



Virtualisation sous Linux une évolution maîtrisée ?

Daniel Veillard
veillard@redhat.com

Une jungle d'acronymes

Xen

QEmu

OpenVZ

UML

VServer

KVM

VMWare

VirtualIron

VirtualBox

Iguest

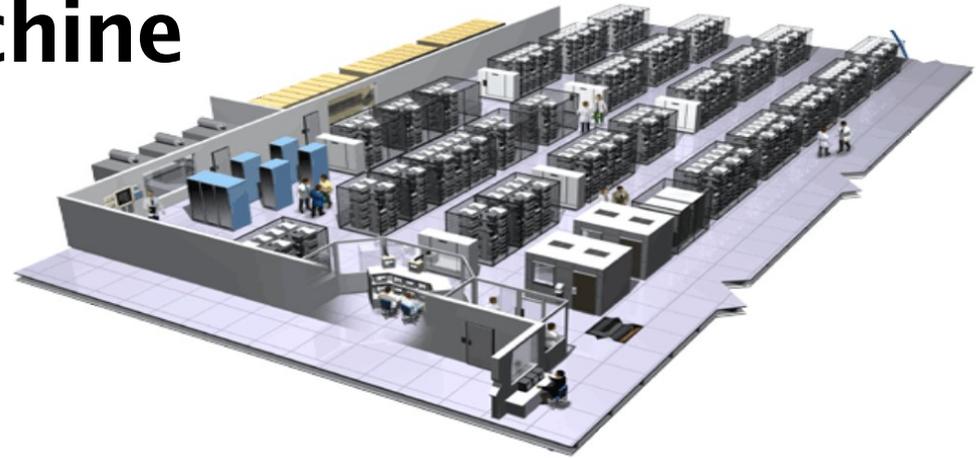
Définition et but

- Définition:
 - Découpler les ressources informatiques de leur implémentation
- But: abstraire le matériel pour s'en affranchir
- Machine Virtuelle
 - Processeurs virtuels
 - Mémoire
 - Périphériques réels ou émulés
- Stockage Virtualisé
 - Disque distants
 - Disques locaux émulés
- Transition du multitâche au multi-OS

Benefices de la virtualisation

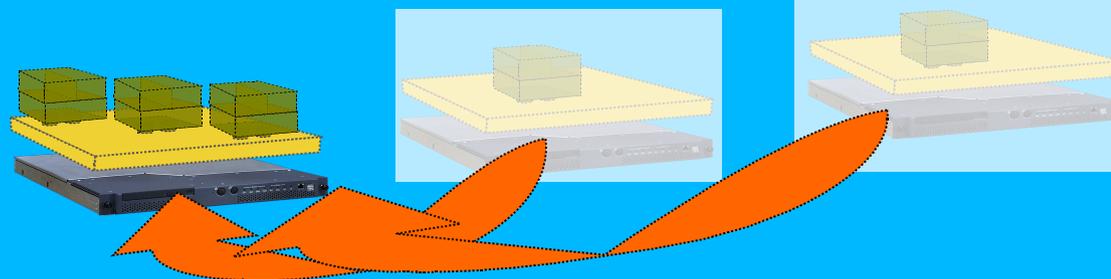
- Les bénéfices sont de deux ordres :
- Réduction des couts
 - Meilleure gestion de la capacité
 - Simplification de la maintenance
 - Mise a l'échelle
- Réduction des risques
 - Isolation et sécurité des services
 - Indépendance matérielle
 - Redondance aisée

Exemple: une salle machine

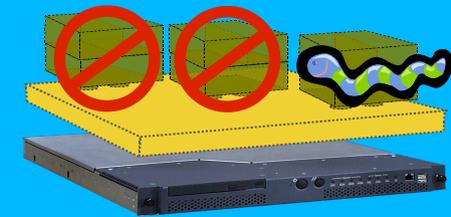


- Contraintes physiques
 - Taille, énergie, climatisation
- Surdimensionnée
 - Utilisation moyenne 10 - 30%
- Un serveur par application/client
 - Simplification de la gestion, facturation
 - Sécurité
- Problèmes
 - Coûts
 - Administration

