

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
 - Économie de matériel
 - Consommation électrique
 - (+ maintenance)



Sur le principe rien de nouveau

- S- . / 0 savait éxécuter plusieurs S- . 10 en parallèle
- %processus Unix
- S!st2 gère de fichiers avec permissions
- Quelle interface est fournie ?
 - Un %* vs un %* simplifié vs des h!percalls vs des appels s!st2 es

3es tonnes de solutions

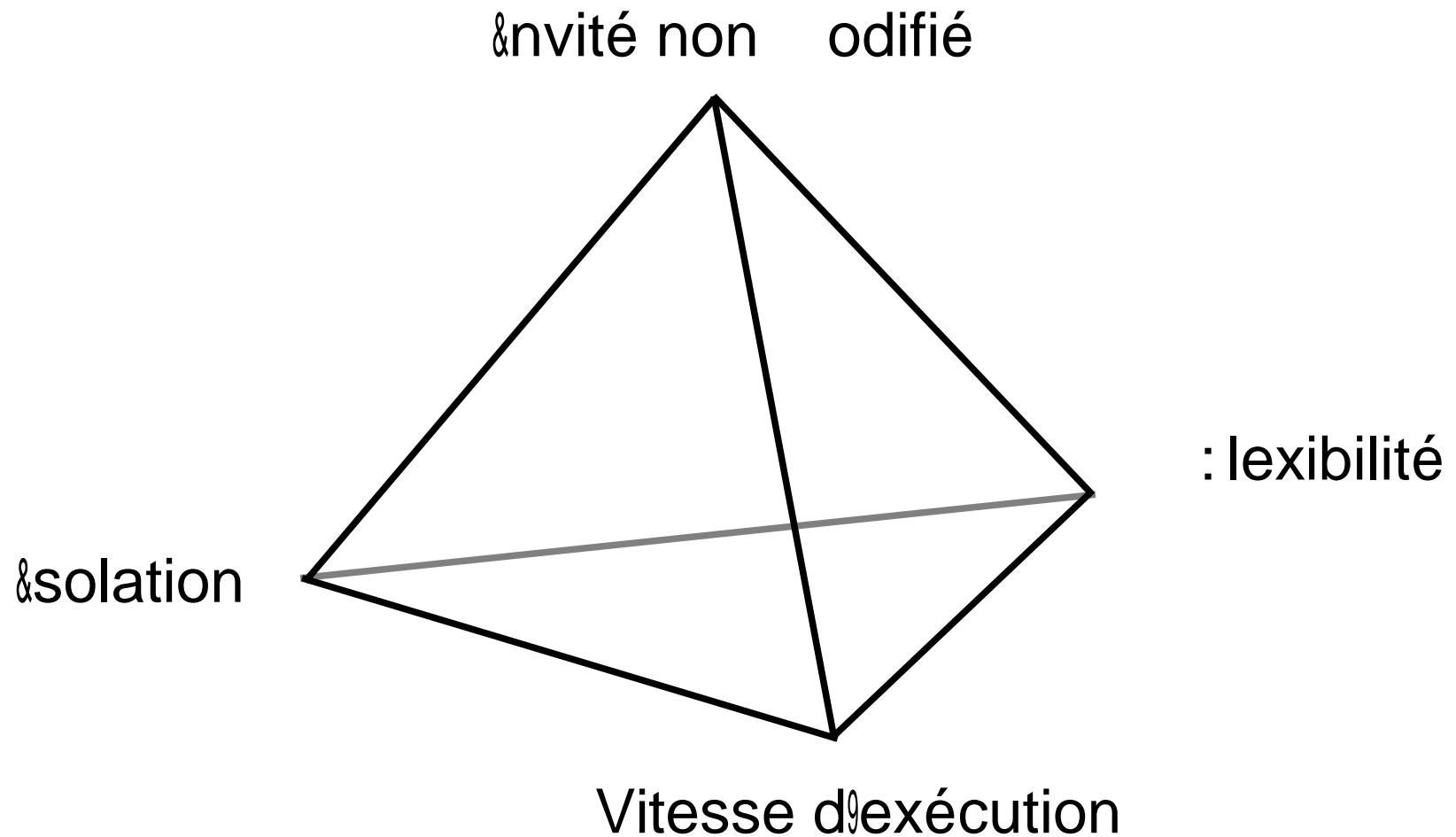
- Virtualbox
- 4V+-"e u
- 5en
- V+ware
- chroot
- 65 *
- %arallels
- OpenV7
- 888

Quelle est la bonne pour vous ?

