Gestion de parc *Windows* depuis *Unix*

Pascal Cabaud & Laurent Joly

Problématiques du gestionnaire de parc

- matériel hétérogène
- logiciels incompatibles
- déploiement de nouveaux systèmes
- lutte anti-virale
- logiciels nécessitant des droits « administrateur »

Environnement

- CRL : ~100 postes pédagogiques
- UFR EILA : ~120 postes pédagogiques
- Nombreuses applications *Windows*
- *GNU/Linux* pour certains cours (*dual-boot*)

et des informaticiens plus agiles avec un terminal *Unix* qu'un écran *Windows*...

Environnement (suite)

Sur chaque poste étudiant, *GNU/Linux* avec :

- Un serveur *Apache* (UFR EILA)
- Un serveur *OpenSSH*
- *Sun xVM VirtualBox*

VirtualBox sert à lancer un système *Windows* par poste.

Plan

- Virtualisation
- Déploiement d'une machine virtuelle
- Sécurisation de la virtualisation
- « Autonomisation » des utilisateurs
- *Windows* depuis *Unix*
- Virus et anti-virus
- Conclusion

Virtualisation (1/3)

Un système virtualisé ne sait quasiment rien du système hôte : « matériel » homogène.

Deux machines virtuelles peuvent chacune héberger deux logiciels incompatibles entre eux (deux versions différentes d'un même logiciel).

Si un logiciel vérifie l'adresse MAC, on peut lancer N systèmes avec la même adresse physique : il suffit d'utiliser le NAT.

Virtualisation (2/3)

Une machine virtuelle *VirtualBox*, c'est :

- Un disque virtuel (un gros fichier de plusieurs giga octets)
- Un fichier XML (définition de la machine virtuelle : nom, processeur, carte réseau, adresse MAC, quantité de RAM…)

Virtualisation (3/3)

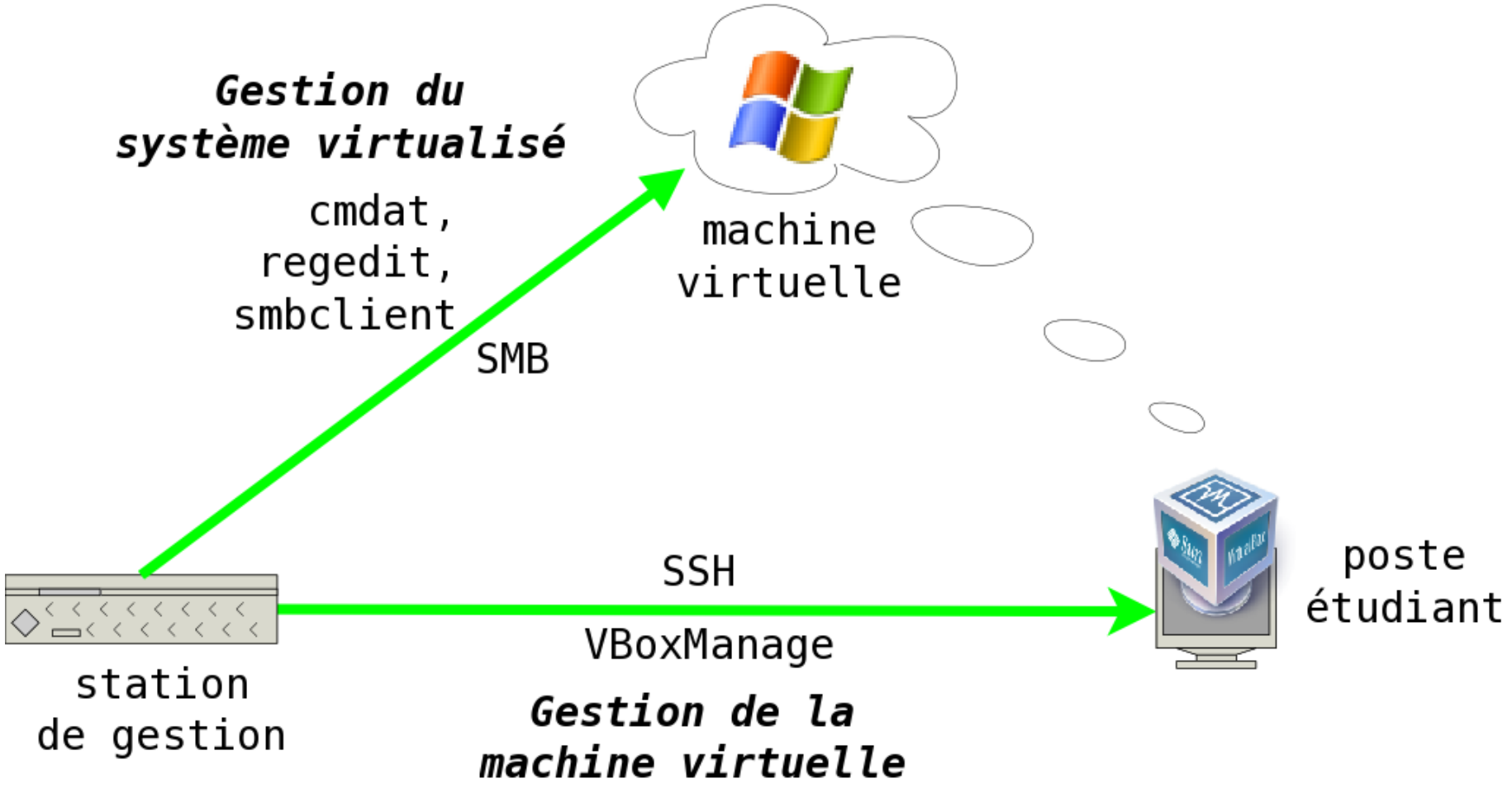
Déployer une machine virtuelle *VirtualBox*, c'est :

