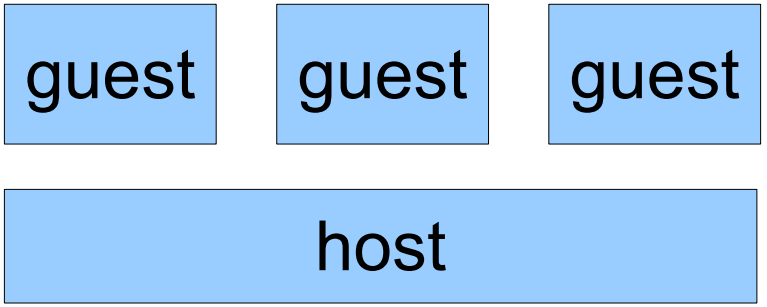
Virtualisation

Qualités recherchées
Solutions existantes
Quelle flexibilité ?

(slides disponibles sur le wiki)

Objectif de la virtualisation

- Une machine physique
  - Hôte (*host*)
- Plusieurs services
  - Invités (*guests*)
- Réduction des coûts
  - Achat de matériel
  - Consommation électrique
  - (Maintenance)



Sur le principe rien de nouveau

- S/370 savait émuler plusieurs S/360 en parallèle
- Processus Unix
- Système de fichiers avec permissions

- Quelle interface est fournie ?
  - Un PC vs un PC simplifié vs des hypercalls vs des appels systèmes

Des tonnes de solutions

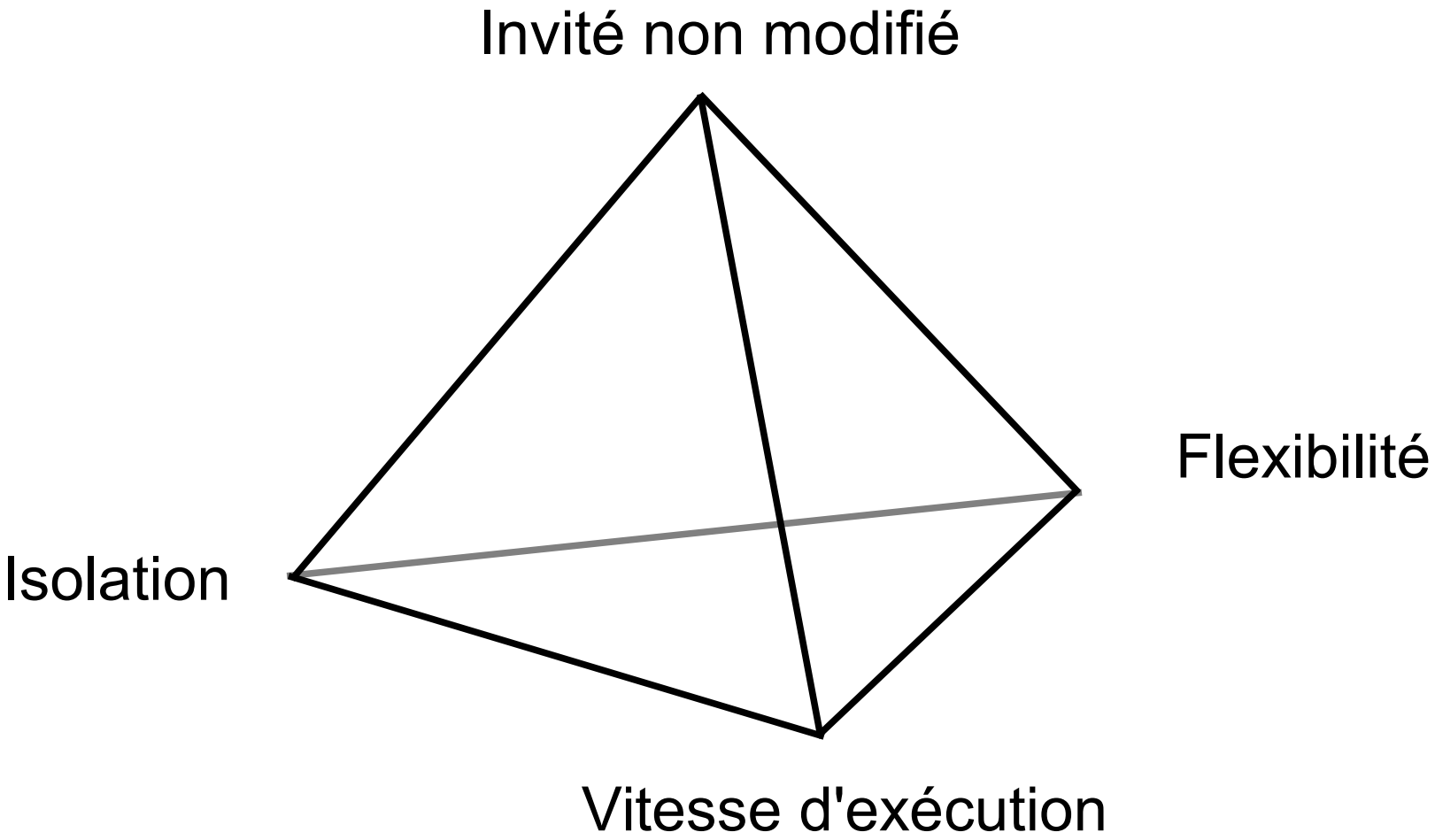
- Virtualbox
- KVM/qemu
- Xen
- VMware
- chroot
- LXC
- Parallels
- OpenVZ
- ...

