

V12N

avec Xen et IBM BladeCenter

- Cas d'étude :
Plateforme d'hébergement Plone/Zope
- Pourquoi Xen ?
- Pourquoi BladeCenter ?
- Conclusions

Contexte

- 10 à 100 serveurs physiques
- Configurations matérielles disparates
- Configurations logicielles disparates
- Besoin de performances
... mais *bottleneck* non identifié
 - CPU ?
 - RAM ?
 - I/O ?

Xen – aspects techniques

- Hyperviseur en version 3.2
(packaging Debian/Ubuntu)
- Noyaux :
 - dom0 – léger *lag*, mais bon support hw
 - domU – peut utiliser le kernel *mainstream*
- Performances :
 - disque : natif
 - CPU/mémoire : 90 à 98% natif
 - réseau : limitations (relatives)

Pourquoi Xen ?

Raisons *objectives*

- Besoin de performances
Exclut les solutions *full HVM*
(qemu, kqemu, parallels, VMWare ...)
- Sécurité et isolation
Exclut les solutions à *namespace*
(Virtuozzo, OpenVZ, VServer ...)
- Utilisation d'outils Open Source
Pour créer des outils *in-house*
(Scripts de déploiement, supervision ...)

Pourquoi Xen ?

Raisons *subjectives*

- Forte culture Debian GNU/Linux
- Utilisation d'outils standard
 - Interconnexion réseau : bridges
 - Configuration : *flatfiles* en Python
 - Disques : gérés par le dom0 ou le domU
(DRBD, AOE, iSCSI, device-mapper, LVM, softraid, ...)

Pourquoi Xen ?

Raisons économiques

Open Source réel :

La version libre est fonctionnelle

Nombreux contributeurs

(individus + entreprises)

Segmentation (presque) claire entre

- les versions gratuites (Shell!)

- les versions payantes (IHM!)

Pouvoir passer de *smallscale* à *largescale*

... sans remettre en question le socle

Le futur de Xen

- Hyperviseur 3.3 déjà disponible
(améliorations sur le HVM, la migration ...)
- Xen 4/3.4 : mini-domaines spécialisés I/O
(devrait lever les limitations évoquées)
- En revanche ...
... Adoption mitigée par les distributions
- Fusion avec KVM ?

IBM BladeCenter : aspects techniques

- Châssis 7U / 14 lames
- Double connectivité gigabit par lame
- Gestion des lames et des *switches* via l'AMM
(Advanced Management Module)
Prise de contrôle (clavier+écran) à distance
- Chaque lame embarque
 - 2 CPU quad-core
 - 24 Go de RAM
 - 2 disques 2 pouces

IBM BladeCenter : mise en œuvre

podp : *Power-On-Demand Platform*

- Démarrage avec 2 lames
- Ajout de lames avec +/- de CPU/RAM
- Disques locaux (si besoins faibles),
ou SAN (si besoins plus élevés)

IBM BladeCenter : avantages

- Intégration des éléments vitaux :
 - console déportée,
 - *switches* administrables + redondants,
 - alimentations administrables + redondantes
- Évolution modulaire
(ajout de lames, RAM, CPU, disques)
- Déploiement physique plus simple et rapide

IBM BladeCenter : intégration et évolution

- Un chassis BladeCenter
= une machine (presque) comme les autres
- Utiliser la virtualisation comme abstraction
- L'AMM intègre KVM-IP + NPDU,
avec une forte valeur ajoutée
- IBM n'est pas seul sur le marché des *blades*

Résultats

- Infrastructure matérielle plus saine
 - Virtualisation des machines obsolètes
 - Réutilisation des machines homogènes
- Infrastructure logicielle inchangée
 - kernel-image-2.6.24-xen
- Procédures *ad hoc* de déploiement, migration, sauvegarde, clonage ...
- Deux fois plus de serveurs virtuels, dans deux fois moins de serveurs physiques

Conclusions

- Xen : une brique logicielle de virtualisation, riche en fonctionnalités, puissante, ...
Et toutefois interchangeable
- IBM BladeCenter : une plateforme matérielle, riche en fonctionnalités, puissante, évolutive ...
Et toutefois interchangeable (aussi!)
- podp : grâce à Xen et IBM,
le « métal » devient un consommable !
(principe du *cloud computing*)