- copier le disque virtuel
- recréer le fichier XML

Déploiement d'une machine virtuelle

- *VirtualBox* se gère intégralement *via* la commande `VBoxManage`.
- Pour lancer des commandes à distance, `ssh` avec authentification par clef.
- Pour envoyer la même commande sur N machines, *ClusterIt* et en particulier `dsh` et `pcp`.
- Ces derniers appellent `ssh` et `scp`.

Déploiement d'une machine virtuelle (suite)



Sécurisation de la virtualisation (1/3)

- Les différentes solutions de virtualisation passent par un module noyau.
- Un utilisateur peut potentiellement lancer sa propre machine virtuelle sur laquelle il est *root*.

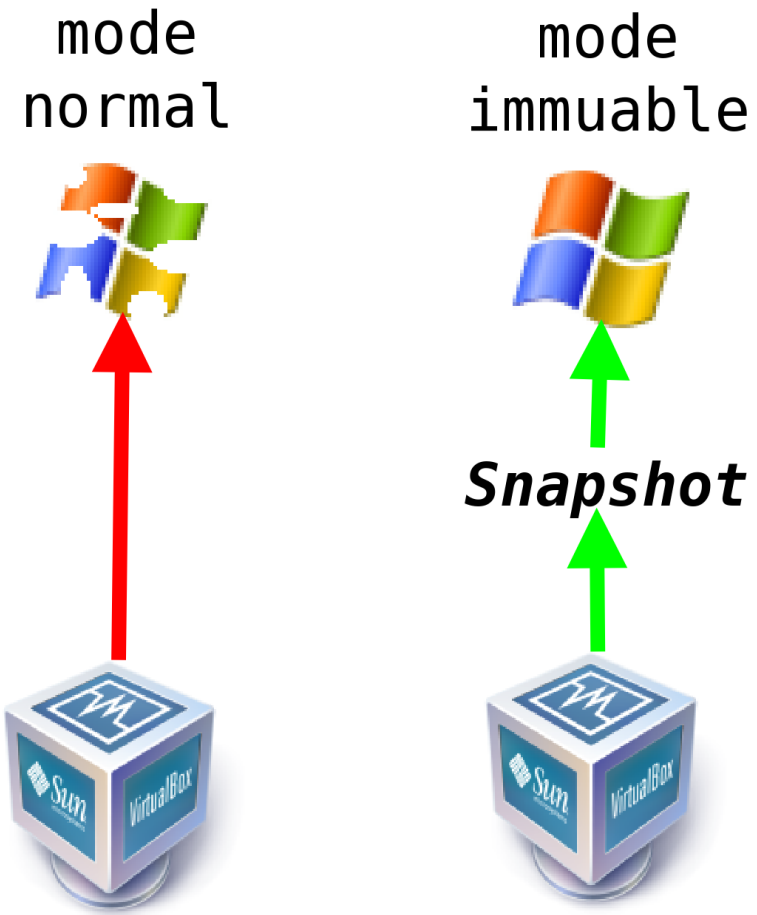
Sécurisation de la virtualisation (2/3)

- Nos machines tournent sous l'identité `vbox`.
- Un *daemon* les lance et les arrête : `vboxd`.
- Il accepte quelques commandes :
  - lister les machines virtuelles,
  - lancer / arrêter une machine virtuelle
  - afficher la machine en fonctionnement

