Xen

Anne Facq
Jérome Castang
Laurent Lavaud

Octobre 2007

RAISIN
Réseau Aquitain des Informaticiens Systèmes INter-établissement

Plan

Introduction / Notions sur la virtualisation

- Couche d'abstraction matérielle et/ou logicielle
- OS hôte (ou système d'exploitation hôte)
  = système installé directement sur le matériel
- OS invités ou OS virtualisés
  = systèmes installés sur le système hôte
- Partitionnement, isolation et/ou partage des ressources physiques et/ou logicielles
- Images manipulables
  - démarrage, arrêt, gel, clonage, sauvegarde et restauration, sauvegarde de contexte, migration d'une machine physique à une autre
- Réseau virtuel : réseau purement logiciel, interne à la machine hôte, entre hôte et invités

Introduction / Systèmes de virtualisation (1)

- Linux-Vserver, BSD Jail, chroot
  → Isolateurs
  - Isole certains aspects ou ressources de l'OS hôte (systèmes de fichiers ou espaces mémoires)
  - Pas d'empilement de l'OS hôte et d'un logiciel de virtualisation
  - Très performant mais environnements virtuels sont peu ou pas complètement isolés
- UML
  → Noyau en espace utilisateur
  - Se lance comme une application dans l'OS hôte
  - UML lance et gère ses applications de manière isolée des autres UML tournant sur la même machine
  - Peu performant car les 2 noyaux sont empilés

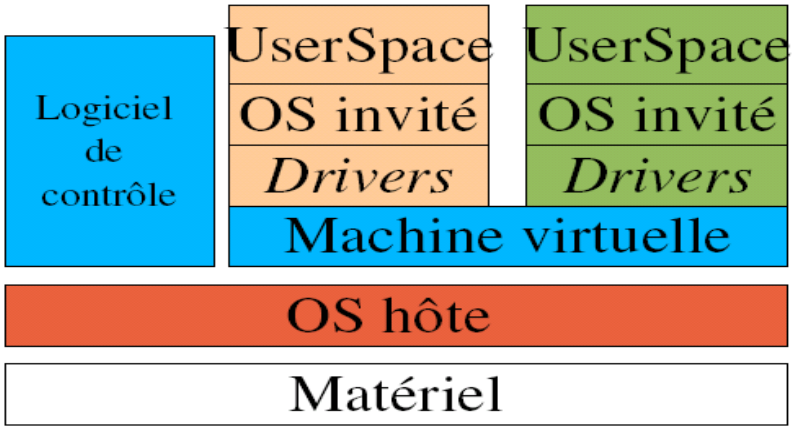
Introduction / Systèmes de virtualisation (2)

- QEMU sans Kqemu, VirtualPC sur PowerPC
  → Emulation matérielle

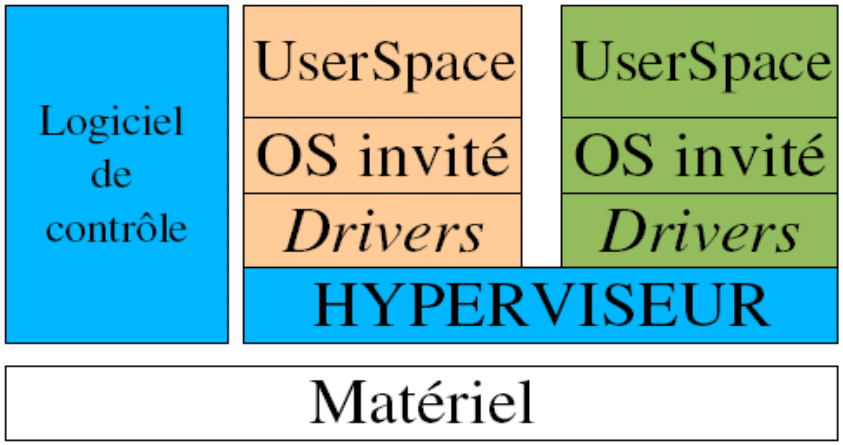
  - Machine virtuelle complète (bios, processeur, mémoire, disque, carte réseau...)
  - Intercepte majorité des instructions de l'OS invité pour les remplacer par leur équivalent sur l'OS hôte
  - Avantages : exécution des applications prévues pour d'autres architectures (ordinateurs, consoles, bornes d'arcade ...)
  - Inconvénients : performances médiocres

Introduction / Systèmes de virtualisation (3)

- QEMU avec Kqemu, Vmware Workstation/GSX
  → Virtualisation complète
  - machine virtuelle complète (bios, processeur, mémoire,
  - disque, carte réseau...)
  - Intercepte certaines instructions particulières de l'OS invité (ne pouvant pas être exécutées sur OS hôte)
  - Simple à mettre en oeuvre
  - Empilement OS hôte + OS invité
    → surcharge
    → performance moyenne

