Virtualisation

Systèmes d'exploitation centralisés 1 IMA

18 mars 2016

Sources :

- Chapitre 16 de "Operating System Concepts" (9ème édition), de Silberschatz, Galvin et Gagne
- Cours de Gérard Padiou, 1IMA 2012-2013

Objectifs de cette partie

- Idée et intérêt de la virtualisation
- Différentes formes de virtualisation
- Mise en œuvre de la virtualisation
  - Virtualisation du contrôle de l'accès aux ressources (au niveau du processeur)
  - Virtualisation des ressources : interface avec le système d'exploitation

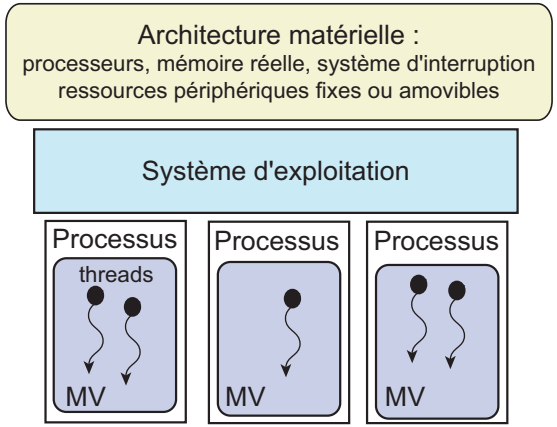
plan

Plan

Vision standard

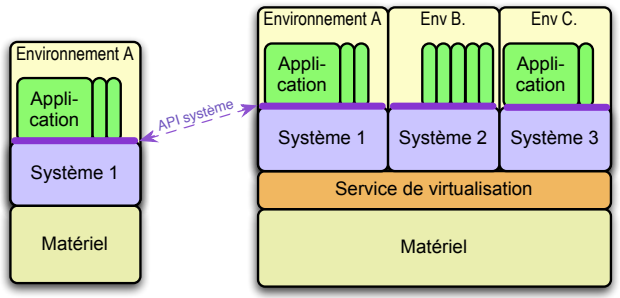
Le système d'exploitation abstrait le matériel pour les applications

→ fournit un environnement d'exécution (ressources, matériel, utilitaires) pour les applications



Besoin supplémentaire

Avoir plusieurs environnements d'exécution sur une même machine



Pour quoi faire ?

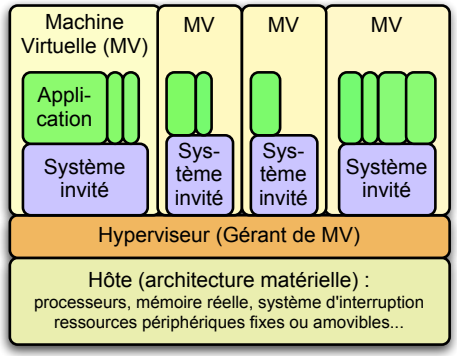
- économique : argent, place, énergie...
- pratique :
  - utilisation simultanée d'applications nécessitant des environnements différents, sans redémarrage ni reconfigurations
  - développement et mise au point d'un environnement ou d'un système sans perturber les autres applications ou utilisateurs

Principe

- Partager le matériel d'une même machine entre plusieurs d'environnement d'exécution.
- Ce partage doit être transparent : chaque environnement doit avoir l'illusion de disposer seul et directement des ressources matérielles qui lui sont nécessaires.

Similaire à ce fait un système d'exploitation pour les applications

- en première approximation, l'objectif est de réaliser un système d'exploitation pour les systèmes d'exploitation ;
- différence : transparence du partage, sans altérer (simplifier, abstraire) l'interface matérielle.



Terminologie

(machine) hôte environnement matériel (physique) sur lequel sera installé le service de virtualisation.

machine virtuelle un environnement d'exécution, requérant une configuration matérielle spécifique

hyperviseur (ou gérant de machines virtuelles) service de virtualisation proprement dit, fournissant une interface identique à celle de l'hôte.

(système) invité système d'exploitation/application utilisant 1 MV

Intérêt et utilisation de la virtualisation

Protection/Isolation : l'hyperviseur seul contrôle l'accès aux ressources réelles → une MV ne peut accéder à d'autres ressources que celles qui lui sont allouées par l'hyperviseur

→ un plantage, un virus sur une MV n'affecte pas les autres MV

→ communication entre MV par réseau ou serveur de fichiers

Sauvegarde/restauration de l'état (cliché) d'une MV

- protection contre les incidents en cours d'exécution
- transfert d'images de MV entre hôtes
  - → équilibrage de charge (mécanisme de **migration à chaud**)
  - → adaptation à la charge : (dés)activation d'hôtes selon la charge
  - → mise à jour, installation d'applications simplifiés et automatisés

Exécution sur un **même hôte** d'**environnements hétérogènes**

- personnalisation : une MV par contexte d'utilisation
- consolidation (adaptation à la charge)

