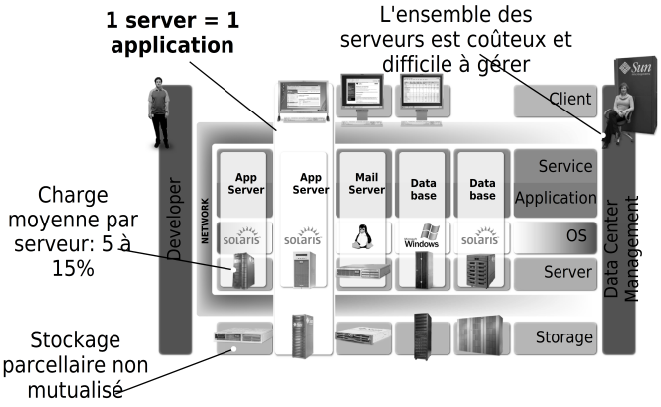
Virtualisation: définitions

- Ensemble techniques logicielles et matérielles permettant de fournir un ensemble de ressources informatiques utilisable indépendamment de la plate forme matériel
- Domaines concernés :
  - Ressources CPU et mémoire
  - Stockage
- Surcouche permettant de s'abstraire des contraintes matérielles
- Exemple: les gestionnaires de volumes logiques vs les disques physiques

Problème des datacenters actuels

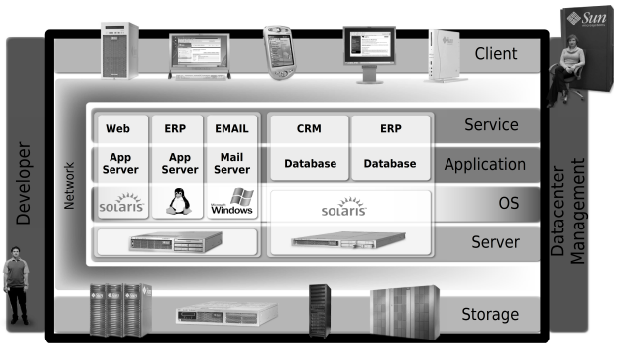
- Serveurs sous-utilisés (en moyenne 5 à 15%)
- Sécurité: 1 service = 1 serveur
  - Implique une multiplication des serveurs physiques
- Coût d'achat
- Coût d'exploitation
  - Énergie: alimentation (#80€/an/serveur), climatisation (> alimentation)
  - Gestion, place occupée,...
- Facilitée de gestion
- Tolérance de panne
- Plan de reprise d'activité difficile et coûteux

Le Data Center d'hier



Source: Sun microsystems

Le DATA Center aujourd'hui



Source schéma: Sun microsystems

Apport de la virtualisation

- Coût
  - Acquisition
  - Exploitation
  - Mais : attention au côut des licences des OS, ...
- Facilité de gestion
  - Sauvegarde des machines virtuelles
  - Déplacement de machines virtuelles
- Sécurité
  - Isolation des machines virtuelles
    - 1 machine virtuelle = 1 service

Historique de la virtualisation

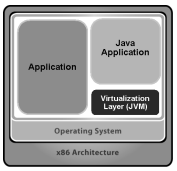
- 1965 IBM M44/44X **paging system**
- 1965 IBM System/360-67 **virtual memory hardware**
- 1967 IBM CP-40 (January) and CP-67 (April) **time-sharing**
- **1972 IBM VM/370 run VM under VM**
- **1997 Connectix First version of Virtual PC**
- **1998 VMWare U.S. Patent 6,397,242**
- **1999 VMware Virtual Platform for the Intel IA-32 architecture**
- **2000 IBM z/VM**
- **2001 Connectix Virtual PC for Windows**
- **2003 Microsoft acquired Connectix**
- **2003 EMC acquired Vmware**
- **2003 VERITAS acquired Ejascent**
- **2005 HP Integrity Virtual Machines**
- 2005 Intel VT
- 2006 AMD VT
- 2005 XEN
- 2006 VMWare Server
- 2006 Virtual PC 2006
- 2006 HP IVM Version 2.0
- 2006 Virtual Iron 3.1
- 2007 InnoTek VirtualBox
- 2007 KVM in Linux Kernel
- 2007 XEN in Linux Kernel
- 2008 microsoft Hyper V