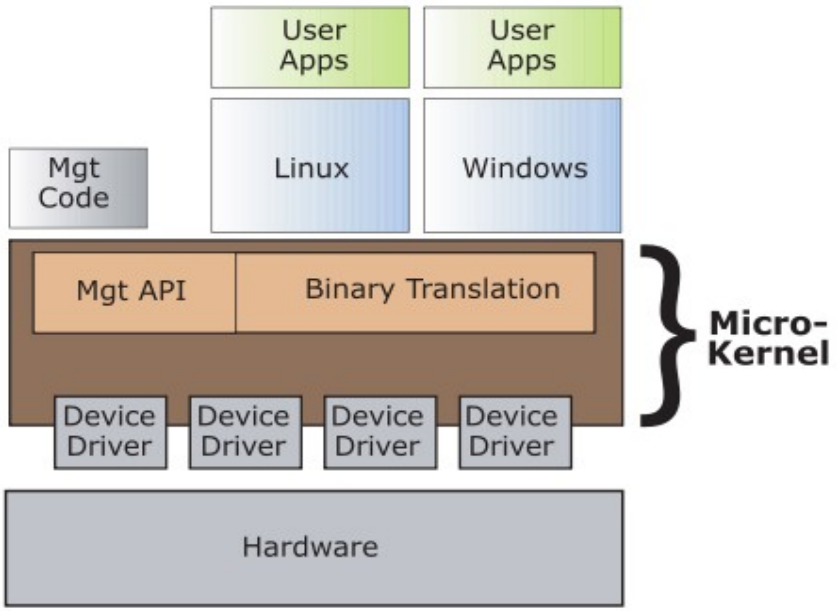


- Xen
  → Paravirtualisation

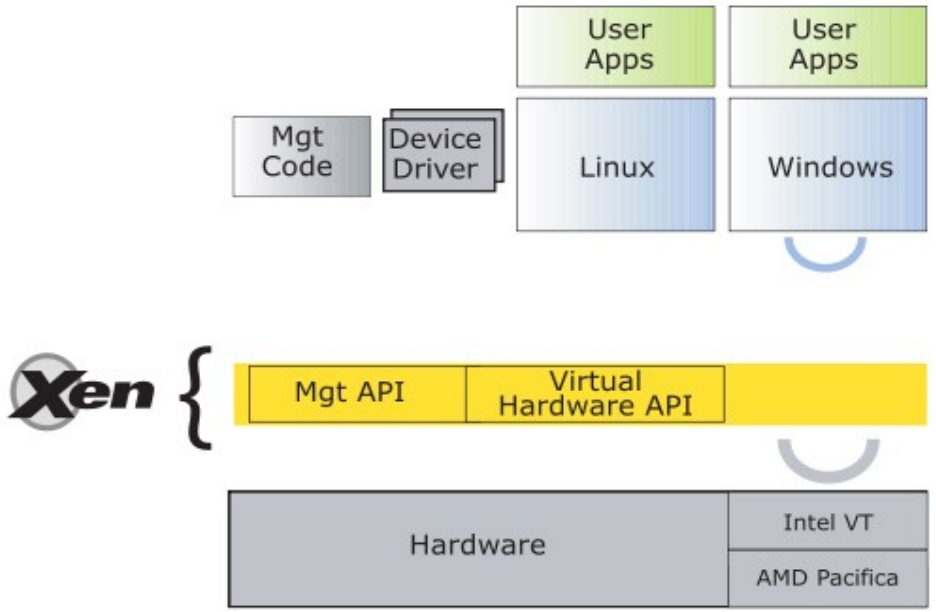


  - Hyperviseur
    = moniteur de machines virtuelles
    (VMM Virtual Machine Monitor)
  - OS invité fonctionne directement sans interception des instructions
  - Xen 3 peut héberger des OS invités non modifiés

Introduction / Système de virtualisation / Comparaison



Première génération des systèmes de virtualisation



Virtualisation avec Xen

Introduction / Systèmes de virtualisation du type de Xen

- Ils disposent de la couche logicielle VMM (Virtual Machine Monitor) dédiée à la gestion de machines virtuelles
  - VMWare ESX
    - Machine virtuelle se repose sur un noyau léger (vmkernel)
    - empilage des machines est léger
  - Microsoft Virtual Server
  - HP Integrity VM
  - Micropartitions d'IBM
- XenSource développe avec Microsoft un hyperviseur Xen pour Windows Server 2008 (Longhorn)

Introduction / Origines de Xen

- Xen était un projet de recherche à l'Université de Cambridge memé par Ian Pratt

- Ian Pratt a fondé XenSource qui développe le projet open source et vend les versions de Xen pour les entreprises

- 1ère version publique de Xen : octobre 2003
- Xen 2.0 : octobre 2004
- Xen 3.0 : décembre 2005

Introduction / Avantages de Xen

- Isolation complète entre les machines virtuelles
- Performances pour les machines virtuelles proche d'un système natif
- Très bon support du matériel (Xen utilise les pilotes du noyau linux)
- Possibilité de migrer des machines virtuelles entre des serveurs Xen sans interruption de service
- Open Source (license GPL)

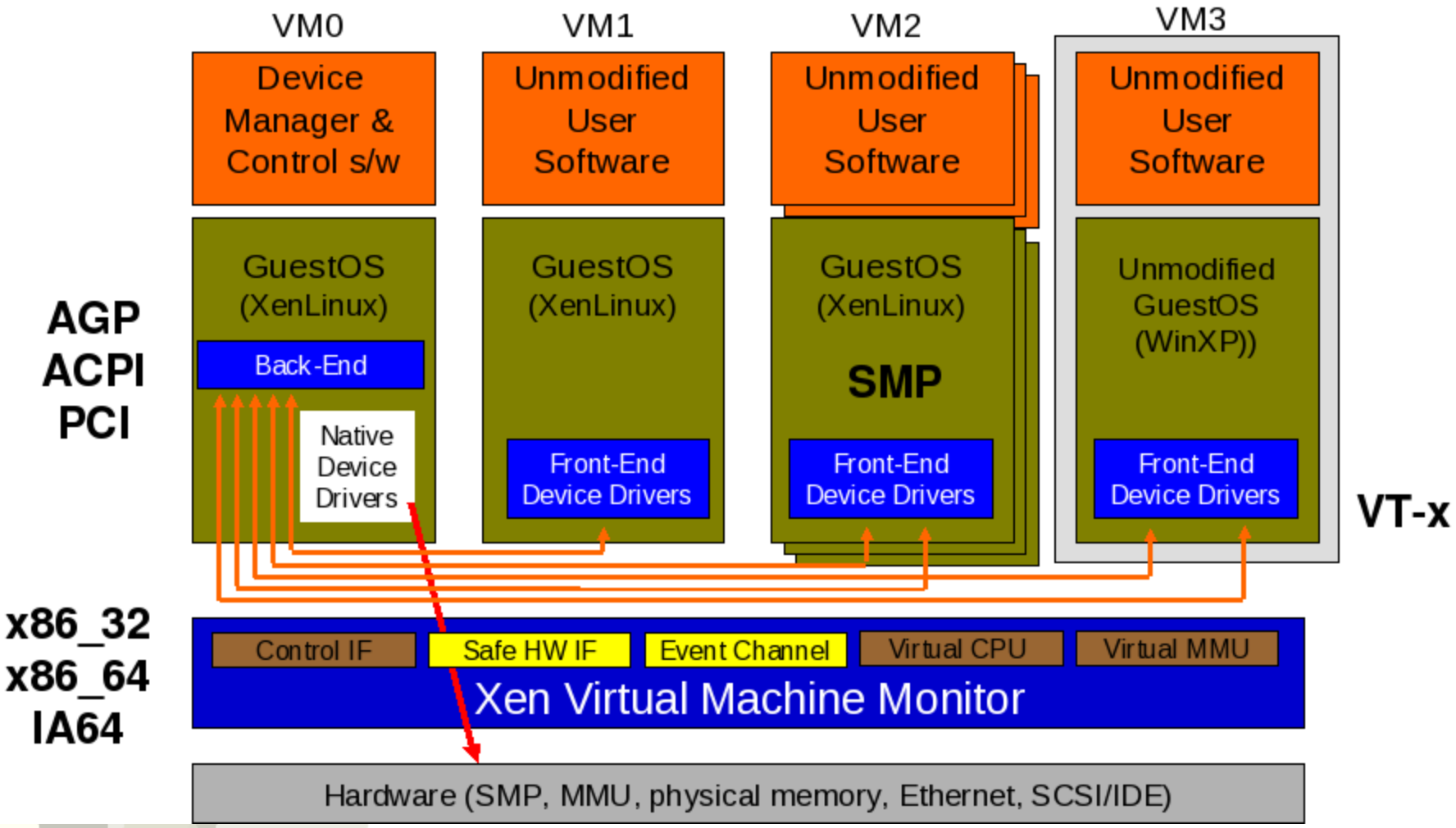
Inroduction / Processeurs supportés par Xen