%as de réponse facile

Savoir "uelles "uestions se poser

3es "ualités contradictoires



Invité non modifié ?

- + modifications no! au pour s'adapter
 - ; exécution en rin , < plus\$ " ue rin , 0
 - %as d'instruction privilégiée
 - ; exécution dans un processus
- 3 rivières spéciales pour accélérer

Utiliser une distribution standard ?

- p8e8 3ebian fournit et supporte une variante
5en

: lexibilité ?

- + é oire
 - ' é , la , e auto ati" ue - stati" ue - = la volée
 - >viter doublons
- 3is" ue
 - ' é , la , e auto ati" ue - stati" ue - = la volée
 -)jouer = la volée
- >chan , er des données
 - +onta , e de fichiers
- * o ptabilité

Isolation ?

; st?ce "ue les invités

- voient les fichiers des autres ?
- se voient ?
- sont sur le @ e réseau ; thernet ?
- utilisent le @ e no!au ?
 -) ttention au crash A

Vitesse d'exécution

- * %U pas vraiment un problème
 - Sauf virtualisation totale sans accélération
- Gestion éolée
 - * création de nombreux processus
 - Cascade entre processus
- 3is "ue" réseau
 - 3ébit
 - 6atence

Solutions existantes
. , randes classes

. , randes classes

- Virtualisation totale

- 4V+-"e u) 5en #V+D VirtualCoxD V+wareD 888

- %ara?virtualisation

- 5en %VD 5en %V?#V+D drivers V+wareD virtioD 888