I - Les différentes techniques

- Virtualisation applicative
- Redondance
- Virtualisation du stockage
- Réimplémentation de librairies
- Isolateur
- Noyau en espace utilisateur
- Superviseur en mode natif
- Superviseur en mode émulé
- Hyperviseur (para-virtualisation)
- Virtualisation matérielle
- Virtualisation du poste de travail

Virtualisation applicative

- Au niveau de chaque application, au sens large
- Exemples:
  - Web: Virtualhosts Apache
  - IP: Linux Virtual Server (ipvsadm)
  - Java (hors sujet)
  - Stockage: hal, lvm, xvm



High-Level Language

Serveur WeB : apache

- Une même machine
  - héberge plusieurs sites WeB
  - Se comporte comme si on avait plusieurs serveurs
- Les différents sites sont distingués :
  - Par leur nom
    - Entrée dns nom -> serveur
  - Par leur adresses ip
    - Machine avec plusieurs cartes réseau ou avec plusieurs adresses ip par carte
  - Dans les deux cas: config d'apache aussi

Redondance et tolérance de panne : exemple des routeurs

- Fait en live au tableau
- HSRP/VRRP/CARP
- Mac et IP virtuelles
- Protocole Hello
- Problème des états du routeurs HS (NAT, FW)

Virtualisation du stockage

- Gestionnaire de volumes logiques
  - Ajoute une couche qui permet de ne plus voir les périphériques physique
  - Volume logique:
    - Composer de plusieurs volumes physiques provenant de sources (disques) variés
    - Extensibles, support du RAID, ...

Virtualisation du stockage

- NAS/SAN
  - NAS:
    - Connecté au réseau
    - Propose un accès via des protocoles de partages réseau (CIFS, NFS, AFS, ...)
    - Accès en mode fichiers
  - SAN
    - Les baies apparaissent comme un disque local
    - Accès en mode bloc

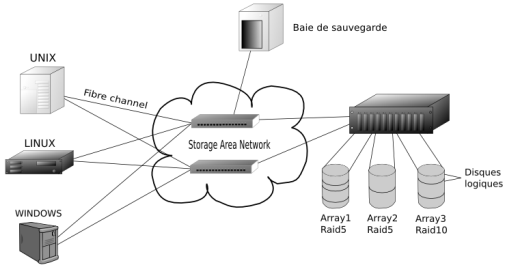
Virtualisation du stockage: SAN

- SAN:
  - Gestion globale et mutualisée de l'espace disque pour l'ensemble des serveurs: souplesse
  - Ajout facile d'espace à un serveur donné
  - Ajout facile d'espace sur le SAN (ajout de disques, de baies, ...)
  - Sauvegarde, tolérance de panne facilitées (raid, redondance de la liaison avec le SAN)
  - Réplication vers une autre baie (éventuellement distante), historisation, snapshot, ...
- Coût élevé: en général 6 chiffres en EUR

Virtualisation du stockage: SAN

- Protocoles des SAN : Lien serveurs-SAN
  - Fiber channel : rapide mais cher
  - Iscsi: scsi sur TCP/IP, supporté par les OS tant en client qu'en serveur
    - RFC 3720 - Internet Small Computer Systems Interface (iSCSI), Avril 2004.
    - RFC 3721 - Internet Small Computer Systems Interface (iSCSI) Naming and Discovery, Avril 2004.
    - RFC 3722 - String Profile for Internet Small Computer Systems Interface (iSCSI) Names, Avril 2004.
  - AoE: ATA sur ethernet (peu utilisé)

Virtualisation du stockage



Virtualisation du stockage: choix de la taille des disques

- Paramètres :
  - Espace disque
  - Débit du disques
  - Nombre d'I/O par seconde
- Dans la vraie vie :
  - SATA:
    - De l'espace pour pas cher
    - Vitesse de rotation modeste
    - Temps d'accès élevé (8ms)
    - Nb d'I/O par seconded modeste
      - File d'attente inexistante ou limitée sur les disques
      - Temps d'accès èlevé
    - Lien SATA unique

Virtualisation du stockage: choix de la taille des disques

- SAS :
  - Coût au Go èlevé (mais on commence à trouver des disques abordables)
  - Vitesse de rotation èlevée (15Krpm)
  - Temps d'accès faible (4 ms)
  - Nb d'I/O par seconde èlevé
    - File d'attente de taille correcte
    - Temps d'accès faible
  - Lien SAS bicanal