- Xen tourne sur
  - Processeurs de type x86 (processeur de type "P6" ou Intel ou AMD des 5 dernières années)
  - Machines multiprocesseur et inclut le support pour l'hyper-threading (SMT).
  - Processeurs de type x86/64 (depuis Xen 3.0)
  - Processeurs de type IA64
  - Processeurs de type PPC
- Xen est en cours de portage sur processeurs de type ARM (liste de diffusion sur xen source)

Plan

- Introduction
- Architecture de Xen
  - Hyperviseurs / domaines
  - Les rings
  - Les technologies de virtualisation matérielles
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Architecture de Xen



Architecture de Xen / Hyperviseur

- Architecture d'un système Xen est composée de
  - hyperviseur Xen
  - machines virtuelles sécurisées appelées domaines
    - dom0 - privileged domain
    - domU - unprivileged domain
    -
- Hyperviseur
  - ordonnance temps d'utilisation de la machine hôte par chaque domaine (dans temps imparti, les OS invités ordonnancent leurs processus)
  - au boot de l'ordinateur, détecte et démarre les processeurs non initialisés par le BIOS
  - route les interruptions, énumère les bus PCI

Architecture de Xen / Le Dom0

- Dom0 (domaine privilégié)
  - crée lors de l'installation de xen
  - lancé automatiquement au boot
  - composé d'un noyau linux modifié et des logiciels de contrôle de Xen
  - le seul à pouvoir interagir directement avec le matériel via les pilotes du noyau linux
  - autres domaines font appel a ces pilotes via l'utilisation des pilotes (virtuels) de Xen
  - assure tache d'administration du système via le démon xend dans espace utilisateur (création, démarrage, arrêt, restauration ou migration des domaines)
  - gère les pilotes natifs et pilotes virtuels des domaines

Architecture de Xen / Les DomU

- DomU (Domaine non privilégié)
  - machines invitées ou OS invités
  - leur noyau est chargé dans un mode non privilégié du processeur (en général ring 1 ou le ring 2)
  - ces machines sont contrôlées par le dom0.

Architecture de xen / Les rings (1)

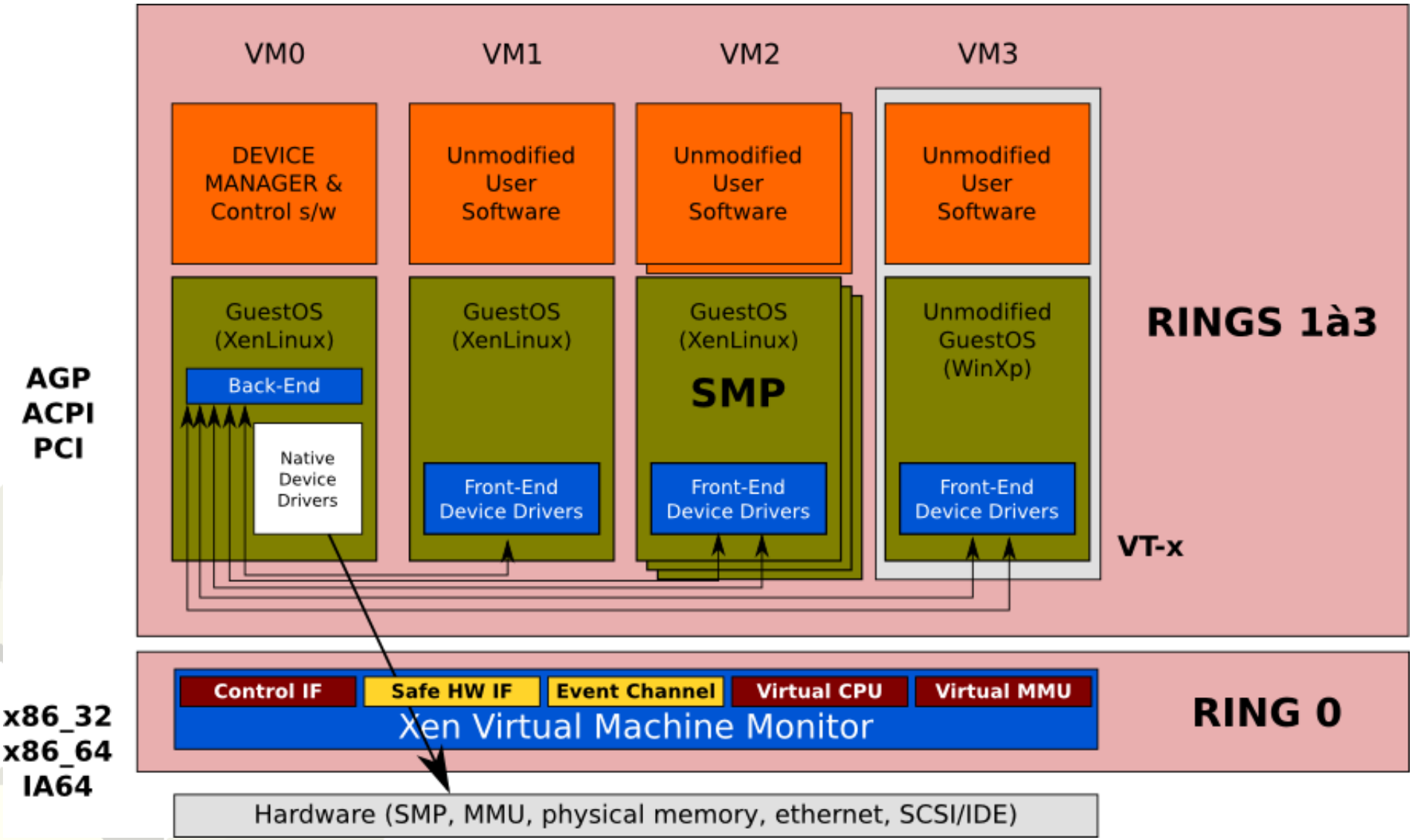
- Processeurs compatibles x86 ont un modèle de protection de 4 niveaux d'exécution = les rings