Quelle est la bonne pour vous ?

Pas de réponse facile

Savoir quelles questions se poser

Des qualités contradictoires



Invité non modifié ?

- Modifications noyau pour s'adapter
  - Exécution en ring 1 plutôt que ring 0
    - Pas d'instruction privilégiée
  - Exécution dans un processus
- Drivers spéciaux pour accélérer

Utiliser une distribution standard ?

- p.e. Debian fournit et supporte une variante Xen

Flexibilité ?

- Mémoire
  - Réglage automatique / statique / à la volée
  - Éviter doublons
- Disque
  - Réglage automatique / statique / à la volée
  - Ajouter à la volée
- Échanger des données
  - Montage de fichiers
- Comptabilité

Isolation ?

Est-ce que les invités

- voient les fichiers des autres ?
- se voient ?
- sont sur le même réseau Ethernet ?
- utilisent le même noyau ?
  - Attention au crash !

Vitesse d'exécution

- CPU pas vraiment un problème
  - Sauf virtualisation totale sans accélération
- Gestion mémoire
  - Création de nombreux processus
  - Basculement entre processus
- Disque, réseau
  - Débit
  - Latence

Solutions existantes
3 grandes classes

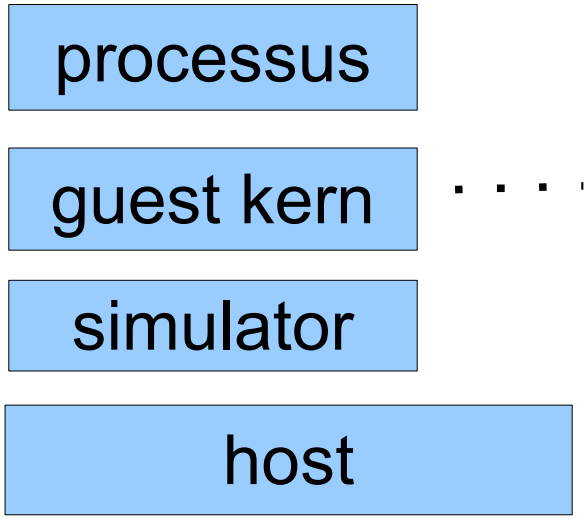
3 grandes classes

- Virtualisation totale
  - KVM/qemu, Xen HVM, VirtualBox, VMware, ...
- Para-virtualisation
  - Xen PV, Xen PV-HVM, drivers VMware, virtio, ...
- Conteneurs
  - Processus, chroot, cgroups, openVZ, LXC, ...