Virtualisation du stockage: choix de la taille des disques

- SSD:
  - Capacité modeste
  - Coût au Go très èlevé
  - Temps d'accès très très faible (0,1 ms) sur certains modèles
  - Durée de vie ?
- Choix de la taille des disques
  - Déterminer le facteur bloquant (I/O/s, capacité,)
  - Prendre des disques de taille modeste pour augmenter le nombre d'O/I/s de l'ensemble
  - Raid 0, raid5,6, 10, 01, … permettent de combiner des disques pour « additionner » les débits et/ou les nombred'I/O par seconde des disques

choix de la taille des disques: exemples concrets d'architecture

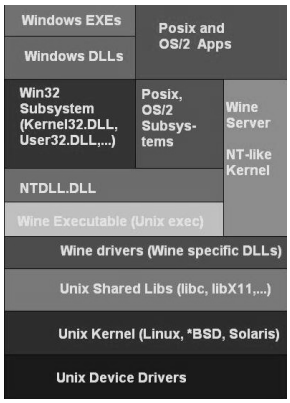
- Serveur de sauvegarde
- Serveur de boîtes aux lettres
- Serveur de courrier entrant

Réimplémentation de librairies

- Réécriture des librairies d'un OS pour faire fonctionner des programmes sous un autre OS
- Exemple: Wine (utiliser des applications windows sous unix)

Architecture de Wine

- Suit l'architecture de NT
  - Implèmente toutes les "core" DLLs (ntdll, user32, kernel32)
- Le "Wine server" fournit l'infrastructure NT :
  - Transmission de messages
  - Synchronization
  - Handles des objects



Isolateur

- Cloisonnement d'environnements (ou contextes) au sein d'un OS
- Une seule instance du noyau de l'OS commune à tous les environnements
- Exemples (de technologies différentes) :
  - Chroot: changement de racine (faible)
  - BSD Jail: isolation en espace utilisateur
  - Linux-Vserver, OpenVZ, Solaris Zones: partitionnement au niveau du noyau de l'OS
- Très performant mais le cloisonnement est-il suffisamment solide ?

Noyau en espace utilisateur

- Le noyau de l'OS de la machine virtuelle est lancé comme n'importe quelle application de l'OS hôte
- Exemples:
  - User Mode Linux (UML)
  - coLinux
  - Adeos
  - L4Linux
- Faibles performances (on empile 2 noyaux)
- Machines virtuelles issues du même noyau

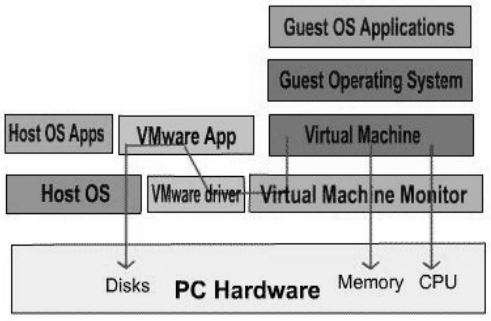
Superviseur en mode natif

- Appellé aussi "**machine virtuelle**, en mode natif". Logiciel complexe qui permet l'exécution de plusieurs OS invités de même architecture processeur que la machine hôte en offrant un accès générique (émulé) aux ressources physiques.
- Exécution des instructions en mode natif (sauf exceptions difficiles à gérer)
- Bonnes performances
- OS invités différents possibles

Superviseur en mode natif

- Exemples:
  - Vmware Player, workstation, fusion
  - Sun Virtual box
  - Vmware Server
  - Microsoft Virtual PC

Vmware workstation

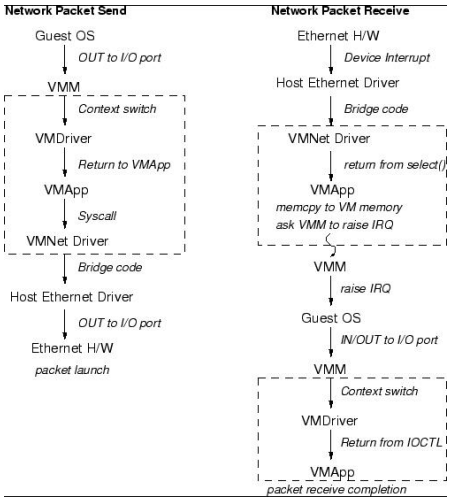


VMware Workstation Architecture

Vmware workstation

- VMM does not have access to I/O
- I/O in “host world”
  - Low level I/O instructions (issued by guest OS) are merged to high-level I/O system calls
  - VM Application executes I/O SysCalls
- VM Driver works as the communication link between VMM and VM Application
- World switch needs to “save” and “restore” machine state
- Additional techniques to increase efficiency