- Niveaux numérotés de 0 → 3
  (0 = plus privilégié, 3 = moins privilégié)
  - ring 0 dédié à l'exécution de l'OS
  - ring 3 dédié aux applications de l'espace utilisateur
  - rings 1 et 2 prévus à l'origine pour virtualisation

Architecture de xen / Les rings (2)

- Système Xen sur architecture x86
  - hyperviseur dans ring 0
  - dom0 = domaine privilégié mais n'est pas dans ring0
    → tout transite par l'hyperviseur
  - OS invités dans ring 1 ou 2
  - applications dans ring 3
- Système Xen sur architecture x86_64 (64 bits)
  - hyperviseur dans ring 0
  - OS invités et applications dans ring 3
  - ring 1 et 2 ont été supprimés

Architecture de Xen / Les rings (3)



Architecture de xen / Technologies de virtualisation matérielle (1)

- Exemples :
  - Intel Vanderpool x86 virtualization (VT-i)
  - AMD Pacifica x86 virtualization (AMD-v)
  - Sun UltraSPARC T1 hypervisor
  - IBM Advanced POWER virtualization
- Depuis 2005 interface commune d'accès aux 2 technologies Intel et AMD : HVM (Hardware Virtual Machine)
- Permettent à Xen d'accepter des OS invités non xenifiés (ex : Windows)

Architecture de xen / Technologies de virtualisation matérielle (2)

- Intel Vanderpool x86 virtualization (VT-i)
  - Intel a ajouté 2 modes d'exécution : VMX root et VMX non-root
  - ces 2 modes supportent les 4 rings de 0 → 3
    - ➔ OS utilisent le ring 0, OS non modifiés
    - ➔ applications en ring 3
  - OS et applications fonctionnent dans mode VMX non-root
  - hyperviseur utilise mode d'exécution VMX root (niveau contrôle et privilège + important)
  - hyperviseur est modifié pour utiliser extension Vanderpool et gérer les modes VMX root et VMX non-root

Architecture de Xen / Xend

- Le démon xend (écrit en python)
  - fonctionne dans dom0
  - interface http sur port 8000
  - répond aux requêtes venant du dom0
    - création, destruction, migration, arrêt, démarrage, sauvegarde, restauration, surveillance des domaines,
  - fichier de configuration : xend-config.sxp
    - Méthode Gestion du réseau
    - Paramètres de migration
  - xend crée 2 démons (xenstored, xenconsoled) et une instance de la classe python SrvDaemon
  - fichiers de log de xend
    - /var/log/xend.log
    - /var/log/xend-debug.log

Architecture de Xen / Relations machines virtuelles - domaines

- Relations entre machines virtuelles et domaines
  = relation entre programmes et processus

  - machine virtuelle = entité permanente qui réside sur disque (comme un programme).
    Quand elle est chargée pour exécution elle fonctionne dans un domaine

  - chaque domaine possède un identificateur identique = identificateur de processus

Architecture de Xen / Avantages

- Garantit à l'hyperviseur protection contre bugs et plantages des pilotes
- Décharge équipe de Xen du développement des pilotes
- Assure au système Xen un large support materiel via pilotes du noyau Linux
- Simple et légère hautes performances de Xen

Plan

- Introduction
- Architecture de Xen
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Le noyau Xen

- Xen fonctionne avec 2 noyaux
  - 1 noyau chargé dans dom0
  - 1 noyau chargé dans domU
- Noyau dans dom0
  - chargé dans ring 1/2
  - gère accès au matériel spécifique (backend)
- Noyau dans domU
  - chargé dans ring3
  - frontend communiquent avec les backend

Le noyau Xen / Virtualisation des pilotes (1)

- Hyperviseur et noyau Linux du dom0
  - ont un accès direct aux pilotes des périphériques et connaissance de ceux-ci
  - utilisent pour chaque périphérique le pilote et la configuration fournis par le dom0
- le dom0 exporte une version virtualisée des périphériques vers les OS invités
- les périphériques sont contrôlés par le système du dom0 et sont disponibles pour tous les OS invités
- Le système linux du dom0 doit être configuré pour
  - supporter le matériel sous-jacent
  - l'hyperviseur
  - fournir des périphériques virtuels

Le noyau Xen / Virtualisation des pilotes (2)

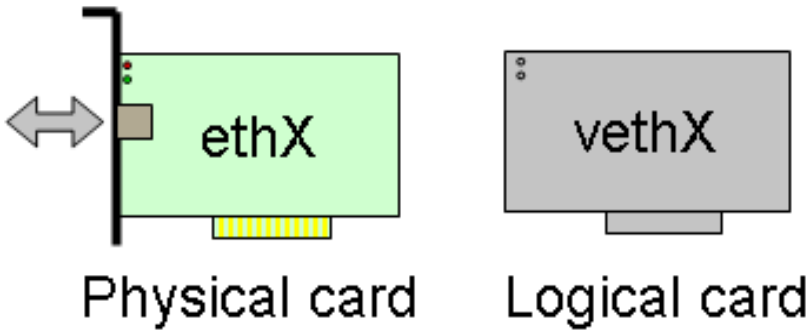
- Communication entre périphériques d'un domU et périphériques virtuels du dom0 passe par un canal (device channel) qui
  - est 1 lien point à point entre les 2 domaines
  - permet envoi de message asynchrone d'un domaine vers un autre
- Messages communiqués via page de mémoire partagée allouée par l'OS invité et mappée dans l'espace d'adressage du dom0
- Avantages
  - limite crash d'un pilote au pilote lui-même sans affecter application située dans autre domaine
  - permet de redémarrer dom0 pour retrouver un pilote opérationnel en perturbant le moins possible les applications des autres domaines