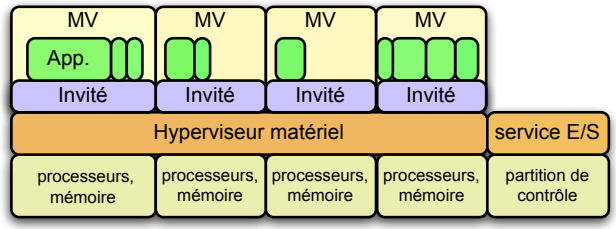
→ ingrédients de l'**informatique en nuage** (cloud computing)

Plan

Objectifs

- transparence vis à vis des applications
  → préservation des performances
- limiter les modifications à apporter sur les systèmes invités

Hypervision matérielle

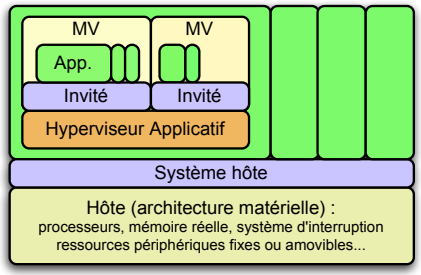


- hyperviseur réalisé par microprogramme (firmware)
- partition des ressources physiques entre invités
- les invités n'ont pas à être modifiés
- efficace mais limité en fonctionnalités et nombre d'invités
- difficulté : partage des périphériques → partition de contrôle pour un invité jouant le rôle de serveur pour les autres invités
- exemples : LPAR (IBM), LDOM (Oracle)

Hypervision logicielle

- système d'exploitation pour les systèmes d'exploitation
  - services de gestion des invités plutôt que d'accès aux ressources
  - exécution en mode superviseur
  - invités vus comme des processus ordinaires
    avec toutefois la nécessité de traiter les demandes d'exécution d'instructions privilégiées par les invités.
  - fourniture de pilotes spécifiques pour les périphériques, pour en contrôler l'accès
  - transparent pour les invités
- essentiel dans les centres de calcul/données intensifs
  - administration automatisée d'un grand nombre de MV, déployées sur un grand nombre d'hôtes (clichés)
  - adaptation et régulation de la charge (migration)
- exemples : ESX (VMware), XenServer (Citrix), SmartOS (Joyent)
- l'hypervision logicielle peut aussi être fournie comme service d'un système d'exploitation standard :
  HyperV (Windows Server), Solaris, KVM (Linux RedHat)

Hypervision applicative



- hyperviseur = application standard, exécutée sur l'hôte
- difficulté : efficacité réduite
  - surcoût induit par la couche système hôte
  - transparent pour l'hôte
    → pas de support pour le mode privilégié et les opérations de support à l'exécution disponibles au niveau du processeur
- avantage : souplesse d'utilisation
  (lancement sans intervention sur le système hôte)
- exemples : Parallels Desktop, VirtualBox(Oracle), VMware Workstation

Exemple de configuration PC virtualisée par Parallels Desktop

- Un processeur Intel Pentium.
- Jusqu'à 1500 MB de mémoire.
- une carte graphique VGA.
- Jusqu'à 4 disques d'interface IDE, dont un disque de démarrage (de 20 à 128 GB représenté par un fichier) .
- Jusqu'à 5 interfaces réseau , dont une carte virtuelle Ethernet.
- 4 ports séries (COM) ports.
- 3 ports parallèles (LPT)
- un contrôleur USB 2.0.
- une carte son.
- un clavier PC générique.
- une souris PS/2.

Paravirtualisation

Idée

gagner en efficacité → abandon de la transparence pour l'invité

- l'hyperviseur propose une interface similaire mais non identique à l'hôte
  → les systèmes invités doivent être adaptés pour cette interface
- l'interface proposée
  - offre un accès simplifié et plus efficace aux ressources gérées par l'hyperviseur (E/S en particulier)
  - permet à l'invité de déléguer explicitement à l'hyperviseur (appels hyperviseur) certaines opérations d'accès aux ressources matérielles (mémore virtuelle, instructions privilégiées)
    → allège la supervision par l'hyperviseur, et limite les traitements en double (par l'invité et par l'hyperviseur)
- de moins en moins nécessaire, le support matériel (par les processeurs) à la virtualisation s'améliorant progressivement.
- exemple : Xen

Environnement d'exécution virtuel

Idée

proposer un environnement d'exécution virtuel, conçu pour une plateforme de développement donnée.

Exemples : Java, .Net
La virtualisation consiste alors à implanter cet environnement (JVM Java) sur une architecture matérielle :

- les programmes Java sont compilés pour un code machine (bytecode) indépendant des plateforems-matérielles, destiné à être exécuté par la JVM
- la JVM est compilée pour chaque architecture matérielle
- la JVM lit et exécute le bytecode

→ les programmes écrits en Java peuvent alors s'exécuter indépendamment de l'hôte sous-jacent

Emulation

Limite de la virtualisation

Le système invité doit (aussi) pouvoir s'exécuter directement sur le système hôte

→ le processeur de l'hôte doit être compatible avec celui pour lequel le système invité a été installé.

