Introduction à la virtualisation

Jean-Patrick Gelas
UE Cloud Computing – M2
Université Claude Bernard – Lyon 1

Introduction

Virtualiser : proposer, par l'intermédiaire d'une couche d'abstraction proche du matériel, une vue multiple d'un matériel unique, en sérialisant les appels vus concurrents de l'extérieur.

Analogie avec les processeurs

- Cadence max atteinte (~3.6GHz)
- Naissance du multi-cœurs suivit de l'
- Hyper-threading (1 cœur = 2 cœurs logiques)
  (Ex: Core i7 => 1 processeur, 4 cœurs physiques hyper-threadés => soit 2x4=8 cœurs logiques).

- La virtualisation consiste à « augmenter » le nombre de cœurs.

Terminologie

- Le **système hôte (*host*)** est l'OS principal de l'ordinateur.
- Le **système invité (*guest*)** est l'OS installé à l'intérieur d'une machine virtuelle.
- Une **machine virtuelle (*VM*)** est un ordinateur virtuel qui utilise un système invité.
- Un ordinateur virtuel est aussi appelé **serveur privé virtuel** (*Virtual Private Server* ou VPS) ou environnement virtuel (*Virtual Environment* ou VE)

Intérêts

- Usage optimale des ressources
- Installation, déploiement et migration facile
- Économie sur le matériel
- Sécurisation
- Isolation
- Allocation dynamique
- Diminution des risques

Historique

- Idée développée au centre IBM de Cambridge et de Grenoble en 1972 (VM/CMS) (pseudo-machine.)
- Mi-90’s émulateurs d’Atari, Amiga,NES,SNES,...
- Début des années 2000 : VMware
- Logiciels libre : Xen, Qemu, Bochs,...
- Propriétaire (mais gratuits) : VirtualPC,...

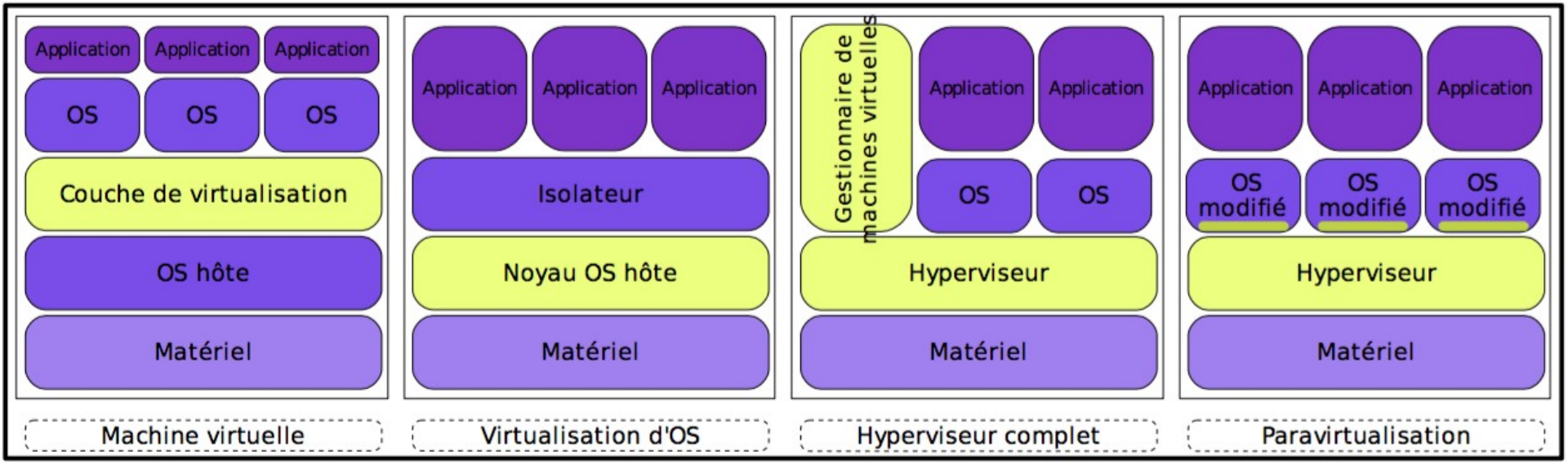
Différents domaines

- Virtualisation d’applications (du contexte d’exécution).
- Virtualisation de serveur
- Virtualisation du réseau (VLAN)*
- Virtualisation du stockage*

(*: non traité dans ce cours)