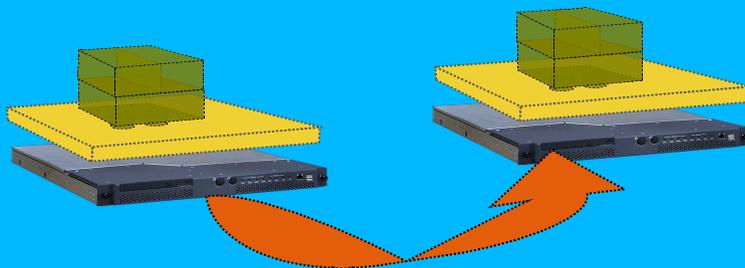
La virtualisation en action



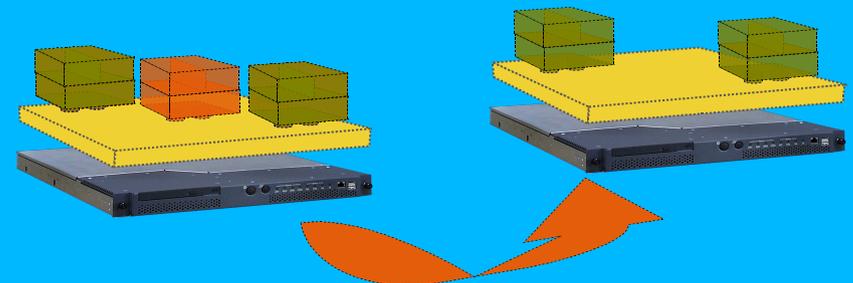
Consolidation de serveurs sous-utilisés



Politique de
Sécurisation



Pas d'interruption de service,
migration



Répartition dynamique de
charge

Des virtualisations

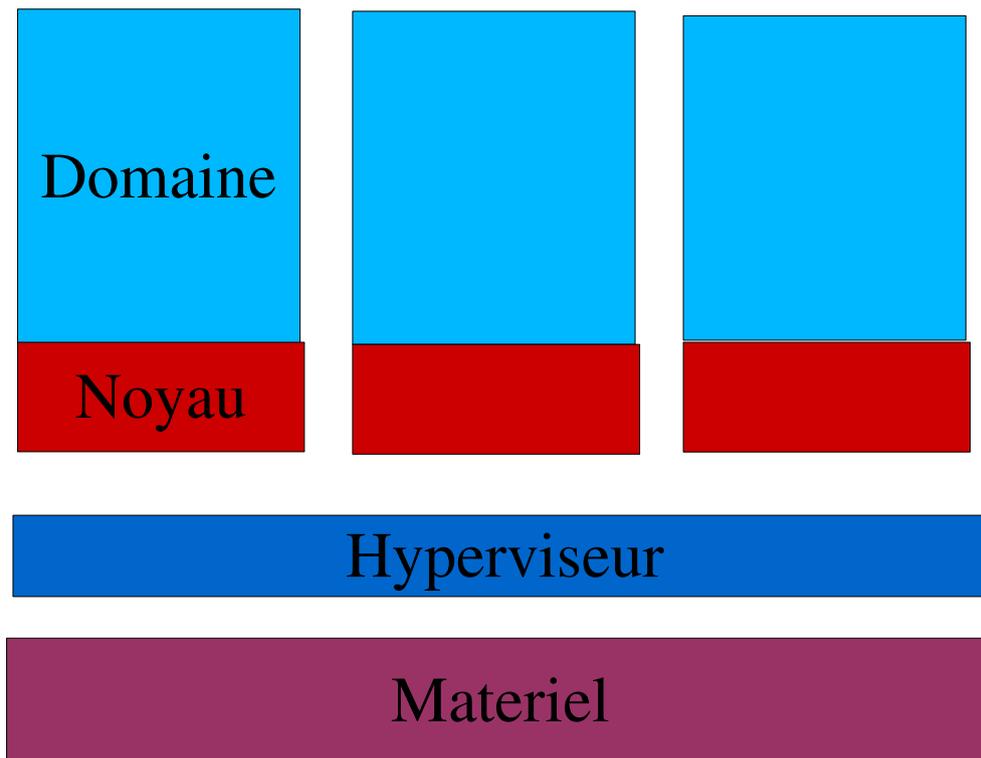
- Émulateurs:
 - simulation complète de la machine ex. QEmu, Hercules
- Virtualisation complète:
 - Réutilise le processeur natif quand possible
 - OS client non modifié, requiert support matériel
- Para-virtualisation:
 - OS client recompilé pour la machine virtuelle
- Conteneurs:
 - Un seul OS, avec des primitives de cloisonnement
- Autres:
 - Ré-écriture au vol (VMWare)
 - Matériel virtualisé (mainframes)