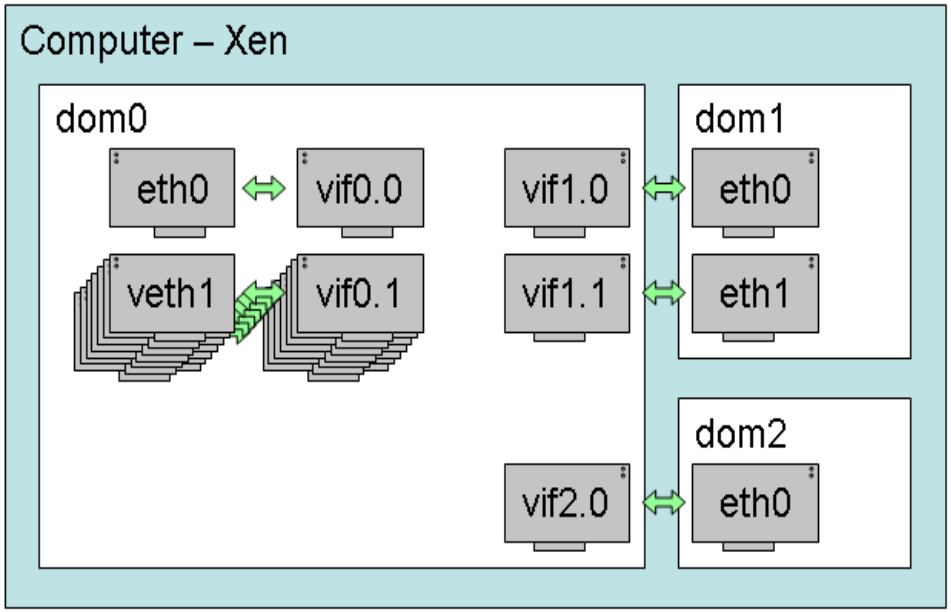
Plan

- Introduction
- Architecture de Xen
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Le réseau dans Xen / Interfaces ethernet virtuelles

- Xen crée 7 paires d'interface virtuelles ethernet utilisées par le dom0
- Ces paires sont 2 interfaces ethernet reliées par un câble interne croisé
  - eth0 est relié à vif0.0
  - veth1 est relié à vif0.1
  - etc...
  - veth7 est relié à vif0.7





Le réseau dans Xen / Interfaces virtuelles (1)

- Chaque fois que l'on crée un domU, un nouvel identifiant est associé
  - identifiant du 1er domU est 1
  - identifiant du 2ème domU est 2 même si le 1er domaine est arrêté
- Xen crée des paires d'interfaces virtuelles ethernet pour chaque nouveau domU
  - une partie de chaque paire est dans le domU. Dans un domU sous Linux le nom du périphérique réseau est eth0
  - l'autre partie de chaque paire est dans le dom0 et se nomme vif<id_domaine>.0

Le réseau dans Xen / Interfaces virtuelles (2)

- Exemple : dans domU dont l'identifiant = 5, eth0 sera attaché à vif5.0 dans le dom0)
- Si on crée des interfaces réseau multiples dans un domU, eth0 et eth1, le dom0 a vif<id_domaine>.0 et vif<id_domaine>.1
- Quand on lance shutdown sur un domU, les adresses virtuelles ethernet sont détruites

Le réseau dans Xen / Adresses MAC (1)

- Par défaut Xen sélectionne une adresse MAC aléatoire pour chaque interface réseau virtualisée des domU
- Adresse MAC est différente sur chaque domU
- Pour fixer une adresse MAC pour un domaine (ex : utilisation de dhcp), il faut utiliser l'option vif dans le fichier de configuration de la machine virtuelle
- Ex : vif = ['mac=aa:00:00:00:00:11']).

Le réseau dans Xen / Adresses MAC (2)

- Choix de l' adresse MAC : adresse de type unicast
  - le dernier bit du 1er octet doit être positionne a 0
  - l'avant-dernier bit du 1er octet doit être positionne a 1
  - forme : XY:XX:XX:XX:XX:XX
  - X = chiffre hexadécimal
  - Y = 2 ou 6 (110) ou A (1010) ou E (1110).

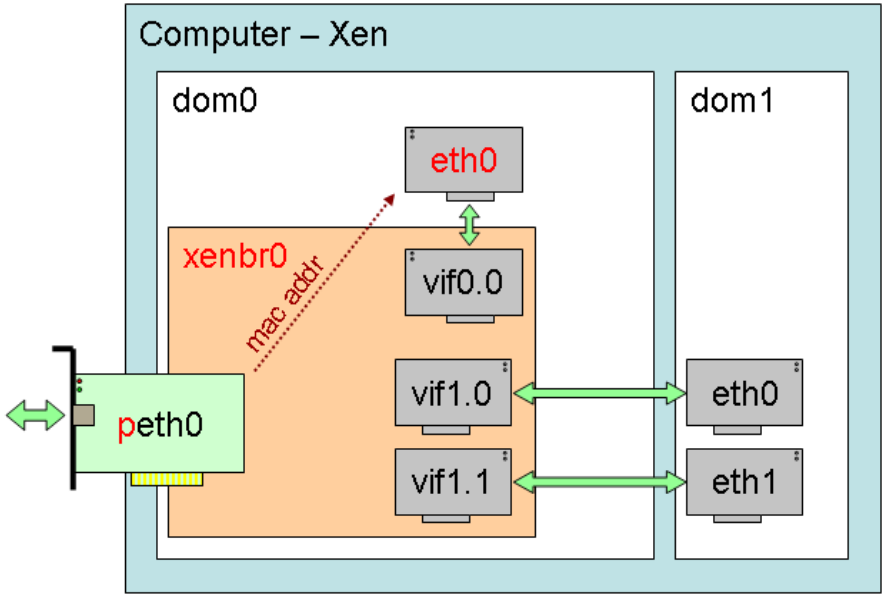
- Exemple
  - AA:XX:XX:XX:XX:XX → CORRECT
  - AB:XX:XX:XX:XX → INCORRECT

Le réseau dans Xen / Modes réseau de Xen

- Mode bridge
  - Par défaut, Xen utilise un pont à l'intérieur de dom0 (xenbr0)
    → permet à tous les domaines d'apparaitre sur le réseau comme des machines individuelles
- Mode NAT
  - dom0 joue le rôle de passerelle pour les domU.
  - les vifX.Y ont pour IP celles des cartes des domU (ex : vif1.0 dans dom0 a pour IP celle de la carte eth0 dans dom1).
  - règles iptables applicables à ces cartes sur dom0.
- Mode route
  - les vifX.Y ont pour IP celle des cartes dans les domU
  - elles ne voient pas passer les paquets.