Différentes techniques de virtualisation

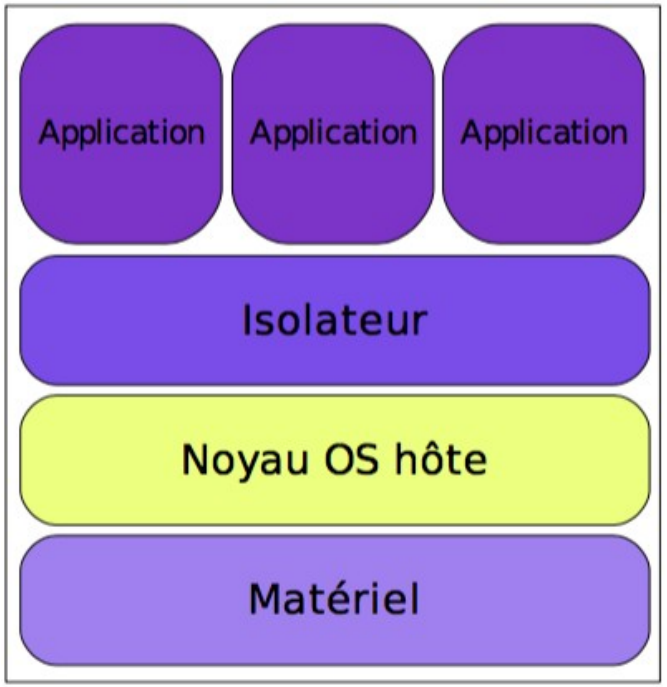
- Usage de couches logicielles intermédiaires
- 4 types de technologies



Techno #1 : Virtualisation d'OS ou Isolateur

- Isole l'exécution des applications dans des contextes d'exécution.
- Généralisation de la notion de « contexte » Unix, plus isolation
  - des périphériques,
  - des systèmes de fichiers
- Solution très performante et économique en mémoire mais
- Partage du code noyau (donc mauvaise isolation).

Exemple : chroot (changement de racine), Linux Vserver, OpenVZ (Virtuozzo), Docker, LXC (Cgroups),...

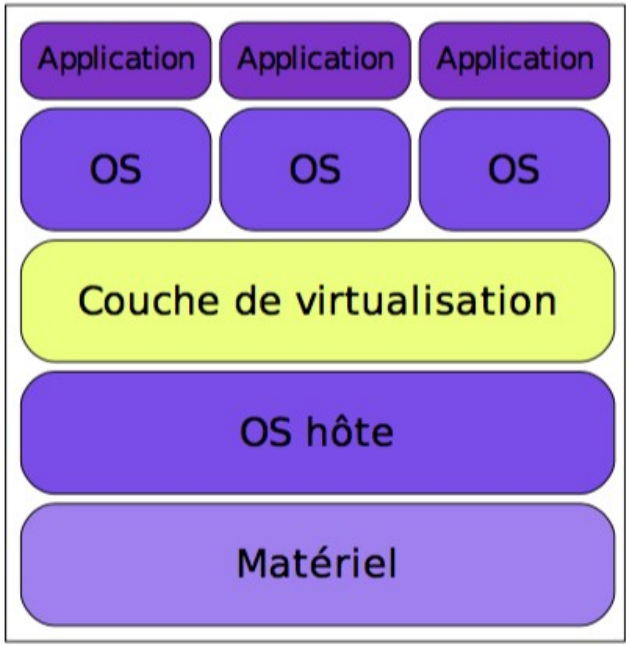


Techno #2 :

Hyperviseur de type 2

(ou Architecture hébergée)

- Application installée sur l'OS
- Virtualise et/ou émule le matériel
- Comparable à un émulateur mais accès « direct » au CPU, RAM, FS.
- Performances réduites si le CPU doit être émulé
- Bonne étanchéité entre les OS invités.

