaris Zones/Containers, Linux-VServer, OpenVZ, Virtuozzo
- supports matériels (Intel VT, AMD-V, HP Superdome, Sun LDOM)

Ce n'est qu'une tentative de classification. Voir Wikipédia...

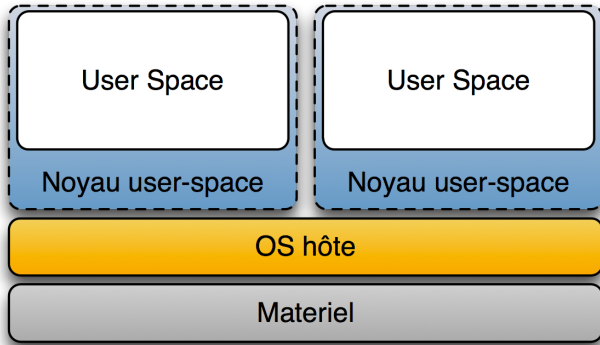
Paravirtualisation (hyperviseur)



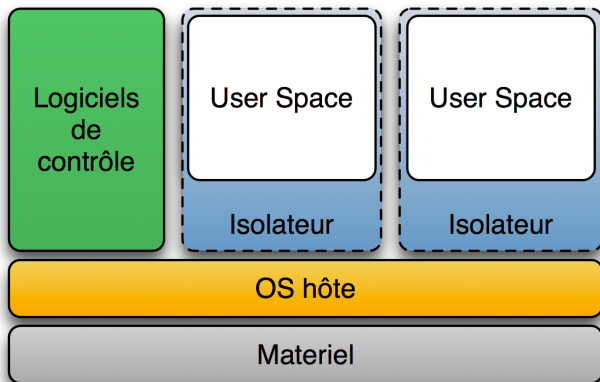
Emulation (machine virtuelle)



User space



Isolation (le cas d'OpenVZ)



Généralités sur la virtualisation (2/2)

Objectifs de la virtualisation

- Consolidation
- Sécurisation
- Hébergement
- Développement et tests
- Education
- ... et tout ce qu'on peut imaginer quand le nombre de machines est « illimité »

OpenVZ en quelques lignes (1/4)

- Virtualisation niveau OS
- Base de Parallels Virtuozzo Containers
- Supporté par Parallels (support payant possible : sales@openvz.org)
- GPL

Fonctionnalités

- Penser à un « super chroot ». Vraiment super.
- Isolation niveau OS : un seul noyau, plusieurs systèmes
- Linux-only, mais multi-distributions (sur même noyau)
- Gestion de ressources par machine : CPU, RAM, disque
- Checkpoints : migration de machine

OpenVZ en quelques lignes (2/4)

Isolation des objets

- Fichiers (y compris /proc, /sys, /dev, etc)
- Utilisateurs et groupes (y compris root)
- Processus
- Réseau et interfaces réseau
- Périphériques (on peut même attribuer un périphérique à une seule machine)
- IPC (mémoire partagée, sémaphores, messages, etc)

OpenVZ en quelques lignes (3/4)

Gestion des ressources

- Quota disque (espace et nombre d'inodes). Quotas unix toujours possibles
- Ordonnanceur «équitable», par machine (avec priorité et limites) et ordonnanceur habituel dans chaque machine
- Les *beancounters* : compteurs, limites et garantie sur un grand nombre de ressources
- But : donner toute les ressources possibles à chaque machine virtuelle, sans qu'une seule puisse accaparer tout la machine hôte.
- La plupart de ces paramètres sont modifiables à la volée

OpenVZ en quelques lignes (4/4)

Checkpoint

- point d'arrêt complet
- permet de passer une machine virtuelle sur un autre hôte (en quelques secondes : pas d'arrêt du service)

Outils

- vzctl : start, stop, set, ...
- Modèle de machine (*templates*) : fournis, mais aussi faciles à construire (personnalisation)

1 Introduction

2 Premiers pas

Création d'une machine

Paramètres à décider

- un numéro de machine (supérieur à 100)
- `--ostemplate` : le modèle
- `--hostname` : nom de la machine
- `--ipadd` : adresse IP de la machine
- `--config` : configuration modèle (valeurs de base)

Résultat : `vzctl create 101 --ostemplate
debian-lenny-ucad --hostname ucad101`

Lancement, arrêt, liste...

Lancement d'une machine virtuelle (container)

```
vzctl start 101
```

Arrêt

```
vzctl stop 101
```

Liste des containers

```
vzlist et vzlist -a
```

Entrer dans une machine

Entrer dans une machine

```
vzctl enter 101
```

... et essayer quelques commandes :

- hostname
- ps waux
- ls -l /boot
- ls -l /dev

Réseautique...

Ajouter une adresse réseau

```
vzctl set 101 --ipadd 172.16.0.101 --save
```

Etudier la configuration obtenue (et regarder au niveau du *netfilter* de la machine hôte)

TP : une seconde machine, des services séparés

- Créer une seconde machine ucad102, adresse IP 172.16.0.102
- Installer apache2 sur ucad101
- Installer proftpd sur ucad102
- Faire le tour des processus et des ports IP ouverts, sur ucad101, ucad102 et sur la machine hôte

Où sont les données ?

- les configurations : `/etc/vz/conf/101.conf`
- `/var/lib/vz` (ou `/vz`)
- les répertoires `private` et `root`
- les `template`