Virtualisation totale

KVM/qemu, xen HVM, VirtualBox, Vmware

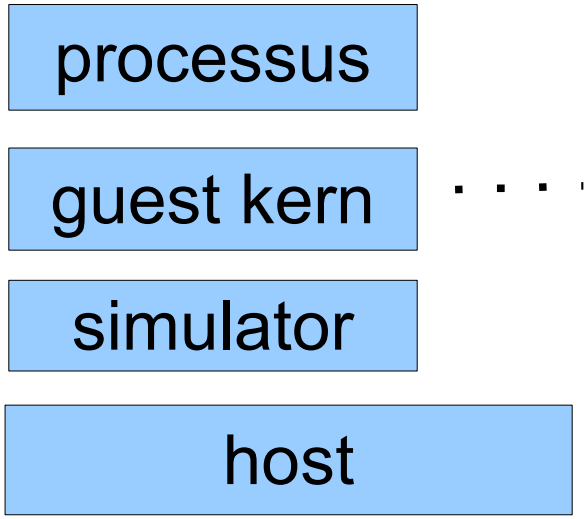
- Simulent un PC complet !
  - BIOS, floppy, ...
  - Ou pas : qboot
- Simulation complexe
  - De nombreux CVE...
- Accélérée par le support matériel
  - Intel (vmx, VT-x, VT-i), AMD (svm), NPT
- Obligatoire pour Windows & co



Virtualisation totale

KVM/qemu, xen HVM, VirtualBox, Vmware

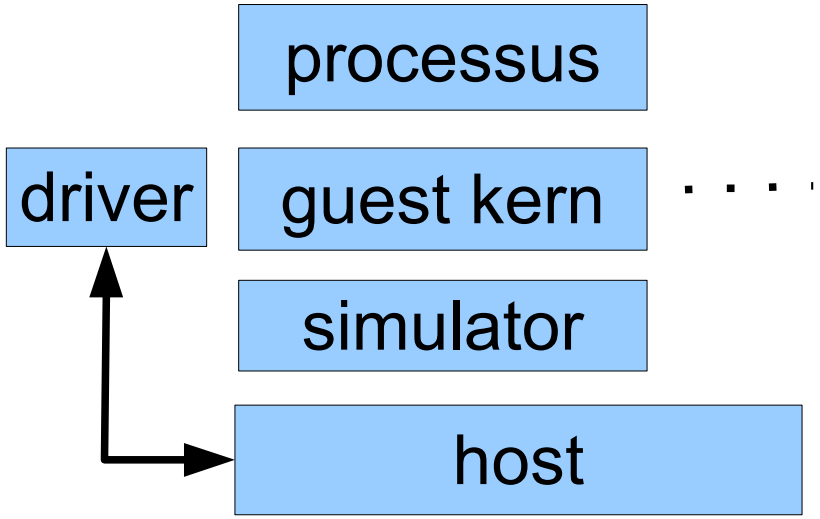
- Invité non modifié
- Pas très flexible
  - Retirer une barrette de RAM ?
- Très isolé
  - Mais attention bugs simulation
    - Utiliser stubdom-dm
- Vitesse raisonnable
  - Grâce au support matériel



Para-Virtualisation

drivers VMware, virtio, PV

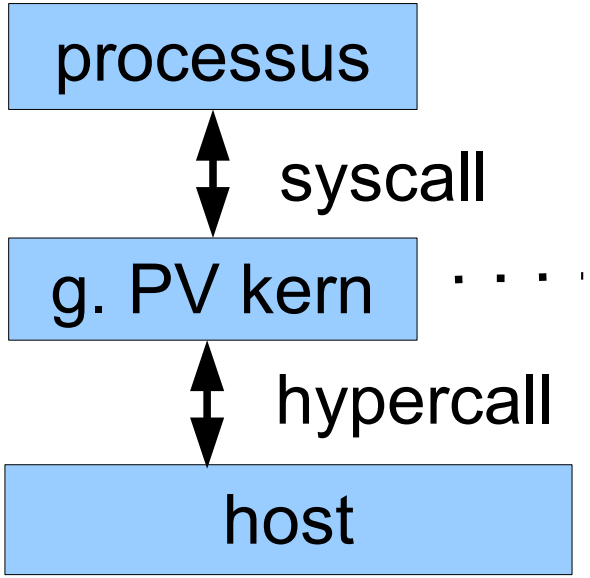
- « drivers VMware »
- Souvent basés sur de la mémoire partagée
  - frontend/backend
- Vitesse, flexibilité