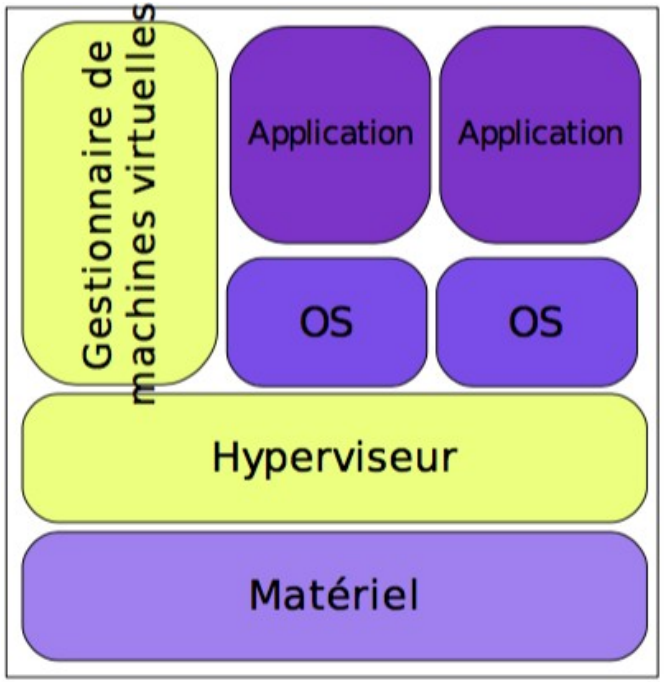


Exemples : VirtualBox, QEMU, Vmware (workstation, fusion,player), Microsoft Virtual PC, Parallels desktop,...

Techno #3 :
Hyperviseur complet
(dit de type-1 ou bare-metal)

- Noyau système léger et optimisé
- Outils de supervision
- Permet l'exécution d'OS natifs
- Usage d'instructions dédiées à la virtualisation (sinon émulation).

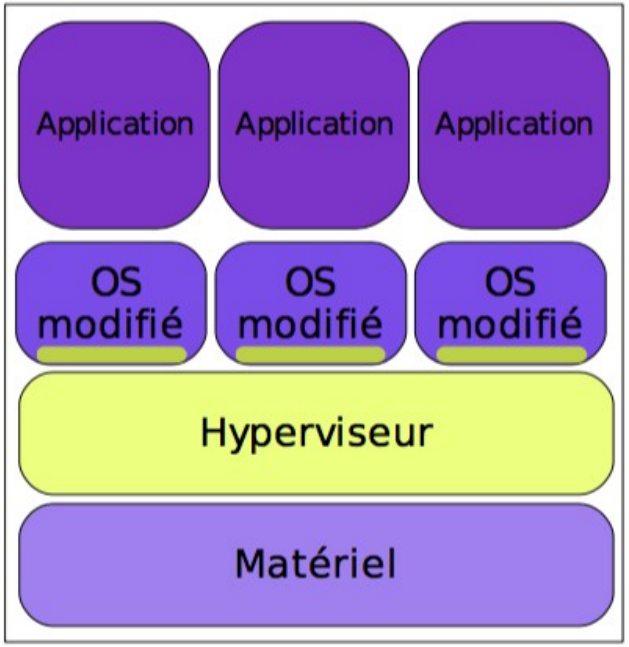
- Ex: XEN, KVM, Vmware vSphere,…



Techno #4 : Paravirtualisation

(Hyperviseur de type 1 également)

- Noyau système allégé et optimisé
- Noyau invités adaptés et optimisés
- Utilisable sans les instructions spécifiques (ex : VT-x ou AMD-v).
- Impraticables pour les systèmes non libres.



- Exemples : Vmware Vsphere, XEN, Microsoft Hyper-V server, KVM,...

Pour info : Une 5ème technologie…

Noyau dans l’espace utilisateur

- Un noyau exécuté comme une application dans le *user-space*.
- Très peu performant (empilement de deux noyaux !)
- Utile au développement noyau.

Ex : UML (User Mode Linux)
http://user-mode-linux.sourceforge.net/

Matériel

- Le support de la virtualisation peut être intégré au processeur
  - Virtualisation des accès mémoire
  - Protection du processeur physique des accès bas niveaux
- Simplifie la virtualisation logicielle et
- réduit la dégradation de performance

