



# Virtualisation sous Linux une évolution maîtrisée ?

Daniel Veillard  
veillard@redhat.com

<http://veillard.com/Talks/JRESLinuxVirt.pdf>

TutoJRES 12 Mars 2008

# Virtualisation: une jungle d'acronymes

Xen

QEmu

OpenVZ

UML

VServer

KVM

VMWare

VirtualIron

VirtualBox

Iguest

# Définition et but

- Définition:
  - Découpler les ressources informatique 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

# Bénéfices 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

# Des virtualisations

- Émulateurs:
  - simulation complète de la machine ex. QEmu, Hercules
- Virtualisation complète:
  - Réutilise le processeur natif quand possible
  - OS client non modifié, requiers 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)

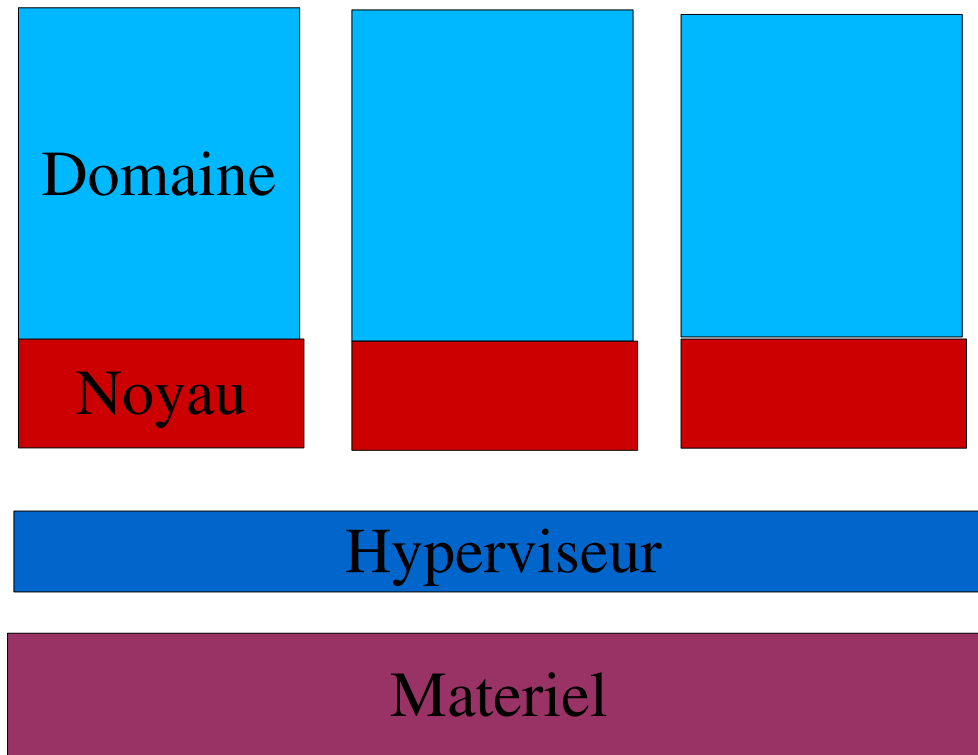
# Techniques de virtualisation

- 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

En pratique, plusieurs méthodes sont mixées

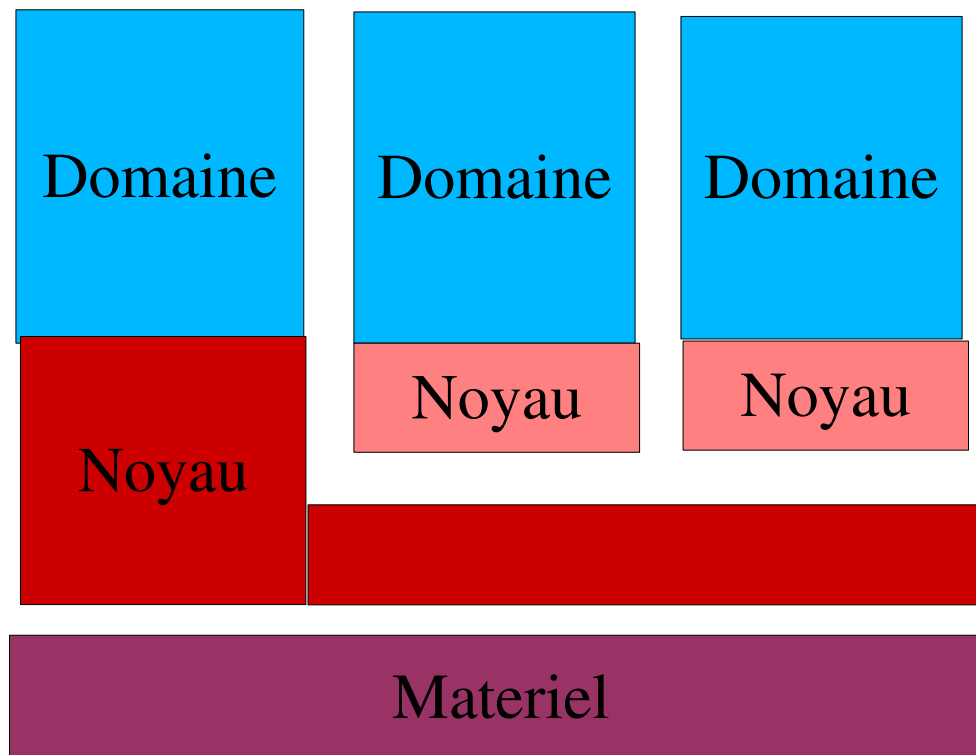
# Virtualisation avec hyperviseur

- Hyperviseur isole les noyaux du matériel, ex. Xen



# Virtualisation sans hyperviseur dédié

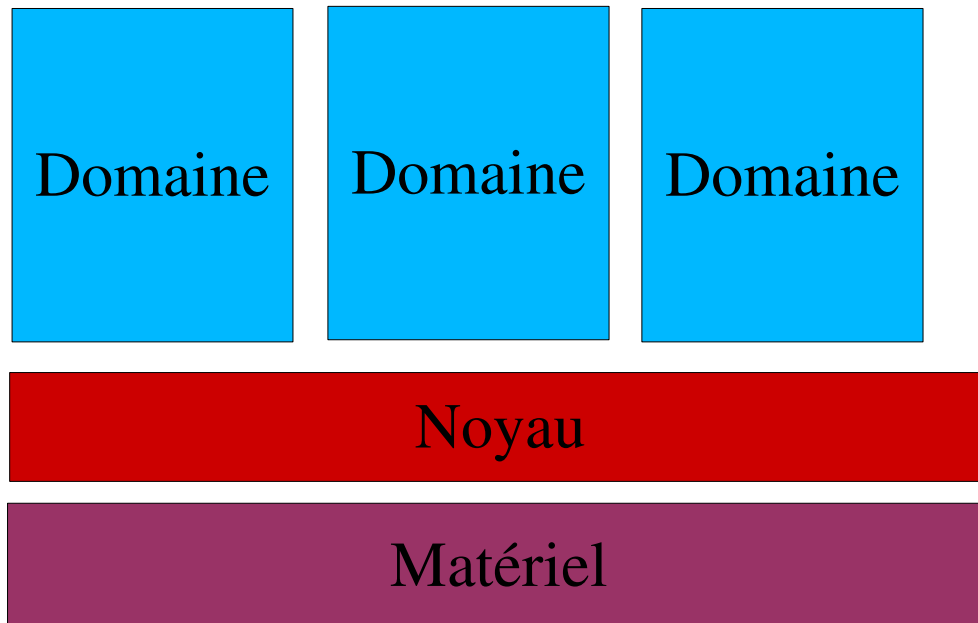
- Le noyau principal est l'hyperviseur, ex. KVM