Para-Virtualisation

Xen PV, Xen PV-HVM

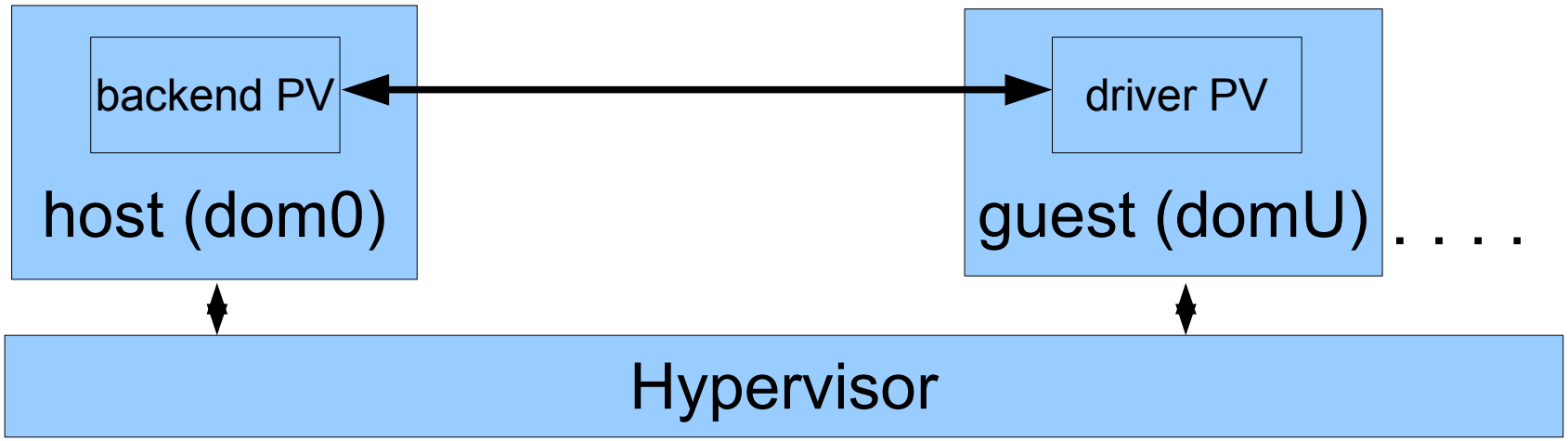
- Kernel PV
  - Sait vraiment qu'il est virtualisé
- « Hypercalls »
  - Essentiellement les instructions privilégiées
    - Contrôle CPU, mémoire virtuelle, ...
- Drivers PV