- *onteneurs

- %rocessusD chrootD c , roupsD openV7D 65 * D 888

Virtualisation totale

4V+-"e uD xen #V+D VirtualCoxD V ware

- Si ulent un %* co plet A

- C&OSD flopp!D 888

- Ou pas E "boot

- Si ulation co plexe

- 3e no breux *V ; 888

-) ccélérée par le support atériel

- &intel (v xD VF?xD VF?i)D) + 3 (sv)D G%F

- Obli ,atoire pour Hindows I co

processus

, uest kern

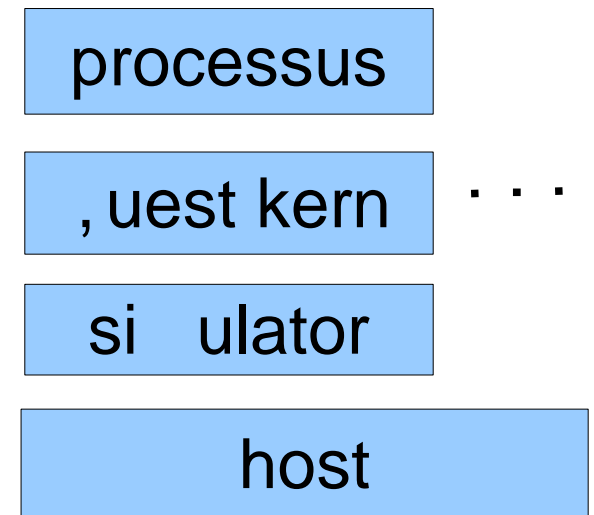
si ulator

host

Virtualisation totale

4V+-"e uD xen #V+D VirtualCoxD V ware

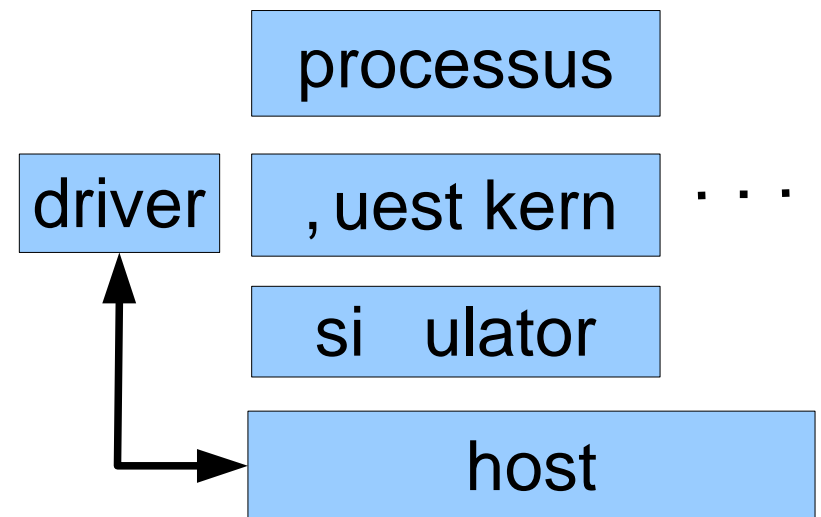
- invité non modifié
- pas très flexible
 - étirer une barrette de mémoire ?
- Très isolé
 - faire attention à la simulation
 - Utiliser stubs
- Vitesse raisonnable
 - Besoin au support matériel



Virtualisation

drivers V + ware virtio %V

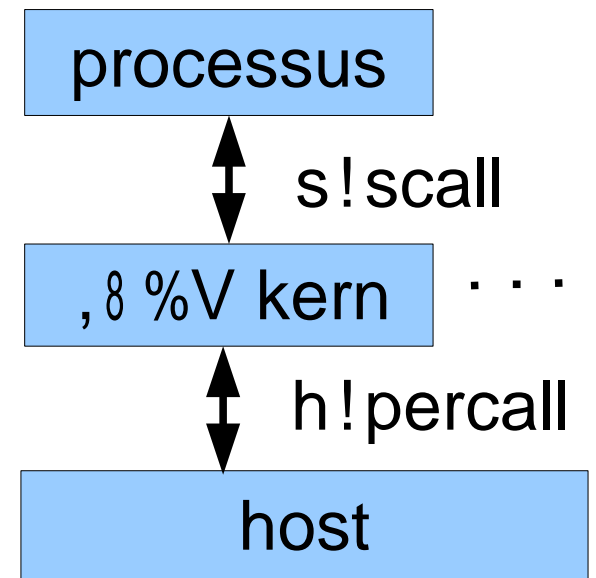
- K drivers V + ware L
- Souvent basés sur de la technologie partagée
 - frontend-backend
- Vitesse et flexibilité



Virtualisation

5 en %VD 5 en %V?#V +

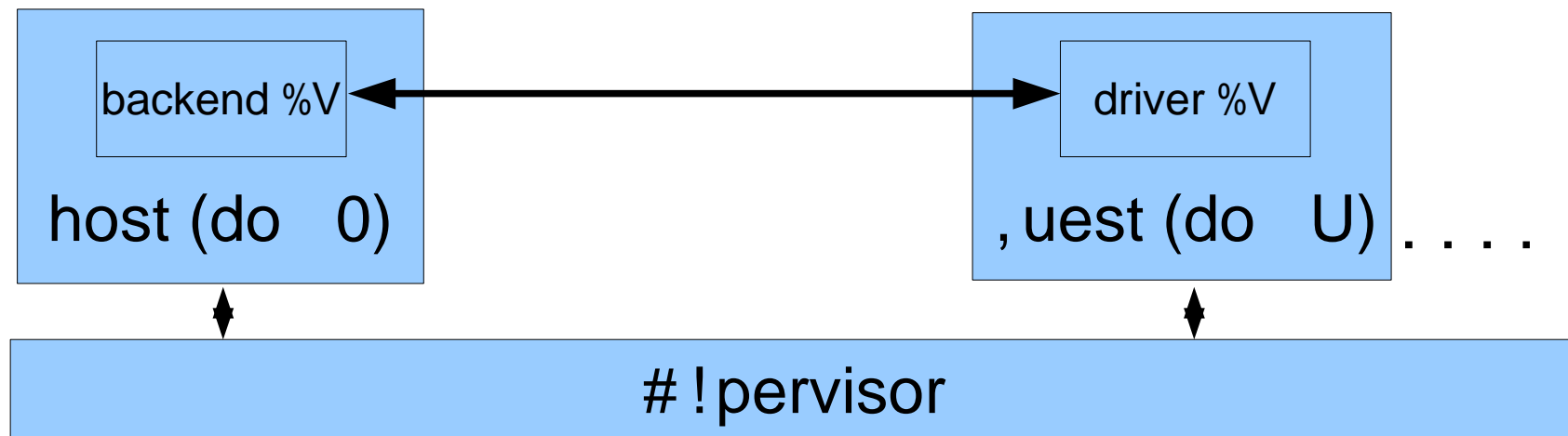
- kernel %V
 - Sait vraiment "un" est virtualisé
- K # !percalls L
 - ; essentielle ent les instructions privilégiées
 - * ontr\$le * %UD é oire virtuelleD 888
- 3 rivières %V