Technique de virtualisation

- Problème
 - Le jeu d'instruction du i386 n'est pas virtualisable
- Émulation complète (QEmu)
 - Contrôle total, mais lent
- Réécriture du code au vol (VMWare)
 - Plus rapide mais très complexe
- Paravirtualisation
 - Recompilation pour la plateforme virtuelle, idéal
- Virtualisation matérielle
 - Nouveaux CPUs, reste complexe

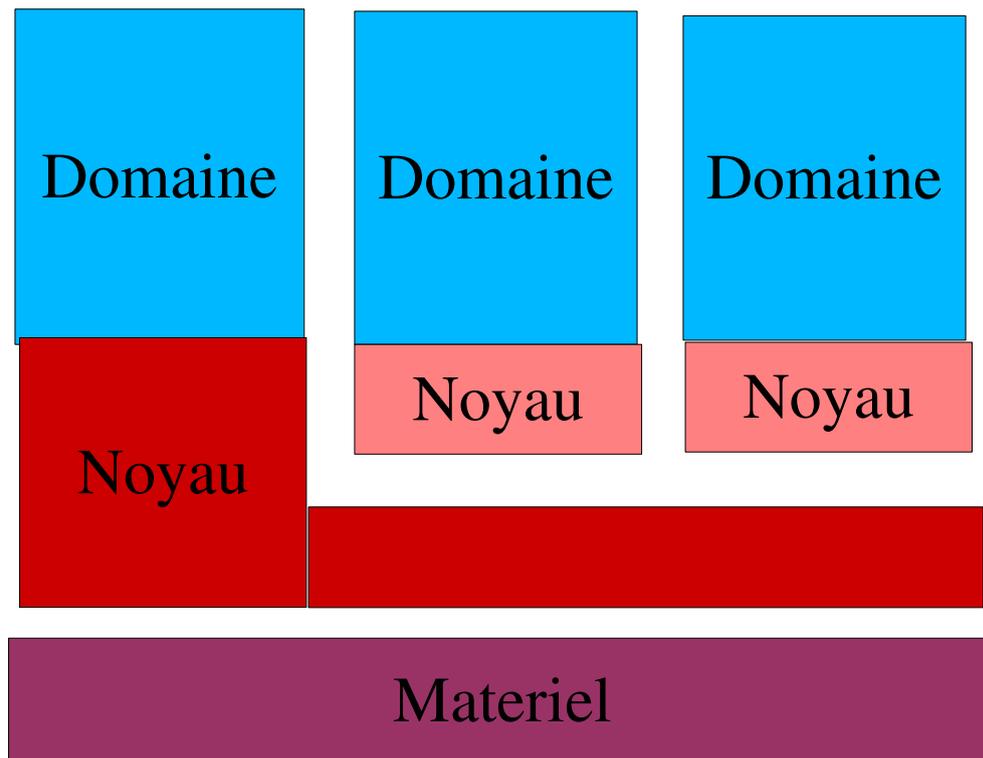
Virtualisation avec hyperviseur

- Hyperviseur isole les noyaux du matériel, ex. Xen



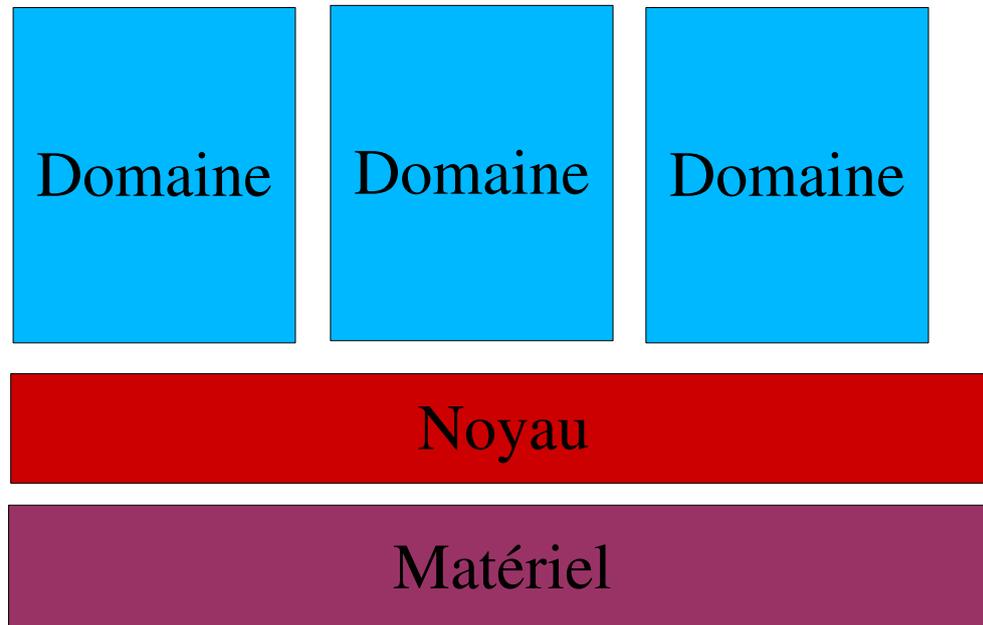
Virtualisation sans hyperviseur dédié

- Le noyau principal est l'hyperviseur, ex. KVM



Virtualisation avec un seul noyau

- Le noyau sert tous les domaines
- Partition des périphériques
- Allocations mémoire/CPU unifiées