# Virtualisation avec un seul noyau

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

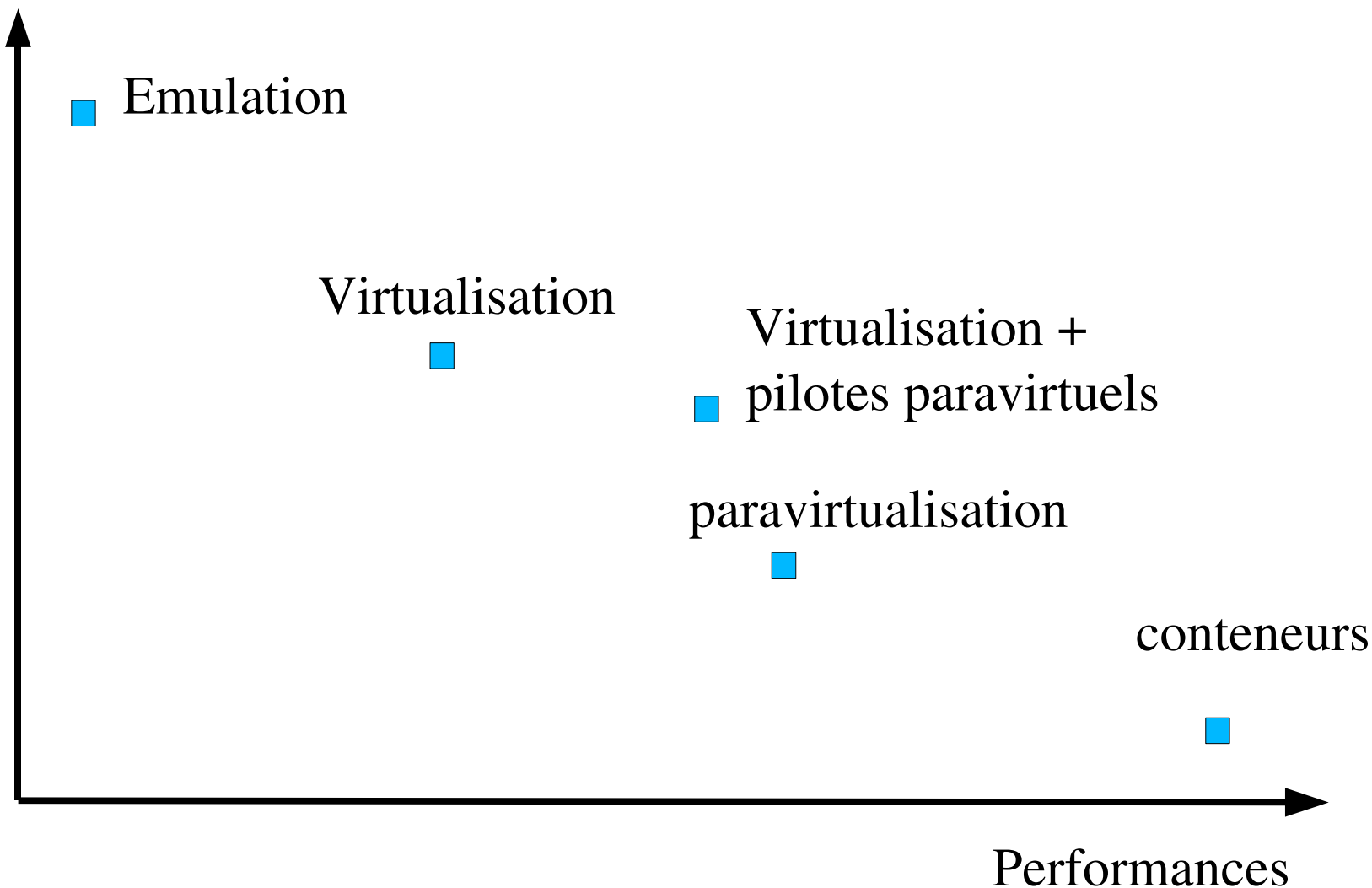


# Facteurs déterminants

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

# Compromis flexibilité/performances

Flexibilité



# QEmu

- Emulateur, CPU et matériel
- Nombreuse plateformes émulées
- Auteur principal Fabrice Bellard
- Réutilisé 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