Para-Virtualisation en vrai chez Xen

Virtualisation de type 1

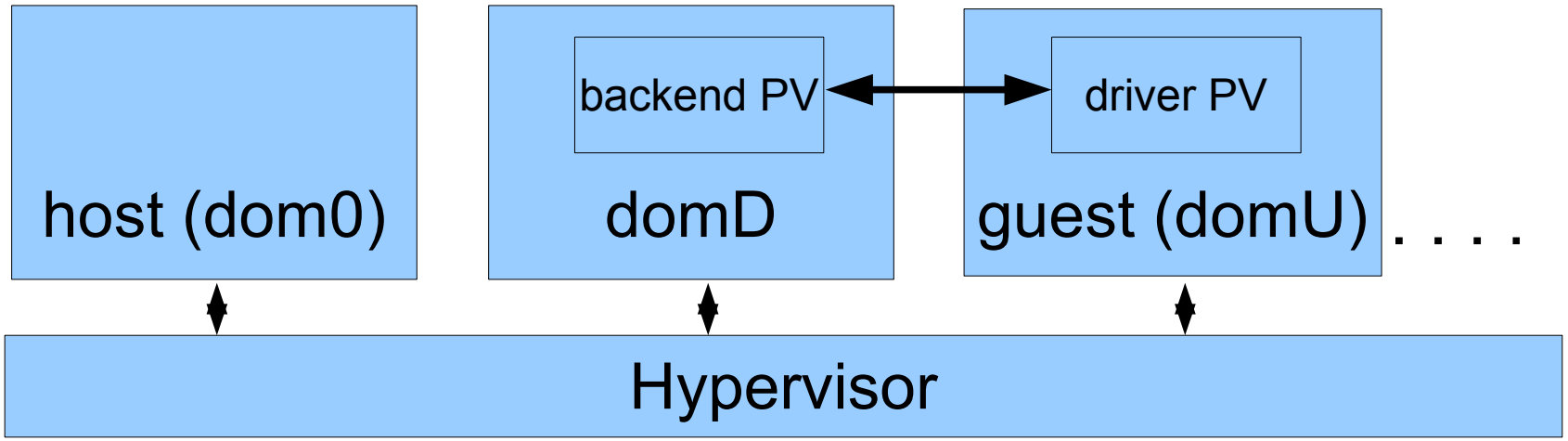
- Un hyperviseur boote avant l'hôte



Para-Virtualisation en vrai chez Xen

Virtualisation de type 1

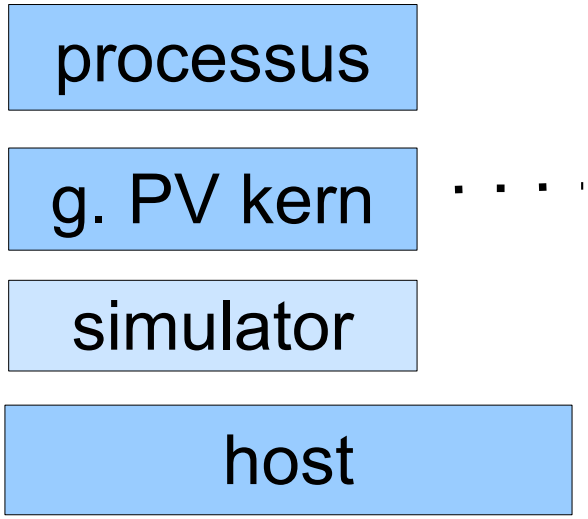
- Un hyperviseur boote avant l'hôte
- Voire désagrégation



Para-Virtualisation

Xen PV, Xen PV-HVM, drivers VMware, virtio, UML

- Invité modifié
  - Profondément (PV), ou drivers
- Très flexible
  - Redimensionnement à la volée
- Très isolé
  - Noyaux séparés
- Vitesse excellente
  - C'est fait pour !



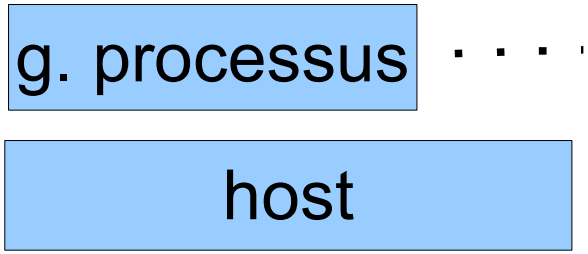
Conteneurs

Processus, chroot, cgroups/NS, openVZ, LXC

- Processus++

Isolation variable

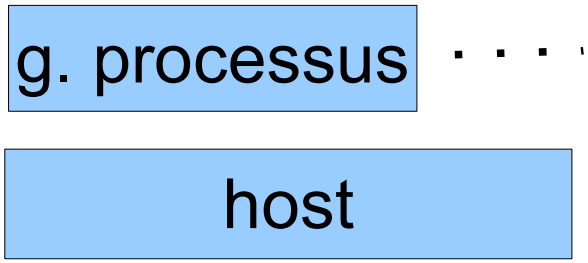
- Espace de fichiers
  - « / privé »
- Espace de pids
- Espace réseau
- …
- oublis ?