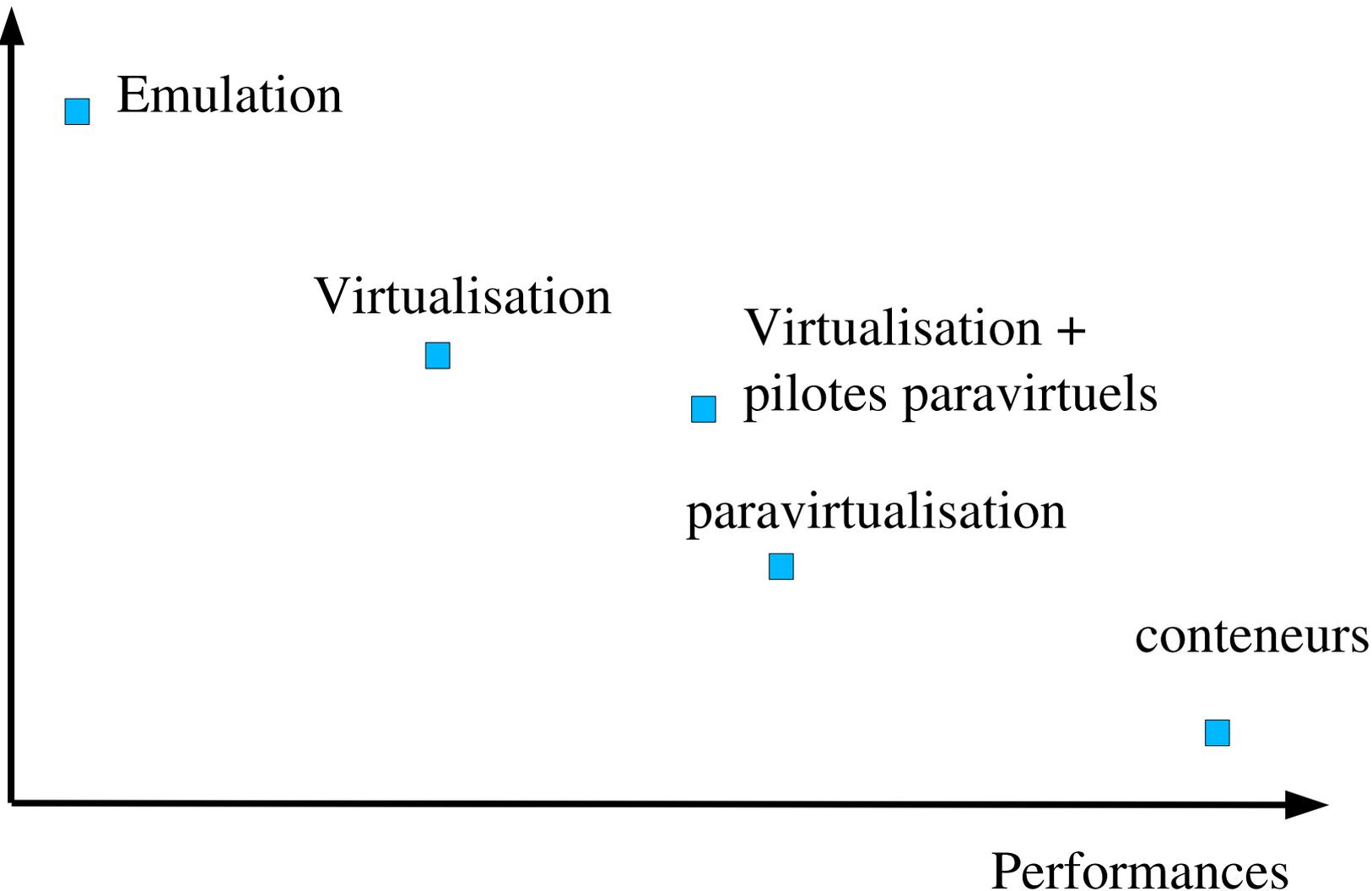


Facteurs determinants

- Capacités d'émulation
- Performances
- Sécurité
- Occupation mémoire, disque
- Fonctionnalités:
 - Tuning
 - drivers
 - Snapshots
 - Migrations a froid/a chaud
- Outils d'administration
- Couts

Compromis flexibilité/performances

Flexibilité



QEmu

- Emulateur, CPU et Materiel
- Nombreuse plateformes emulees
- Auteur principal Fabrice Bellard
- Reutilise par d'autres outils (Xen, KVM)