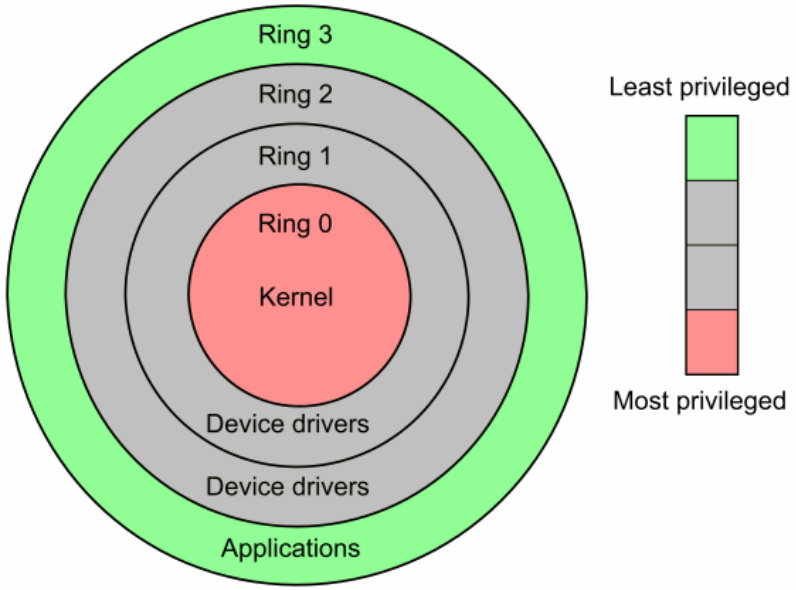
Ex : AMD-v et Intel VT

AMD-V et Intel VT

- Jeu étendu d’instructions de virtualisation.
- Un « super Bios » fait l’interface avec la puce.
- Simplifie
  - la virtualisation logicielle
  - Réduit la dégradation de performances

Solution x86

- 4 niveaux de privilèges mal exploité
  - Système : au niveau le plus privilégié (*ring 0*)
  - Applications : au niveau le plus faible (*ring 3*)
- Ajouts d'instruction dédiés par AMD et Intel pour une virtualisation matérielle (~2007)
  - Coexistence de plusieurs *ring 0* simultanée
  - On parle de *ring -1*



Bit NX/XD

- NX (Non eXecutable) ou XD (eXecute Disable)
- Bit spécial qui permet de marquer des zones mémoires comme non exécutable.
- Améliore l’isolation des VM.

LAHF/SAHF

Load AH From flags/Save AH from Flags

- Autorisent un contrôle direct du registre AH.
- Un hyperviseur utilise ces instructions pour assurer un contrôle plus direct du traitement des E/S et IRQ de chaque cœur de processeur.

EPT (Extended Page Table)
RVI (AMD) / SLAT (Intel)

- Dans une VM la traduction d’adresses est faite deux fois !!
- La traduction d’adresses est nécessaire car les processeurs doivent utiliser une table de pages ou un tampon de traduction (TLB) pour convertir les adresses relatives en adresse physique.

Cgroups

- Fonctionnalité du noyau Linux pour limiter, compter et isoler l'utilisation des ressources (processeur, mémoire, disque, etc.).
  - Limitation des ressources
  - Priorisation
  - Comptabilité
  - Isolation
  - Contrôle
- Isolation par espace de nommage

Quelques solutions
(Open source)

- QEMU
- KVM
- XEN
- VirtualBox

- OpenVZ
- Docker

Qemu

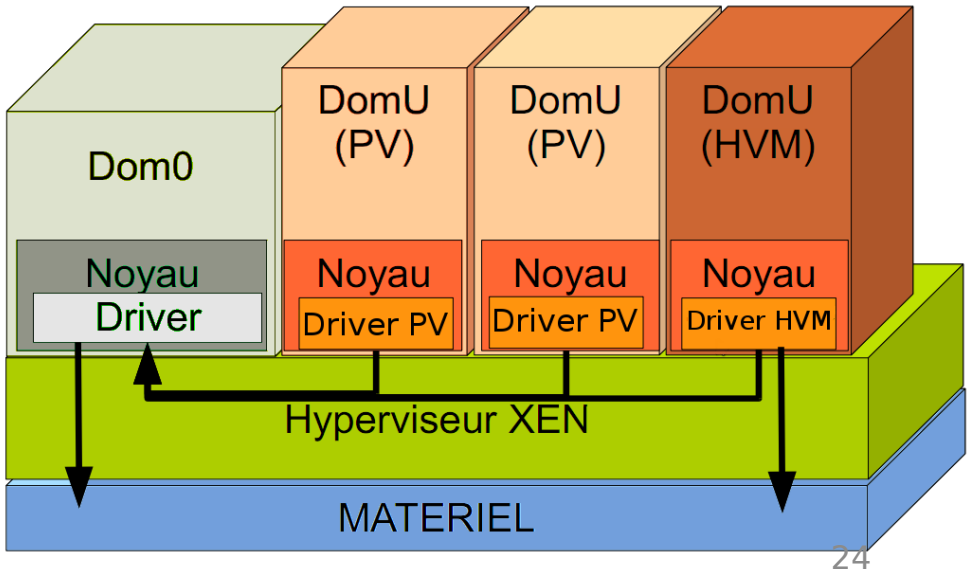
- Machine virtuelle complète
- Techniquement très aboutie
- Émulation complète de machine (x86,ARM,MIPS,…)
- L’usage du module *kQemu* pour une virtualisation accélérée.
- Émulation par recompilation sur un modèle « just-in-time »
- Gourmand en mémoire
- Sans accéleration lent et charge l’hôte.