Conteneurs

Processus, chroot, cgroups/NS, openVZ, LXC

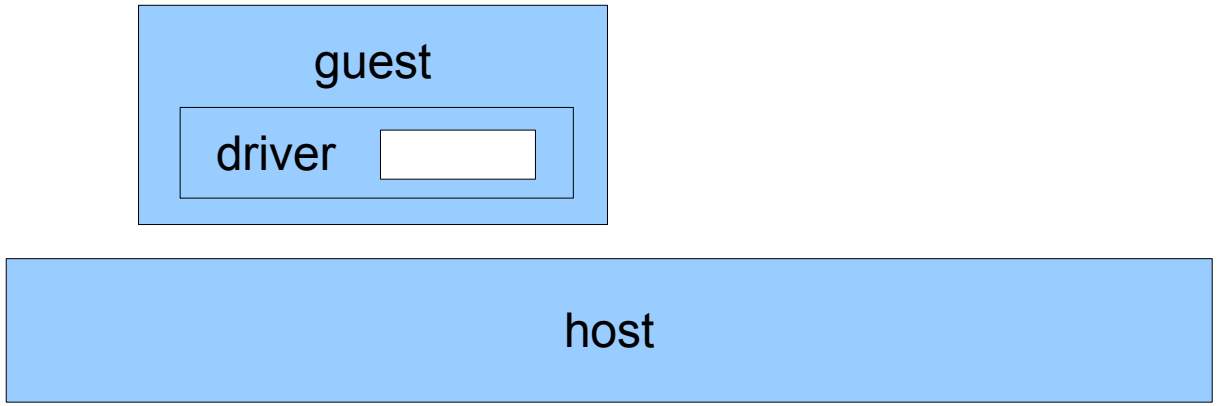
- Invité non modifié
  - Processus normal
- Très flexible
- Plus ou moins isolé
  - Selon support
  - Même noyau, problème incompatibilités (udev)
- Vitesse parfaite
  - Identique à un processus



Flexibilité ?

Flexibilité mémoire

- Ajouter de la mémoire
  - Pas « trop » dur
- Enlever de la mémoire
  - Oulaaa…
  - Ballooning