Xen

- Projet de Recherche, Cambridge UK
- Paravirtualisation de Windows et Linux
- Licence GPL
- Présentation a OLS 2004
- Performances intéressantes
- Virtualisation complète (Qemu + support processeur)
- Intégration partielle dans Linux 2.6.23
- Rachat par Citrix
- Coopération avec Microsoft

Architecture de Xen

Domaine0

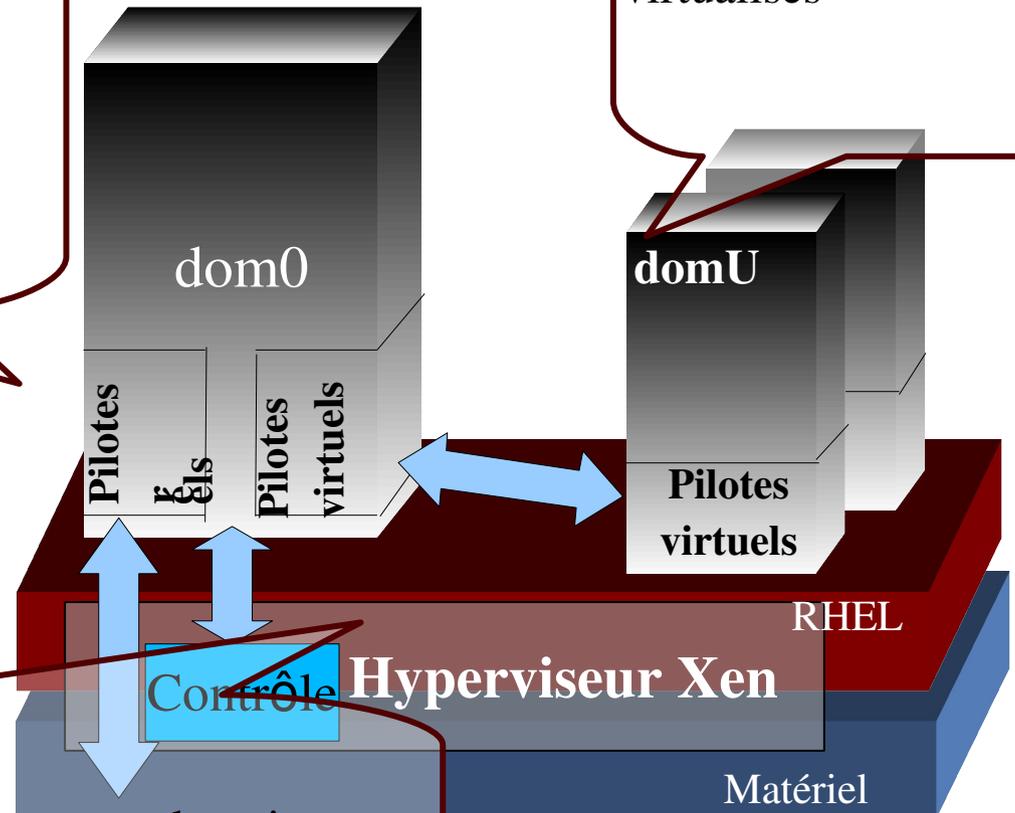
Privilégié, gère les périphériques.
Fournit les interfaces de contrôle
et outils de gestion