Ex : VirtualBox et KVM reposent sur Qemu

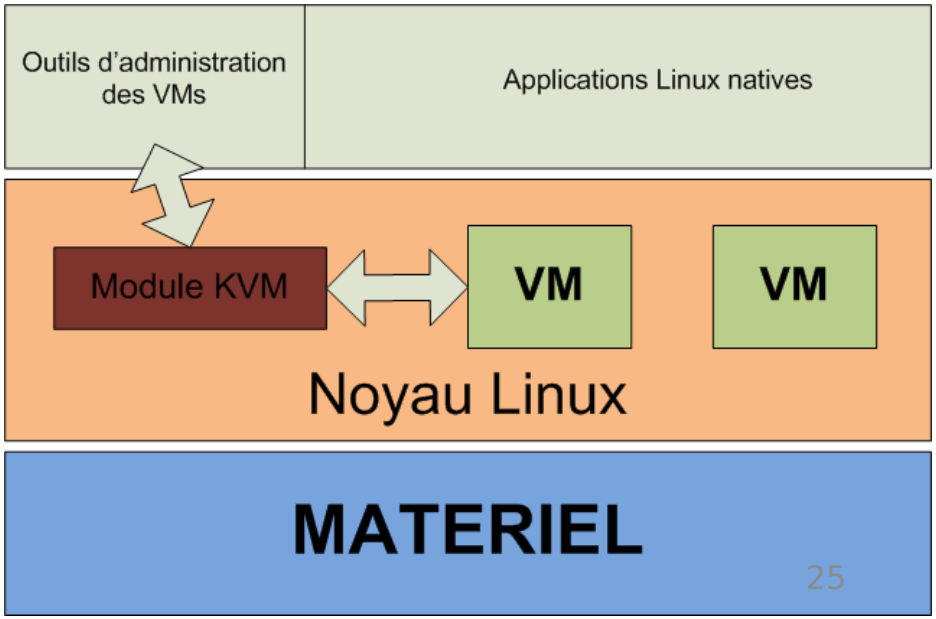
VirtualBox

- Machine virtuelle, émule un PC complet
- Support des instructions de virtualisation
- Solution de virtualisation efficace
- Repose sur Qemu (au boot au moins)
- Gourmand en mémoire
- Simple à utiliser

- Solution libre ancienne mais encore utilisée.
- Vocabulaire :
  - OS privilégié : Dom0
  - OS invités : DomU
    - DomU standard (paravirt.)
    - DomU HVM (hardware assisted)
- Deux modes d'usage :
  - Paravirtualisation :
    - Noyau speçifique dans le Doml
    - Très bonnes performances
  - Virtualisation matérielle
    - Virtualisation transparente pour le système invité.
    - Besoin d'un support dans le processeur (AMD-V ou Intel VT)



- Projet plus récent que Xen mais très populaire.
- Basé en partie sur QEMU (pour le supports des périphériques)
- Entièrement intégré au noyau Linux -> Facile à utiliser.
- Support de la virtualisation dans les processeurs indispensable.
- Paravirtualisation (virtio) pour les performances.



OpenVZ

- PVS (Private Virtual Server)
- Virtualisation de niveau OS basée sur Linux
  - Un Linux avec plusieurs tables de processus
  - Chacune son contexte

Virtualisation : Les inconvénients

- Un point de défaillance unique
- Un recours à des machines puissantes
- Une dégradation des performances
- Une complexité accrue de l’analyse d’erreurs
- Parfois inadapté (Ex : I/O intense).

Conclusions

- Afin d'avoir une idée théorique des performances des applications au sommet, il faut comparer verticalement l'empilement de couches.
- Ne pas virtualiser un serveur déjà beaucoup sollicité (la virtualisation rajouterai de *l'overhead* inutile).
- Réduit les coûts, facilite l'administration mais
- Il faut être capable de gérer un grand nombre de serveurs.
- Deux grandes familles de virtualisation :
  - Containers/Isolateurs
  - Hyperviseurs

- Concept indispensable et étroitement lié à la réussite du Cloud !

Sources

- Livre blanc sur la Virtualisation, Groupe Linagora
- https://storify.com/selossej/la-virtualisation
- https://fr.wikipedia.org/wiki/Virtualisation