Vmware workstation



Processeurs x86 et virtualisation

- 2 modes de fonctionnement sur un processeur X86:
  - Mode réel: utilisé au démarrage, compatibilité 8086
  - Mode protégé:
    - Utilisé par tous les OS modernes
    - Ajoute des mécanismes de protection mémoire à l'architecture
- Mode protégé:
  - 4 niveaux de privilèges pour assurer la protection :
    - Niveau 0 : mode noyau, le plus privilégié
    - Niveau 1 et 2: peu ou pas utilisés
    - Niveau 3: applications

Processeurs x86 et virtualisation

- Niveau 0: on peut exécuter toutes les instructions dont la gestion mémoire, interruptions, changement d'état, ...): réservé au noyau de l'OS
- Niveau 3: certaines instructions sont interdites pour assurer la fiabilité du système vis à vis des utilisateurs
  - Pour accèder à une ressource de niveau 0, il faut passer par un point de contrôle (syscall)

Processeurs x86 et virtualisation

- L'hyperviseur s'exécute en niveau 0, Les OS invités à un autre niveau
- Le code exécutable de l'OS invité se décompose en 3 catégories d'instructions :
  - Le code invité non privilégié et non dangereux: s'exécute directement par le processeur pour des raisons de performances
  - Le code invité en mode privilégié
  - Des instructions sensibles, non privilégiées mais incompatibles avec un environnement de virtualisation.
    - 17 instructions qui rendent les processeurs X86 non virtualisables
  - Cf http://www.ecsl.cs.sunysb.edu/~susanta/slides/virt.ppt pour plus de détails.

Processeurs x86 et virtualisation

- 2 stratégies possibles :
  - Supprimer les instructions privilégiées et celles posant problèmes
    - Il faut réécrire une partie de l'OS invité
    - Solution utilisée par les versions 1 et 2 de Xen
  - Intercepter les instructions sensibles pour les faire exécuter sous le contrôle de l'hyperviseur ou les réécrire au vol
    - Facile pour les instructions privilégiées (via des exceptions)
    - Délicat pour les autres: plus value des solutions de virtualisation comme vmware

Processeurs x86 et virtualisation: support matériel de la virtualisation

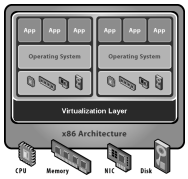
- Le processeur fournit des fonctionnalités pour faciliter la prise en charge de la virtualisation
- Intel VT ou AMD-V (incompatibles :-))
- 2 impacts :
  - Améliorer les performances
  - Simplifie la mise en oeuvre des solutions de virtualisation
    - => il n'y a plus de valeur ajoutée à proposer seulement de la virtualisation

Superviseur en mode émulé

- Appellé aussi “**machine virtuelle**, en mode émulé” ou encore “**émulateur**”
  A la différence de la machine virtuelle en mode natif, le processeur est aussi émulé randant ainsi possible l'éxécution d'OS pour des plateformes différentes de celle de l'hôte
- Pb: performances modestes
- Exemples:
  - QEMU, Plex86, Bochs
  - Kego-fusion (émulateur console SEGA)
    - Executor (émulateur Mac sur pc) : http://www.emaculation.com/executor.php

Hyperviseur (para-virtualisation)

- C'est un noyau dédié à la gestion de machines virtuelles. C'est une optimisation de la technique “superviseur”: au lieu d'avoir un OS et un logiciel de virtualisation, on a directement un noyau dédié à la virtualisation.
- Exemples:
  - Vmware ESX server
  - XEN
  - Hyper-V
  - Historiquement: IBM CP, VM



Hardware Level

Virtualisation matérielle

- Support de la virtualisation directement dans le processeur. Pour les x86, il s'agit surtout de rendre possible l'éxécution des instructions privilégiées du mode protégé par les guests directement afin d'optimiser les performances ou de ne pas avoir à modifier l'OS invité.
- Exemples:
  - Mainframes VM/CMS
  - Intel VT
  - AMD-V

Virtualisation du poste de travail