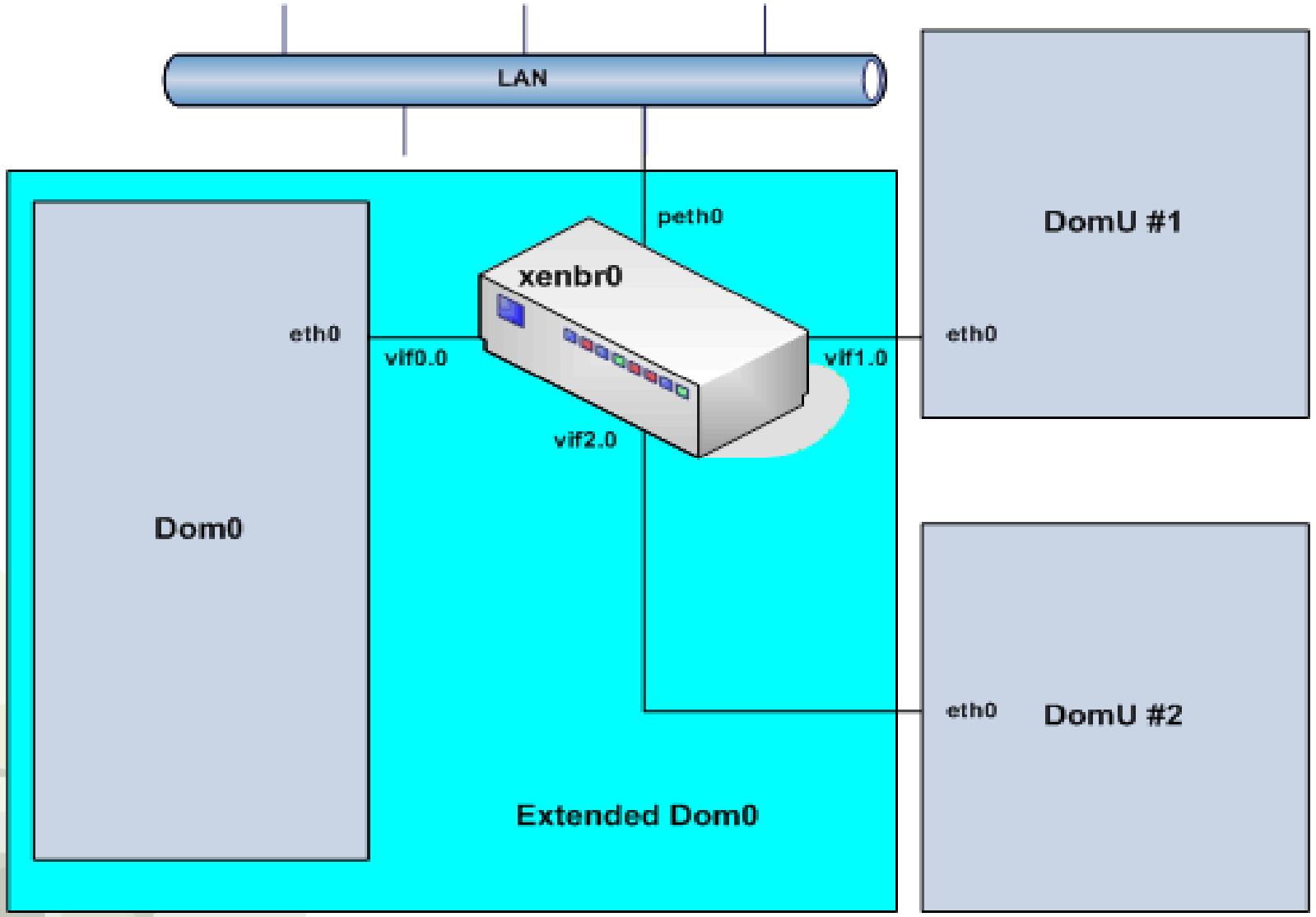
Le réseau dans Xen / Xen en mode bridge (1)

- peth0
  - interface physique (pas configurable)
  - lien entre le système et carte réseau
- eth0
  - interface virtualisée fournie par Xen (configurable)
  - connexion réseau pour le système hôte
- vifX.Y
  - cartes réseau des OS invités
  - ex : vif1.0 correspond à eth0 du domU dont l'id est 1



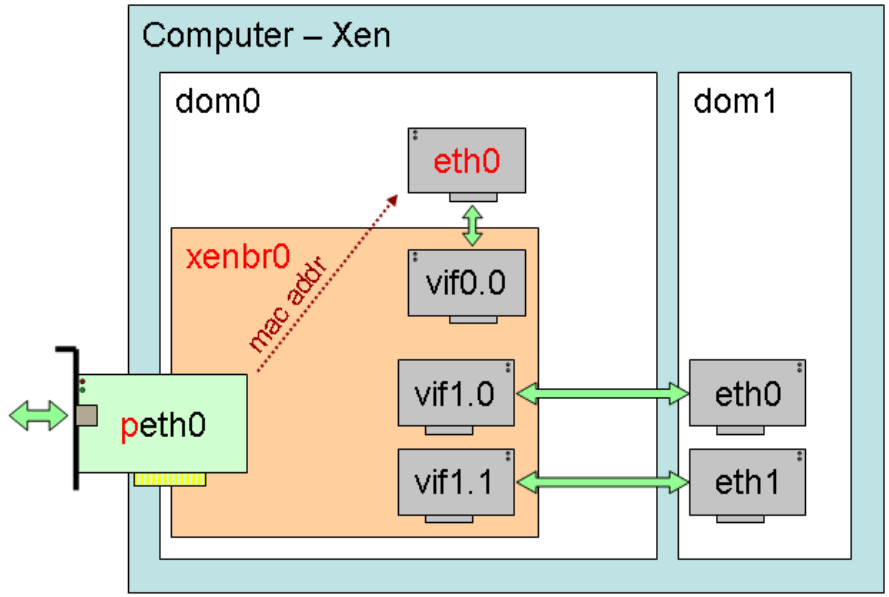
Mode bridge
= pont entre
carte physique
et cartes virtuelles

Le réseau dans Xen / Xen en mode bridge (2)



Le réseau dans Xen / Xen en mode bridge (3)

- Paquets sortants des cartes des domU sont directement retransmis sur carte physique peth0
- Aucun contrôle sur les paquets au niveau de dom0 (par exemple iptables).
- Les vif :
  - pas d'adresse ip,
  - directement reliées à eth0
- Paquetage bridge-utils est nécessaire



Le réseau dans Xen / Xen en mode bridge (4)