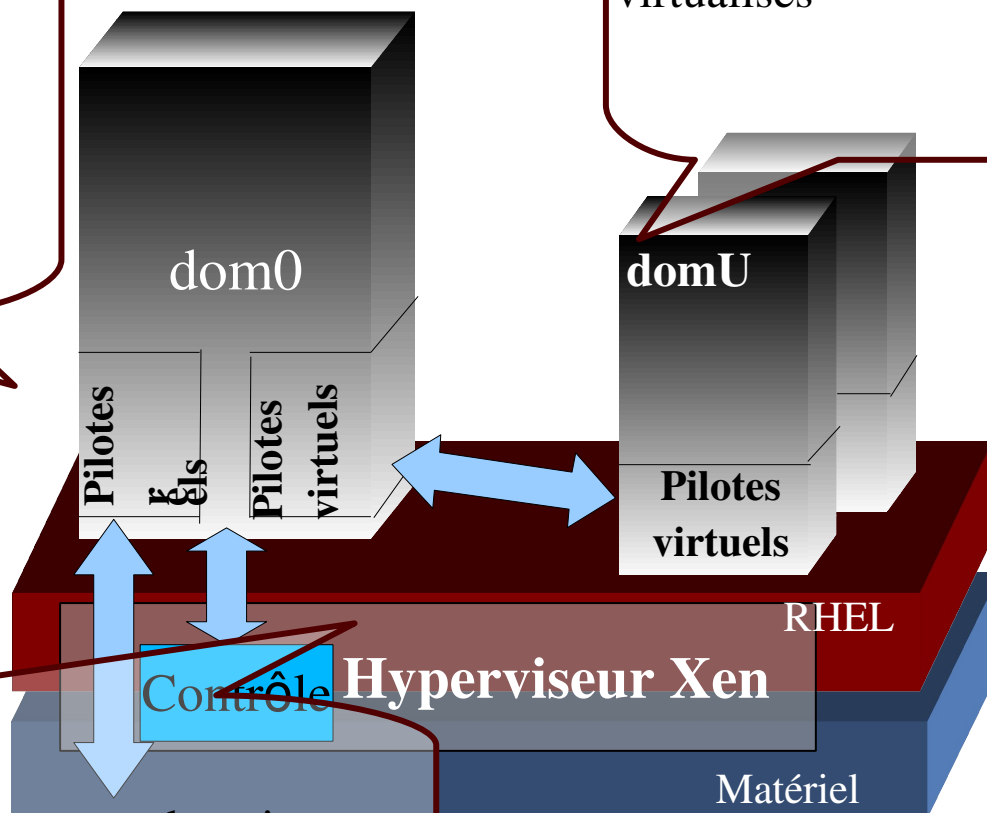
Privilégié, gère les périphériques.  
Fournit les interfaces de contrôle  
et outils de gestion

## 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

- Intégré 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
- Très simple a mettre en oeuvre