Domaine non privilégié:

Machine Virtuelle
CPU, Memoire, Stockage
virtualisés

Hyperviseur Xen

routing IRQ, séquenceur, and communications entre domaines.
Implémente le partage transparent des ressources avec les pilotes réels et limitations d'usage



Xen: nombreuses fonctionnalités

- Cycle de vie classique de la machine virtuelle
- Sauvegarde/restauration de l'image mémoire
- Migration a froid et a chaud
- Redimensionnement memoire et CPUs a chaud
- Pilotes de périphériques complets via Linux
- Bon support i386/x86_64 , ia64
- Robuste, beaucoup de tests
- Isolation correcte mais pas absolue
- Bonne performances en paravirtualisation

KVM: le challenger

- Integre au source de linux depuis 2.6.20
- Virtualisation complète
- Nécessite support matériel (Inter VT or AMD-V)
- Utilise QEMU pour la partie émulation
- Avi Kivity (Qumranet) et de nombreux partenaires
- Fonctionnalités augmentent:
 - Support processeurs S390, ia64, ppc64
 - Migration
 - Drivers paravirtuels linux/Windows
 - Port FreeBSD

UML User Mode Linux

- Inclus dans les noyau 2.6.x
- linux est un programme
- Faible performances
- Tres simple a mettre en oeuvre