Flexibilité disque

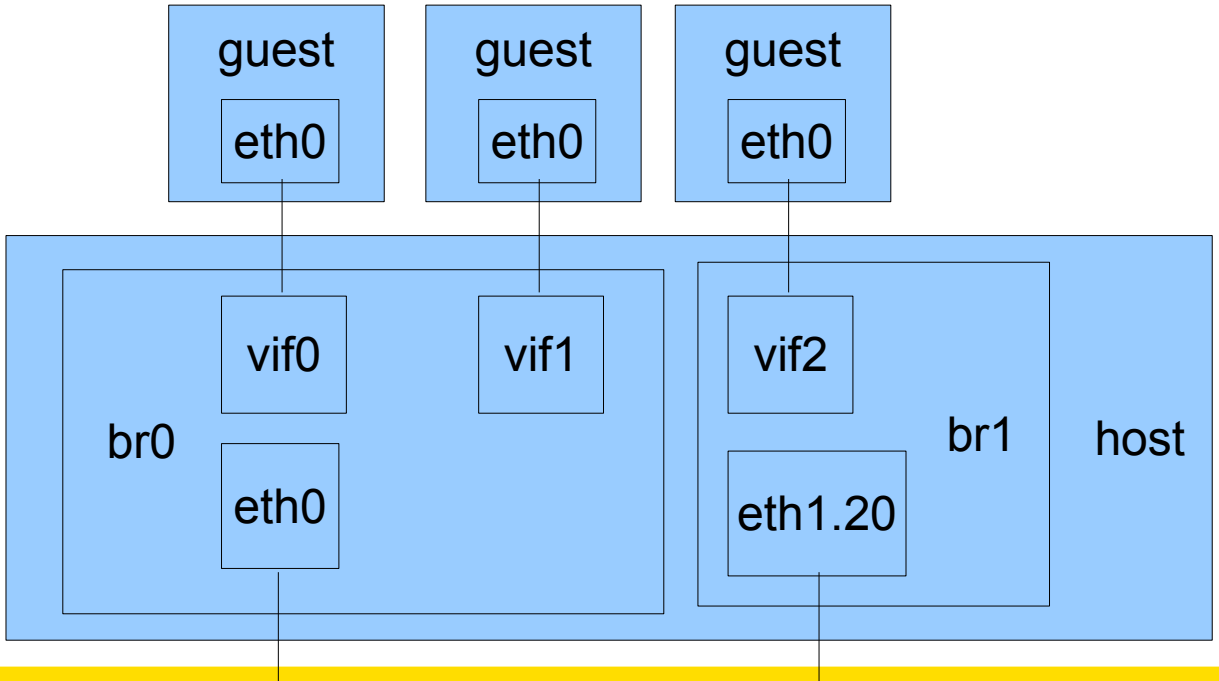
Retailler à la volée

- Conteneurs : ~= quotas
- Utiliser lvm sur l'hôte
- Utiliser resize2fs sur l'invité
- Sans douleur !

Flexibilité réseau

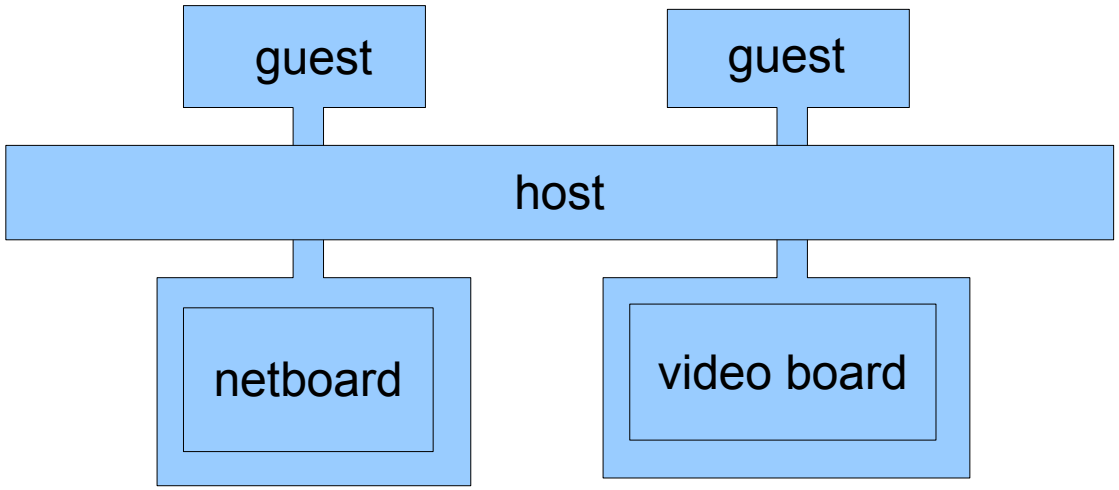
Branchements virtuels

- Bridge ~= switch réseau
- Gestion habituelle hôte (bridge, VLAN, ...)
  - Ou openvswitch



Flexibilité matériel

- PCI passthrough
- USB passthrough
- ! Sécurité ! (DMA, ...)
  - Support VT-d (IOMMU)



Flexibilité des VMs

Pouvoir éteindre la machine physique

- sans interruption de service

Migration de VM

- À froid
- À chaud (live)

La difficulté : le disque

- drbd, iscsi, NFS, ...

Outils de gestion

- Clickodrome ou ligne de commande ?
- Bibliothèques de gestion
  - libvirt
- Outils de gestion
  - http://blog.circleci.com/its-the-future/
  - Aujourd'hui, on a parlé de Vagrant, et on va parler de docker

Conclusion

- Toutes essaient d'optimiser les 4 qualités
  - Plus ou moins de succès
  - Évoluent en permanence
    - Ne pas croire ce que les gens disent !
  - Savoir ce qu'on veut
- Des outils par-dessus pour homogénéiser