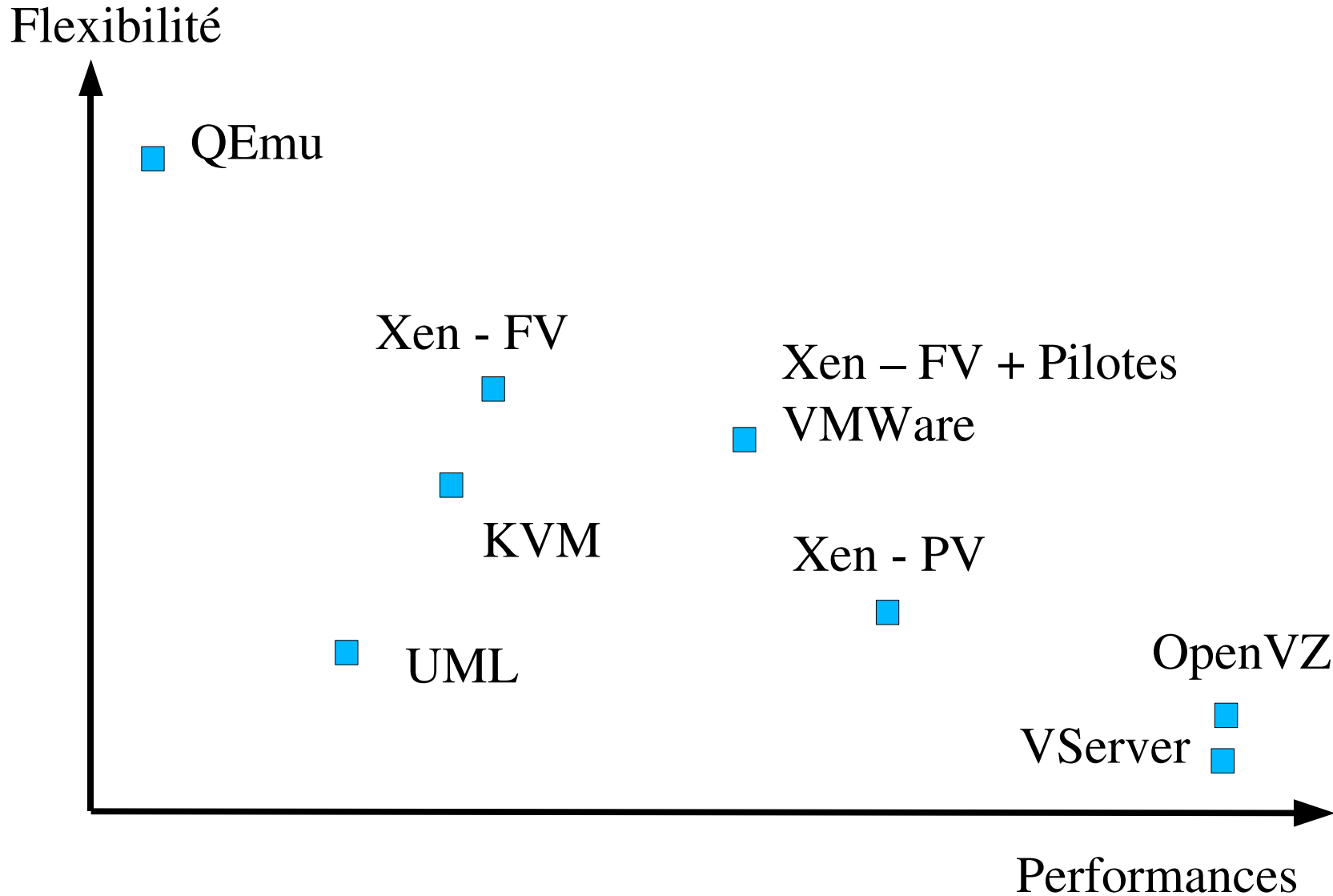
# VServer

- Containers pour Linux depuis très longtemps (patch)
- Partage du système
- Interception des appels systèmes
- Pas d'émulation, très rapide
- Ordonnanceur unique
- Un seul noyau
- Modèle réseau très simple

# OpenVZ

- Patch pour noyaux récents, modèle a container
- Implémentation plus complexe que Vserver
- Ordonnanceur a deux niveau
- Virtualisation du réseau
- Tuning très avance, I/O, réseau et CPU
- Checkpointing, migration a chaud
- Base de Virtuozzo solution propriétaire

# Compromis flexibilité/performances



# Evolution: intégration, 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 paravirt-ops (pvops)
  - Intégration progressive de Xen
  - Amélioration de KVM
- Au niveau applicatif
  - API d'isolation libvirt
  - Problème de l'affichage, accès GPU
  - Outils systèmes
- 'Appliances' Système + Application prêt a l'emploi

# Paravirt-ops

- VMI patch from VMWare, Xen from XenSource
- Linus 'mettez-vous d'accord !'
- API dans le noyau pour la virtualisation
  - Virtualisation de l'espace mémoire, CPU et I/O
  - Réutilisable par Xen, VMWare, KVM ...
  - Fin des patchs noyaux
  - Interface stable
- I386 pv-ops DomU dans le noyau standard
- X86\_64 en cours (prévu pour 2.6.25)
- Noyau unifié pour Domaine 0 et les noyaux clients (futur)
- Travail en cours pour Fedora 9 et 10

<http://fedoraproject.org/wiki/Features/XenPvops>

# Libvirt



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

# Libvirt et compagnie

- Virsh: script 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



# Libvirt et compagnie (suite)

- Virt-p2v
  - Outil de migration automatique
  - CD Boot, accès distant
  - <http://et.redhat.com/~rjones/virt-p2v/>
- Ovirt
  - Interface de gestion d'un ensemble de serveurs
  - Intègre le stockage, kerberos, DHCP, PXE
  - Développement en cours
  - <http://ovirt.org/>

# Conclusions

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



**Questions?**

**veillard@redhat.com | www.redhat.com**

**<http://veillard.com/Talks/JRESLinuxVirt.pdf>**