Virtualisation en vrai chem 5en

Virtualisation de t!pe <

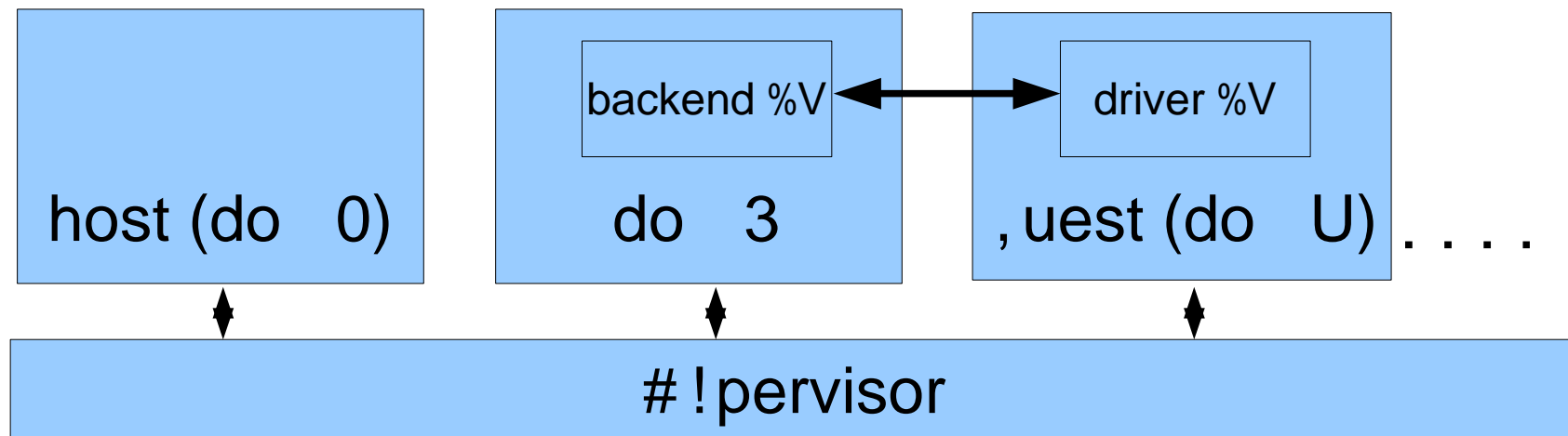
- Un h!perviseur boote avant l!h\$te



Virtualisation en vrai chem 5 en

Virtualisation de t!pe <

- Un h!perviseur boote avant l!h\$te
- Voire désa , ré , ation



Virtualisation

5 en %VD 5 en %V?#V +D drivers V + wareD virtioD
U+6

- invité odifié

- profondément (%V)D ou drivers

- Fr2s flexible

- 'éditions = la volée

- Fr2s isolé

- Go!aux séparés

- Vitesse excellente

- *est fait pour A

processus

, 8 %V kern . . .