- Démon xend gère création/suppression des ponts et interfaces virtuelles via les scripts
  - network-bridge
  - vif-bridge
- Quand xend démarre, il lance le script network-bridge qui :
  - crée un nouveau pont nommé xenbr0
  - la « vraie » interface ethernet eth0 est mise à down
  - les adresses IP et MAC de eth0 sont copiées sur la nouvelle interface réseau virtuelle veth0
  - l'interface réelle eth0 est renommée peth0
  - l'interface virtuelle veth0 est renommée eth0
  - peth0 et vif0.0 sont attachées au pont xenbr0
  - le pont,peth0, eth0 et vif0.0 sont positionnés à up
- Quand un domU démarre, xend (tournant dans dom0) lance le script vif-bridge qui :
  - attache vif<id#>.0 à xenbr0
  - vif<id#>.0 est positionnée à up

Le réseau dans Xen / Les VLANS (1)

- Il est possible de faire tourner plusieurs machines virtuelles dans des vlans différents
- Il faut ajouter le support 802.1q dans dom0
- Un pont est configuré pour chaque VLAN, et les OS invités attachent leurs interfaces au pont approprié
- C'est au niveau de dom0 que les trames sont taggées. Les domU n'ont pas connaissance des vlans.

Le réseau dans Xen / Les VLANS (2)

- démon xend qui gère la création/suppression des ponts et interfaces virtuelles via deux nouveaux scripts qui remplacent network-bridge :
  - network-multi-vlan
  - network-bridge-vlan
- quand xend démarre, il lance le script network-multi-vlan qui appèle le script network-bridge-vlan autant de fois que de vlans désirés
- network-bridge-vlan :
  - crée un pont nommé vlanbrnumvlan (p.ex. vlanbr100)
  - crée une nouvelle interface eth0.numvlan (p.ex. eth0.100) dans dom0 à l'aide de la commande vconfig
  - cette nouvelle interface et le pont sont positionnés à up
  - eth0.100 est attachée au pont vlanbr100
  - quand domU démarre, xend (tournant dans dom0) lance le script vif-bridge qui attache vif<id#>.0 au nouveau pont vlanbrnumvlan et positionne vif<id#>.0 à up

Plan

- Introduction
- Architecture de Xen
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
  - Avec xm
    - Arrêt / Démarrage
    - Gestion de la mémoire
    - Gestion des processeurs
    - Migration
  - Autres outils
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Gestion des domaines avec xm / Arrêt - Démarrage

- xm = (Xen management)
  = interface texte de gestion de Xen
  - xm list :
    affiche tous les domaines en fonctionnement

  - xm create [-c] <Configfile>
    démarre la machine virtuelle définie dans <configfile>
    Exemples :
    - xm create /etc/xen/vm1.cfg
      démarre vm1
    - xm create -c /etc/xen/vm1.cfg
      démarre vm1 et attache une console à vm1

Gestion des domaines avec xm

- xm shutdown [-a|-w] <Domain>
  provoque l'arrêt d'un domaine
  -a : arrête tous les domaines.
  -w : attend d'avoir terminé le shutdown pour redonner la main

- xm console <Domain>
  attache une console à un domaine
  Pour quitter la console : Ctrl -]

- xm top
  affiche monitoring temps réel des domaines

Gestion des domaines avec xm / Gestion de la mémoire

- Dans Xen hyperviseur contrôle allocation de mémoire des domaines
- Par défaut on ne donne pas quantité de mémoire pour machine dom0
- Quantité de mémoire lui étant attribuée par hyperviseur = quantité totale de mémoire restante sur la machine physique
- Dans fichier de configuration de chaque OS invité, on spécifie quantité de mémoire à attribuer lors de son lancement. Cette quantité de mémoire est :
  - déduite de la quantité de mémoire restante
  - donc retirée de la quantité de mémoire du dom0.

Gestion des domaines avec xm / Gestion de la mémoire

- Quantité de mémoire donnée à un OS invité n'est pas définitive.
- On peut modifier la quantité de mémoire attribuée à un système invité tout en continuant son exécution.
  - Si quantité de mémoire d'une machine virtuelle est insuffisante, on peut augmenter la taille mémoire qui lui est attribuée sans interruption de service.
  - Si quantité inutilisée de mémoire, on peut récupérer cette mémoire sans interruption de service.

Gestion des domaines avec xm / Gestion de la mémoire

- Attention :
  - Impossible de diminuer quantité de mémoire d'une machine virtuelle en dessous de la quantité initialement donnée
  - Quantité totale de mémoire attribuée pour domaines ne peut pas dépasser quantité de mémoire présente sur machine physique.

- Le système dom0 a les privilèges sur l'hyperviseur Xen,
  → une machine virtuelle ne peut pas changer sa quantité de mémoire (ni celle d'une autre machine).
  → toutes les opérations concernant mémoire attribuée sont effectuées par la machine dom0.

Gestion des domaines avec xm / Gestion de la mémoire

- Augmentation ou diminution de la mémoire (balloon driver) à l'aide de la commande xm :

xm mem-set <Domain> <Mem>

Exemple : modification mémoire dans domaine 5
xm mem-set 5 512

Gestion des domaines avec xm / Gestion des processeurs

- VCPU = Virtual Central Process Unit
- Dans Xen, 1 processeur = 1 VCPU
  (Sur un système hyperthread, 1 thread = 1 VCPU)
- Dom0 gère attribution de processeur
- Ajout/suppression à la volée de CPU virtuels

  xm vcpu-set <Domain> <nVCPUs>

  Exemple : ajout d' un 2eme cpu dans domaine 3

  xm vcpu-set 3 2
- Pour lister les VCPU affectés à une machine

  xm vcpu-list
- Pour forcer un domU sur un VCPU donné

  xm vcpu-pin <Domain> <VCPU> <CPUs>

Gestion des domaines avec xm / La pause

- Mise en pause du domaine
  - xm pause <Domain>
  - Exemple : xm pause vm1
    - Mise en pause de la machine vm1
    - vm1 continue de consommer des ressources (mémoire, ...)
    - vm1 est toujours présente dans la liste des machines
    - Statut de vm1 est gelé (pas éligible dans file d'attente de ordonnanceur de Xen)
- Pour sortir un domaine de l'état de pause
  - xm unpause <Domain>

Gestion des domaines avec xm / Sauvegarde et restauration

- Sauvegarde
  - xm save <Domain> <CheckpointFile>
  - Exemple : xm save vm1 vm1.chk
    - arrêt de la machine vm1 (état est similaire à l'hibernation)
    - sauvegarde d'une machine à un instant donné