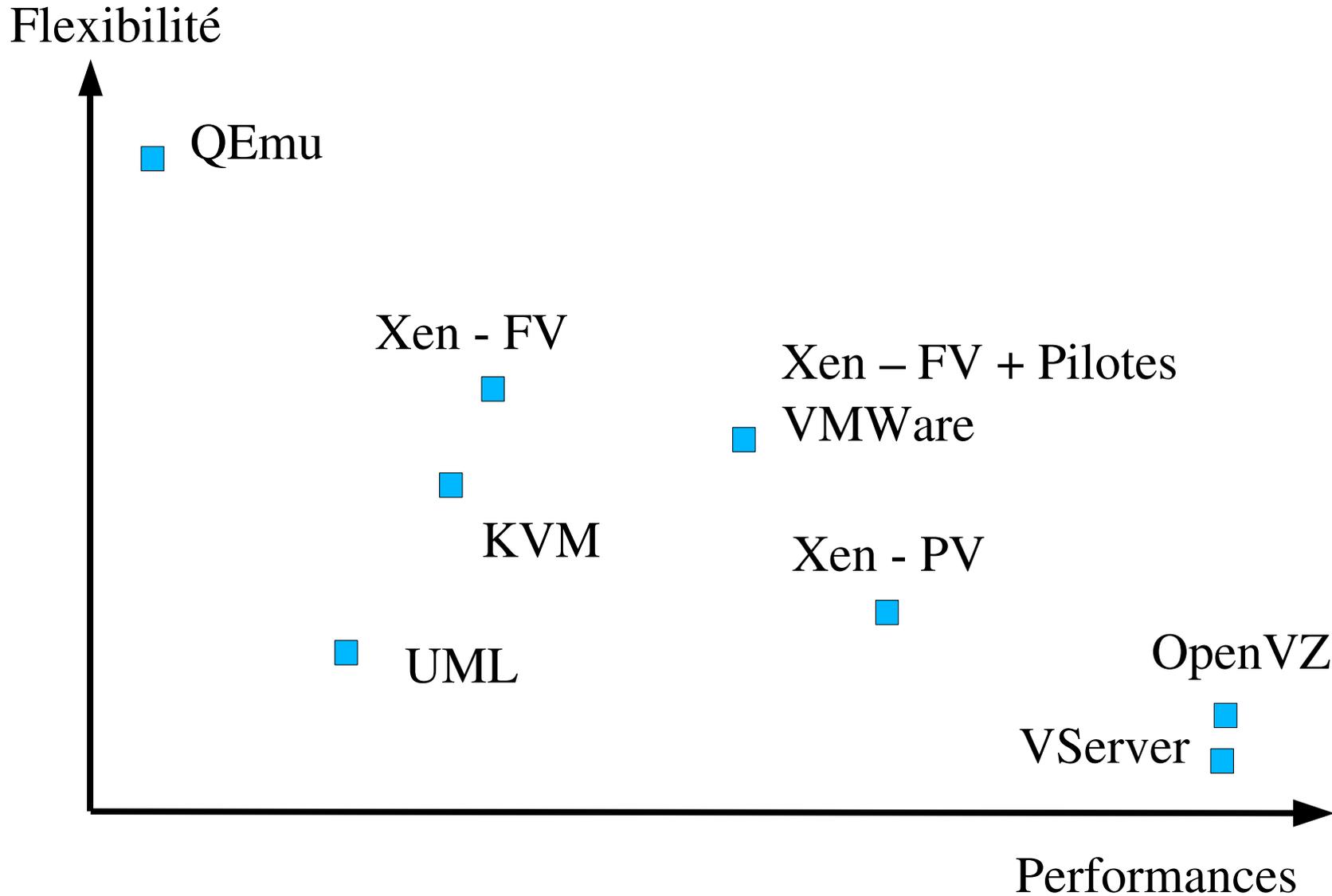
VServer

- Containers pour Linux depuis tres longtemps (patch)
- Partage du systeme
- Interception des appels systemes
- Pas d'emulation, tres rapide
- Ordonnanceur unique
- Un seul noyau
- Modele reseau tres simple

OpenVZ

- Patch pour noyaux recents, modele a container
- Implementation plus complexe que Vserver
- Scheduler a deux niveau
- Virtualisation du reseau
- Tuning tres avance, I/O, reseau et CPU
- Checkpointing, migration a chaud
- Base de Virtuozzo solution proprietaire

Compromis flexibilité/performances



Evolution: integration, uniformisation

- Au niveau du matériel
 - Support pour la virtualisation, partitionnement des CPU
 - Partitionnement, QoS pour les entrées/sorties
- Au niveau du noyau linux
 - Primitives de virtualisation av-ops
 - Intégration de Xen
 - Amélioration de KVM (appel systèmes)
- Au niveau applicatif
 - API d'isolation libvirt
 - Problème de l'affichage, accès GPU
 - Outils systèmes
- 'Appliances' Système + Application prêt a l'emploi

PV-Ops

- Linus 'mettez-vous d'accord !'
- API dans le Noyau pour la virtualisation
 - Virtualisation de l'espace memoire, CPU et I/O
 - Réutilisable par Xen, VMWare, KVM ...
 - Plus de patch noyaux
 - Interface stable
- I386 pv-ops dans le noyau standard
- X86_64 en cours
- Noyau unifie pour Domaine 0 et les noyaux clients

Libvirt

- API stable pour la virtualisation
 - LGPL <http://libvirt.org/>
 - API Generique (Xen, Qemu, KVM, OpenVZ, ...)
 - Relativement portable Solaris, Windows, OS-X...
- Operations de base
 - Créer/détruire/suspendre/sauver/migrer
 - Description XML des machines virtuelles
 - Statistiques, support NUMA
- Ecrit en C
- Bindings python, perl, Java, Ocaml, Ruby



Libvirt et compagnie

- Virsh: shell de contrôle de la virtualisation
- Virt-manager: interface graphique de management
 - Cycle de vie complet
 - Création, migration
 - Console locale et distantes
- Libvirt-CIM
 - implémentation CIM par IBM (LGPL)
 - Pegasus et SFCB
 - Xen et KVM
- Cobbler:
 - Environnement d'installations automatiques
 - Koan agent de réinstallation

Conclusions

- De nombreuses solutions
- Des besoins très divers
- Intégration lente mais en progrès
- Le matériel doit évoluer
- Changement des habitudes de travail



Questions?

veillard@redhat.com | www.redhat.com