

%BEGINLATEX%

Présentation OpenVZ

Marc SCHAEFER

Revision: r6 - 30 Dec 2008 - 18:25:03 - [MarcSCHAEFER](#)

Le but de cette présentation est de montrer ce qu'est la virtualisation et ce que peut apporter OpenVZ dans le contexte usuel de virtualisation de systèmes standards.

%STOPLATEX% %STOPPUBLISH%



%STARTPUBLISH%

[Slide 1: Cet atelier](#)

[Slide 2: La virtualisation](#)

[Slide 3: Les types de virtualisation](#)

[Comments](#)

[Slide 4: Les avantages de la virtualisation](#)

[Slide 5: Les inconvénients de la virtualisation](#)

[Comments](#)

[Slide 6: Terminologie](#)

[Slide 7: OpenVZ](#)

[Slide 8: La solution OpenVZ proposée ici](#)

[Comments](#)

[Slide 9: Vos buts](#)

[Slide 10: Démonstration](#)

[Slide 11: "Best practices"](#)

[Slide 12: Sujets de réserve](#)

[Slide 13: Discussion](#)

[Slide 14: Références](#)

%STARTLATEX% %LATEXTOC%

Slide 1: Cet atelier

- vos raisons d'être ici
- vos besoins et intérêts
- Atelier OpenVZ

Slide 2: La virtualisation

- technique assez ancienne, *enfin* disponible sous environnement PC

- avantages principaux
 - mutualisation des ressources
 - *scalability* et flexibilité
 - peut être un atout pour des solutions HA (haute disponibilité)
 - aide à la migration de systèmes
 - économies d'énergie

très à la mode!

Slide 3: Les types de virtualisation

diffèrent par

- degré d'isolation
- ce qui est virtualisé
- impact sur la performance
- pré-requis matériel (p.ex. processeur virtualisable)

degré d'isolation

- isolation (à différents degrés)
 - chroot(2) UNIX (et BSD jail)
 - linux-vserver (VLS)
 - OpenVZ (containers)
- virtualisation partielle du CPU, coopération pour le matériel (Xen avec CPU compatible)
- virtualisation du CPU, et du matériel (VMware, virtualbox)
- émulation de processeur et du matériel
 - qemu, virtualbox

Comments

Par exemple, le logiciel kvm (Kernel-based Virtual Machine) disponible dès Linux 2.6.15, peut utiliser la virtualisation de CPU complète des processeurs récents Intel ou AMD si elle est disponible:

```
schaefer@reliant:~$ uname -a
Linux reliant 2.6.24-22-generic #1 SMP Mon Nov 24 18:32:42 UTC 2008 i686 GNU/Linux
schaefer@reliant:~$ egrep '^flags.*(vmx|svm)' /proc/cpuinfo
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
                dts acpi mmx fxsr sse sse2 ss ht tm pbe nx lm constant_tsc arch_perfmon pebs bts
                pni monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr sse4_1 lahf_lm ida
```

Sinon, KVM fonctionne exactement comme qemu (sans kqemu)

Slide 4: Les avantages de la virtualisation

- (tests de) migration logicielle ou matérielle
- coexistence de différents OS et versions de logiciels
 - p.ex. 32 bits / 64 bits

- version ou distribution différentes
- plus grande indépendance du matériel (encore que!)
- meilleure rentabilisation des investissements
- options HA

Slide 5: Les inconvénients de la virtualisation

encore que

- passer de 5 systèmes réels à une machine réelle et 5 machines virtuelles
 - ça fait un système de plus à gérer (mises à jour, sécurité)
- un crash du système hôte et tout est en panne, corollaires:
 - pour patcher le kernel du système hôte il faut le redémarrer
- certains types de virtualisation diminuent les performance, ou n'offrent pas une virtualisation suffisante dans certains cas (p.ex. pas de virtualisation de Microsoft Windows).
- est-ce vraiment plus qu'un simple `chroot(1)` dans certains cas ?

Comments

Cependant, la virtualisation peut amener des solutions au problème de la mise à jour logicielle ou matérielle du système hôte, notamment via les outils de migration de VM online sans arrêt des services ni des connexions et un système matériel de réserve.

Exemple:

- économie d'énergie par consolidation
 - de jour, l'entreprise a besoin de 5 serveurs réels pour gérer 10 machines virtuelles
 - de nuit, toutes les VM sont migrées sur une seule machine physique et les autres machines sont éteintes automatiquement

Slide 6: Terminologie

Acronyme description

VM *virtual machine*: machine virtuelle (hébergée, virtualisée)

host système hébergeant: machine physique

COW *copy-on-write*: se dit de données partagées en lecture et qui lorsqu'elles sont écrites par un des partenaires, le sont de façon privées sans répercuter les modifications aux autres partenaires. C'est un système de stockage (plus) efficace). Malheureusement il dérive avec le temps. Une stratégie plus efficace dans le cas de GNU/Linux serait de partager

équivalents

guest, VZ, VPS, instance, container, ...