→ tous les services de virtualisation présentés jusqu'ici sont destinés aux processeurs de la famille Intel x86

L'émulation consiste à traduire (généralement à la volée) les instructions du processeur invité en instructions du processeur hôte

- Avantage : pas de modification du système « invité »
- Inconvénient : performances faibles.
- Exemples : MAME (machines d'arcade), projet Bochs (http ://bochs.sourceforge.net/), environnements de développement pour applications mobiles. . .

Plan

Contrôle de l'accès aux ressources

Problème

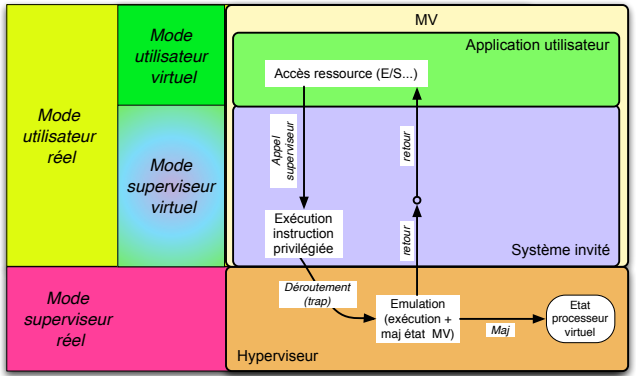
- garantir que l'hyperviseur est seul à pouvoir accéder directement aux ressources matérielles
  → nécessité d'un processeur avec deux modes : utilisateur et superviseur
- les utilisateurs de l'hyperviseur sont eux mêmes des systèmes d'exploitation qui ont besoin de deux modes : utilisateur et superviseur
- le mode superviseur réel doit rester réservé à l'hyperviseur

→ implanter/émuler en mode utilisateur réel
un mode utilisateur virtuel et un mode superviseur virtuel.

Technique 1 : trap-and-emulate

Principe

- l'exécution d'une instruction privilégiée par l'invité provoque un déroutement (trap)
- l'hyperviseur prend le contrôle, analyse l'erreur et exécute l'opération pour l'invité avant de lui rendre le contrôle
- perte d'efficacité limitée aux instruction privilégiées



Technique 2 : traduction binaire

Limite de trap-and-emulate : certaines instructions « spéciales » (sur les x86 en particulier) peuvent avoir un comportement différent en mode utilisateur et en mode superviseur, sans engendrer de déroutement...

Exemple : `popf` (x86)

- modifie le registre de flags à partir du contenu de la pile
- en mode privilégié, modifie tous les flags
- en mode utilisateur, n'en modfie que certains

Solution : émulation des instructions « spéciales »

- l'hyperviseur examine chaque instruction de l'invité
- les instructions ordinaires sont exécutées normalement
- les instructions « spéciales » sont traduites et exécutées à la volée

Remarques

- Des optimisations (p. ex. utilisation de caches pour les traductions) permettent de ne pas trop dégrader les performances
- Les processeurs actuels permettent d'éviter la traduction binaire en proposant plus de deux modes.

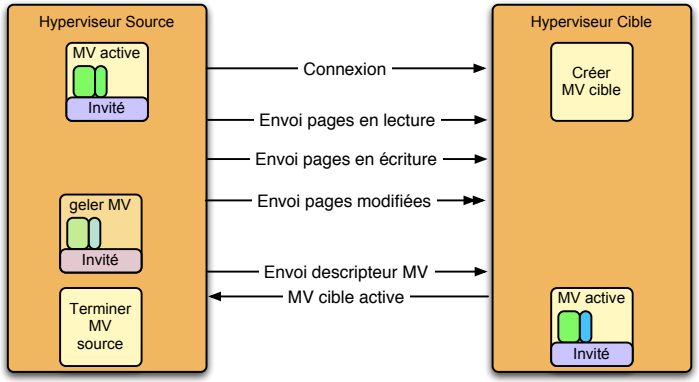
Interface avec les systèmes invités

Principale difficulté : surréservation de ressources

La demande (instantanée) de ressources par les invités peut excéder les ressources effectivement disponibles.

- Processeur : peu à faire. (Correction du retard pris par les horloges)
- Mémoire
  - installation d'une application « ballon » dans chaque invité, contrôlée par l'hyperviseur
    - réclamant (et verrouillant) des pages lorsque la mémoire manque. Ces pages de mémoire peuvent alors être récupérées par l'hyperviseur.
    - libérant ces pages (rendues par l'hyperviseur) lorsque la pression diminue
  - détecter les pages chargées en lecture en double par des invités différents, et les rendre partagées.

Migration à chaud de machines virtuelles



Remarque

transfert seulement de l'image mémoire et de l'état processeur

→ accès disques par serveur de fichiers