si ulator

host

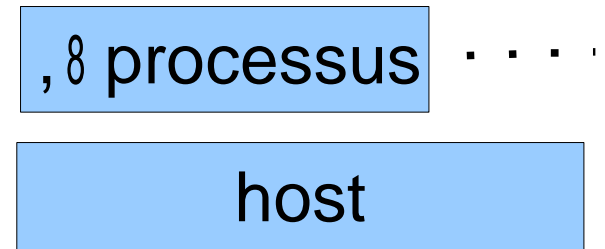
* onteneurs

%rocessusD chrootD c , rroups-GSD openV7D 65 *

- %rocessusNN

&solution variable

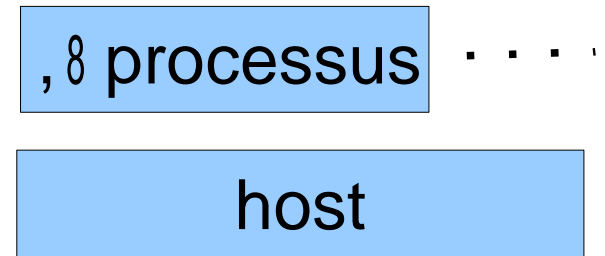
- ; space de fichiers
 - K - privé L
- ; space de pids
- ; space réseau
- O
- oublis ?



* onteneurs

%rocessus chroot c , rroups-GSD openV7 65 *

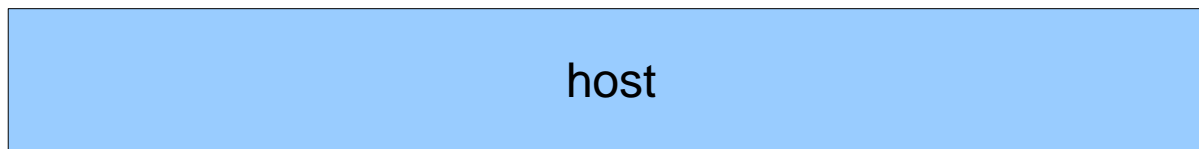
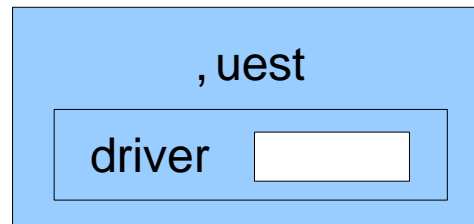
- invité non odifié
 - %rocessus normal
- Fr2s flexible
- %lus ou oins isolé
 - Selon support
 - +@ e no!au probl2 e inco patibilités (udev)
- Vitesse parfaite
 - &dentit" ue = un processus



: lexibilité ?

: lexibilité é oire

-)jouter de la é oire
 - %as K trop L dur
- ;nlever de la é oire
 - OulaaaO
 - Calloonin ,