VS

en lecture seulement le système, puis de faire les mises à jour sur un seul des systèmes. On peut l'implémenter de diverses manières: *snapshot* LVM, unionfs, etc

Slide 7: OpenVZ

- produit de virtualisation **Linux/Linux** libre
 - *isolation* avancée, avec un seul kernel (*container*)
 - non intégré aux kernels des distributions
 - sous Debian, patching facilité par les packages standards et make-kpkg
 - supporte tout sauf la virtualisation du matériel ou du CPU (single-kernel virtualization)
 - fonctionnellement équivalent à la meilleure des solutions propriétaires de virtualisation -- si Linux/Linux suffit.
- marché principal: hébergeurs de services
- existe aussi sous forme propriétaire: Virtuozzo
 - GUI d'administration
 - support
 - très populaire parmi les hébergeurs de masse

Slide 8: La solution OpenVZ proposée ici

- package CRIL `openvz-support`
 - préparation du kernel patché
 - génération de templates
 - scripts de support à la maintenance
 - dépend de `vztools` Debian
 - encore quelques petits problèmes de jeunesse, mais largement fonctionnel!
- images kernels binaires déjà préparées pour Debian stable (pour gagner du temps)

Comments

Inconvénients

- pas de package standard en Debian stable, patch de 2.6.18 difficile
 - le patch n'applique pas proprement: évt. utiliser (sans garantie) `kernel-patch-openvz` de `packages.cril.ch`
 - pas de support inotify

Solutions

- utiliser le package proposé lors de l'atelier

Slide 9: Vos buts

A faire par les participants

- utilisation des outils de base (vz*) et des scripts de openvz-support
 - génération du kernel binaire
 - génération et utilisation de templates
 - configuration de la surveillance système (nullmailer, logcheck, inform-updates).
- connaître les principaux fichiers de configuration et les commandes de gestion
- utiliser un concept simple d'adressage réseau pour les VM
- créer un template
- créer une VM à partir d'un template créé
- administration d'une VM (démarrage, arrêt, etc)
- installer un service Apache et y accéder
- dupliquer une machine virtuelle ou réelle

Optionnel

- installer un système OpenVZ à partir d'une installation par défaut standard Debian stable (etch), en utilisant le kernel binaire déjà préparé
- la sauvegarde (sans et avec checkpointing)
- le partage de données entre VM
 - cas d'espèce: /home partagé

Au travail!

Slide 10: Démonstration

Si j'ai réussi à le préparer:

- migrer une VM atomiquement d'une machine à l'autre avec `vzmigrate`, alors que les connexions réseaux sont maintenues

Slide 11: "Best practices"

- utiliser LVM et allouer un volume logique à chaque VM, monté sous `/var/lib/private/NUM`
- ajouter un serveur de mail minimal dans chaque VM (`nullmailer`) et le configurer pour délivrer au serveur de mail de l'entreprise (ou à défaut: de la machine hébergeante)
- ajouter un script de mises à jour automatique dans les VM, ou au minimum une notification par e-mail
- ajouter une surveillance standard des logs de chaque VM (`logcheck`)
- ajouter une surveillance globale de l'espace disque des VMs

- serveur DNS pour configuration des noms (ou rsync des /etc/hosts)
- documenter les actions

Slide 12: Sujets de réserve

- configuration avancée de firewall
- éviter le "downtime" à l'aide d'une paire de serveur en HA (panne) ou avec migration manuelle (prévision) avant p.ex. upgrade du kernel ou ajout de RAM
- hébergement de volume: les diverses stratégies
 - partage des pages
 - / read-only partagé
 - copy on write
- le concept de l'interface `venet0`
- intégration de plusieurs machines physiques
- sécurisation
 - `ip_forwarding`
 - IPv6

Slide 13: Discussion

... ouverte

Slide 14: Références

- <http://wiki.openvz.org/Category:HOWTO>

%ENDLATEX% %STOPPUBLISH%

Slide 15: Administration

- Set THISDOCUMENTNAME = Présentation OpenVZ

-- [MarcSCHAEFER](#) - 21 Dec 2008 %STARTPUBLISH%

This topic: Main > [TWikiUsers](#) > [MarcSCHAEFER](#) > [MarcSCHAEFERLeftBar](#) > [MarcSCHAEFERHomePage](#) > [MarcSCHAEFERConferencesLinuxNE](#) > [OpenVZ](#) > [OpenVZPresentation](#)
History: r6 - 30 Dec 2008 - 18:25:03 - [MarcSCHAEFER](#)

Copyright © by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? [Send feedback](#)