- Restauration
  - xm restore <CheckpointFile>
  - Exemple : xm restore vm1.chk
    - restauration de vm1
    - reprise de l'exécution de vm1

Gestion des domaines avec xm / La migration (1)

- Transfert d'un domaine entre 2 hôtes (cad 2 machines physiques)
- 2 types de migration
  - « regular migration »
    - Mise en pause du domU à transférer
    - Copie du contenu de sa mémoire
    - Reprise de l'exécution dans la machine de destination
  - « live migration »
    - Idem à migration offline sauf mise en pause de domU
    - → La migration n'est pas visible par l'utilisateur
- « live migration » est préférable à « regular migration »
- Important : Les 2 hôtes doivent avoir accès aux images disques et doivent avoir même version de xen

Gestion des domaines avec xm / La migration (2)

- Migration de vm1 sur tp.raisin.fr
  - xm migrate --live vm1 tp.raisin.fr

- xend fait en sorte que le domaine continue de fonctionner alors que la migration est en cours
  → La durée de l'arrêt est de 60 à 300ms.

- Nécessaire de se reconnecter sur la console du nouveau domaine en utilisant la commande xm console

- Si un domaine migré à des connexion réseau déjà ouvertes, elles sont conservées

Gestion des domaines / Autres outils

- Outils graphiques
  - xenman (ubuntu)
  - virt-manager (redhat, fedora)
    → permet d'effectuer des operations de base (create, pause, destroy, top...)
- Outils textes
  - xen-create-image (ubuntu)
  - virsh (redhat, fedora)
  - exemples :
    - virsh dumpxml 3 > vm1.xml permet d'afficher le domaine (dont l'identifiant est 3) au format XML sur la sortie standard
    - virsh create vm1.xml permet de creer un domaine a partir d'un fichier xml

Plan

- Introduction
- Architecture de Xen
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Installation de Xen / Les fichiers de configuration

- Fichiers de configuration dom0:
  - xend-config.sxp
    - mode réseau
    - migration
  - script VLANs
  - un fichier de configuration par domU
    - caractéristiques de la machine
    - (Nom, @mac, mémoire, disque...)
- Fichiers de configuration domU:
  - pas de fichiers spécifiques à xen dans les DomU
  - configuration habituelle d'un OS Linux
- Remarque : ne pas activer NTP dans les domU

Installation de Xen / Gestion de l'espace disque

- Stockage des DomU
  - image disque (loop) sur disque local ou distant
  - partition d'un disque
  - partition LVM (Logical Volume Manager)
- NFS
  - export d'un espace commun
- NBD (Network Block Device)
  - stockage, duplication des machines
  - migration
- ISCSI
  - baie matérielle
  - export via iSCSI Enterprise Target
- GFS (Global File system)
  - système clusterisé
  - migration avec accès concurrent

Plan

- Introduction
- Architecture de Xen
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Sécurisation de Xen (1)

- Le dom0 doit être sécurisé le plus possible car si le dom0 est compromis tous les autres domaines sont également vulnérables

- Bonnes pratiques pour le dom0 :
  - ne lancer que le plus petit nombre de service nécessaires
  - utiliser un firewall pour restreindre le trafic vers le dom0
  - ne pas autoriser les utilisateurs à accéder au dom0
  - dom0 devrait être inaccessible

Sécurisation de Xen (2)

- Xen et Shorewall : doc de configuration de Shorewall dans le Dom0
  - http://www.shorewall.net/Xen.html
  - http://www1.shorewall.net/XenMyWay.html
- Faille de sécurité dans xen-pygrub
  - http://www.zataz.com/alerte-securite/15182/xen-pygrub-grub-conf-dom0-vuln.html

Plan

- Introduction
- Architecture de Xen
- Le noyau Xen
- Le réseau dans Xen
- Gestion des domaines
- Installation de Xen
- Sécurisation de Xen
- Utilisation de Xen
- Références

Utilisation de Xen (1)

- Consolider des serveurs en hébergeant sur une machine physique plusieurs serveurs différents (OS et applications)
- Réaliser des plans de reprise d'activité ou de continuité de service
- Tester le fonctionnement d'une application sur plusieurs systèmes sans disposer d'autant de machines
- Surveiller des machines virtuelles, allouer des ressources, détecter des comportements inhabituels

Utilisation de Xen (2)

- Réaliser un cluster en utilisant la flexibilité de gestion des machines virtuelles, leurs contrôles, leurs isolations et la possibilité de migrer pour repartir la charge
- Tester une architecture réseau
- Remplacer le dual-boot
- Tester et debugger des modifications du noyau dans une machine virtuelle « bac à sable » (sandboxed)

Utilisation de Xen (3)

- Listes de diffusion:
  - Xen-devel
    http://lists.xensource.com/xen-devel
  - Xen-announce
    http://lists.xensource.com/xen-announce
  - Xen-changelog
    http://lists.xensource.com/xen-changelog
  - Xen-users
    http://lists.xensource.com/xen-users
  - Fedora-xen
    http://www.redhat.com/mailman/listinfo/fedora-xen

La bibliothèque d'accès bas niveau libvirt

- API de manipulation de machines virtuelles
- Langage C
- Indépendante des technologies de virtualisation : fonctionne également avec KVM et Qemu
- Ne manipule que les machines virtuelles du serveur sur lequel sera utilisée l'API
- Exemple

```
#include <libvirt/libvirt.h>
...
ret = virDomainGetInfo(dom, &info);
printf("Le domaine %d a %d CPUs\n", id, info.nrVirtCpu);
```

- http://libvirt.org/architecture.html

Références (1)

- Site officiel du projet
  - http://www.cl.cam.ac.uk/research/srg/netos/xen/architecture.html
- Xensource
  - http://www.xensource.com/
  - http://wiki.xensource.com/xenwiki/XenIntro
  - http://wiki.xensource.com/xenwiki/XenDocs
  - http://wiki.xensource.com/xenwiki/XenFaq
- Xen sur Wikipedia
  - http://en.wikipedia.org/wiki/Xen
- Doc sur Xen en francais
  - http://xenfr.org
  - http://doc.ubuntu-fr.org/virtualisation

Références (2)

- GNU Linux Magazine / France
  - no 85, 87, 89, 92
- Cours Xen pour les journées Mathrice - Mars 2007 - J. Castang et P. Depouilly
- JRES2005
- Benchmarks de Xen
  - http://www.cl.cam.ac.uk/netos/paper/2003-xensosp.pdf