: lexibilité dis " ue

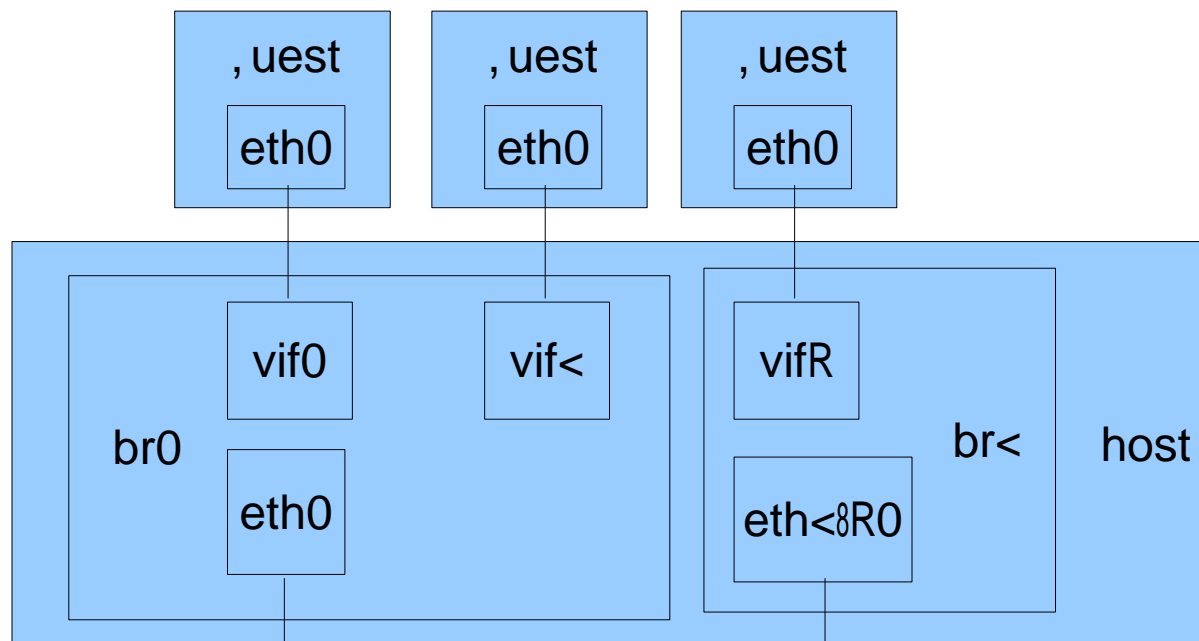
' etailer = la volée

- * onteneurs E PQ " uotas
- Utiliser Iv sur l9h\$te
- Utiliser resiMeRfs sur l9invité
- Sans douleur A

: flexibilité réseau

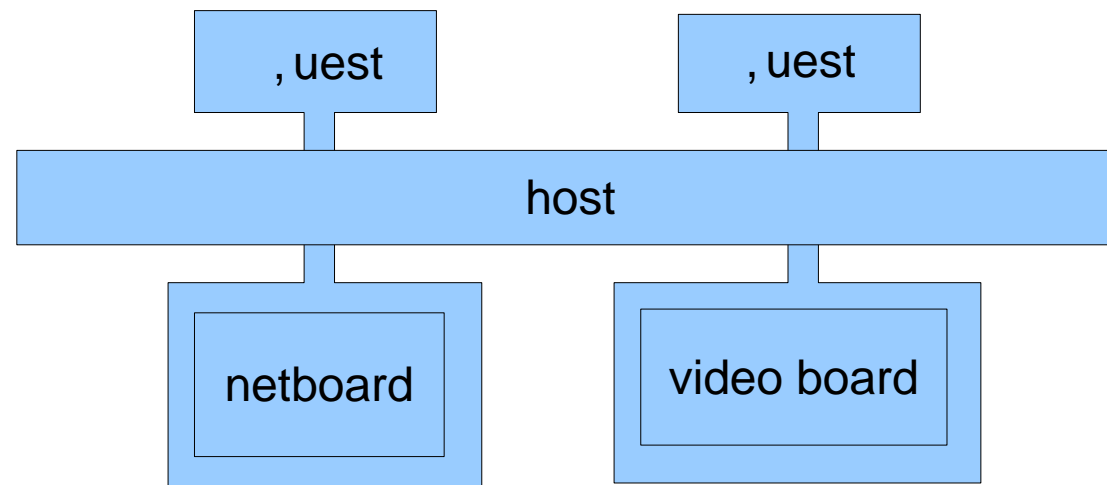
Création de machines virtuelles

- Création de machines virtuelles
- Gestion habituelle d'une machine (bridge, ethernet, VLAN, etc.)
- Ou openvswitch



: flexibilité matériel

- % * & passthrou , h
- USC passthrou , h
- A Sécurité A (3 +) D 888)
 - Support VF?d (&O + + U)



: lexibilité des V + s

%ouvoir éteindre la machine physique

– sans interruption de service

+ i, ration de V +

- S froid
- S chaud (live)

6a difficulté E le dis " ue

- drbd iscsi G : SD 888

Outils de gestion

- * lickodro e ou li, ne de co ande ?
- 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

- Foutes essaient d'optimiser les T "ualités
 - %lus ou moins de succès
 - >volent en permanence
 - Ge pas croire ce "ue les ,ens disent A
 - Savoir ce "u'on veut
- 3es outils par-dessus pour ho o, énéiser