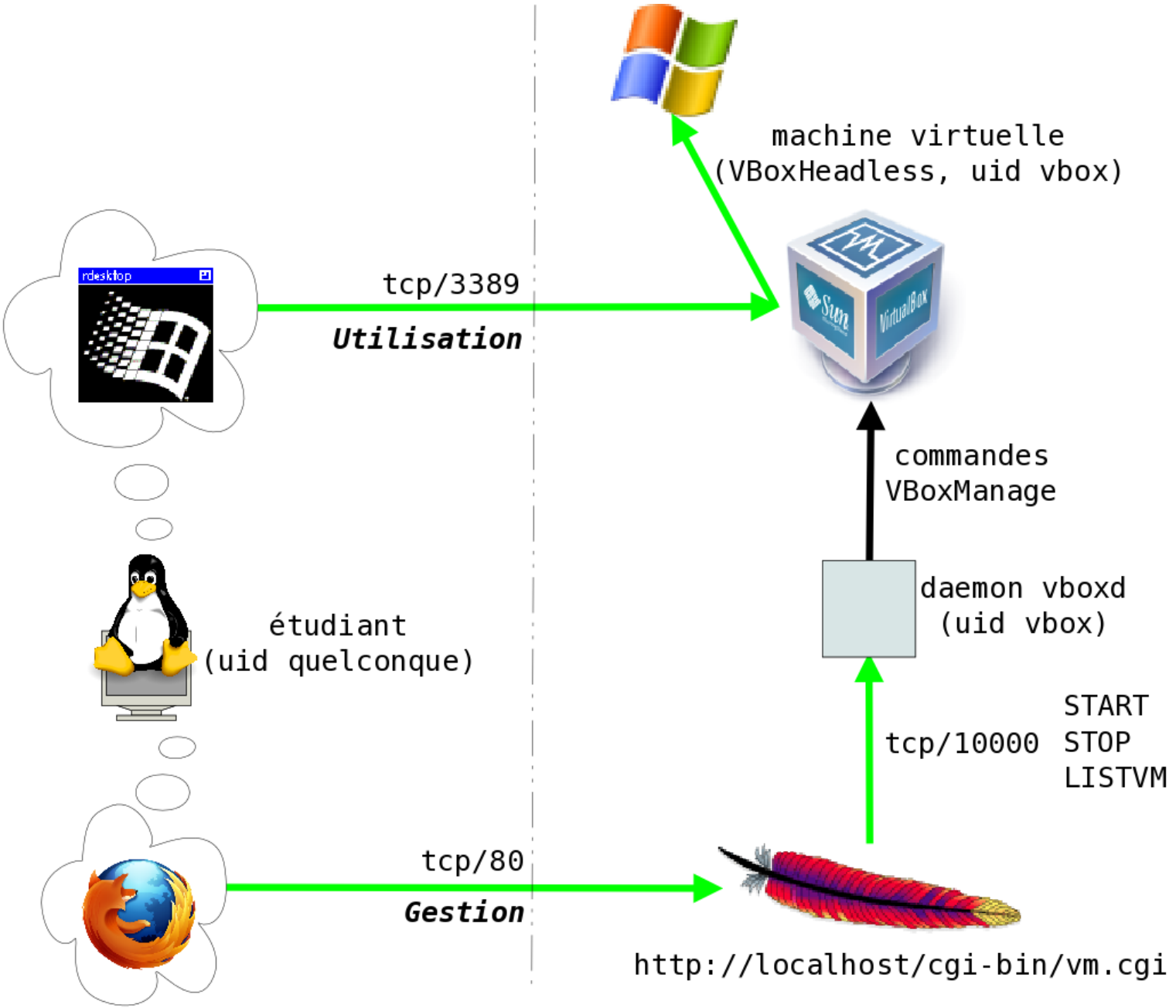
Sécurisation de la virtualisation (3/3)

- Les machines virtuelles sont lancées sans écran (*headless*).
- On accède à une machine virtuelle en se connectant à *VirtualBox* en RDP sur `localhost:3389`.

Réinitialisation : le mode immuable



Autonomisation des utilisateurs



Windows depuis *Unix* (1/4)

Besoin d'un domaine *Windows* ?

Parmi les outils fournis avec *Samba*, on trouve :

- `smbclient` pour gérer les fichiers à distance
- `smbcacls` pour gérer les ACL sur les fichiers
- `rpcclient` pour envoyer quelques commandes (dont en particulier *reboot*)

Windows depuis *Unix* (2/4)

- *Samba-TNG* est un *fork* de *Samba* et reste à un stade expérimental.
- Parmi les outils fournis avec *Samba-TNG*, on trouve :
  - `regedit` pour accéder au registre à distance
  - `cmdat` pour interagir avec `at.exe`

Windows depuis *Unix* (3/4)

Après déploiement d'une machine virtuelle sur des postes étudiants :

1. Changement du nom de chaque système :
   - on modifie 6 clefs registre (à distance *via* `regedit` ou par un script `.bat` appelé par `cmdat`)
   - on reboote avec `rpcclient`

Windows depuis *Unix* (4/4)

2. Connexion au domaine *Samba* :
   - copier un script `netdom.bat` dans `C:\sysadmin`
   - indiquer à `at.exe` de lancer ce script dans la minute qui suit (le script fait appel à `netdom.exe`, commande des *Support Tools*)
3. Vérification :
   - `nmblookup` affiche le nom et le domaine

Virus et anti-virus (1/4)

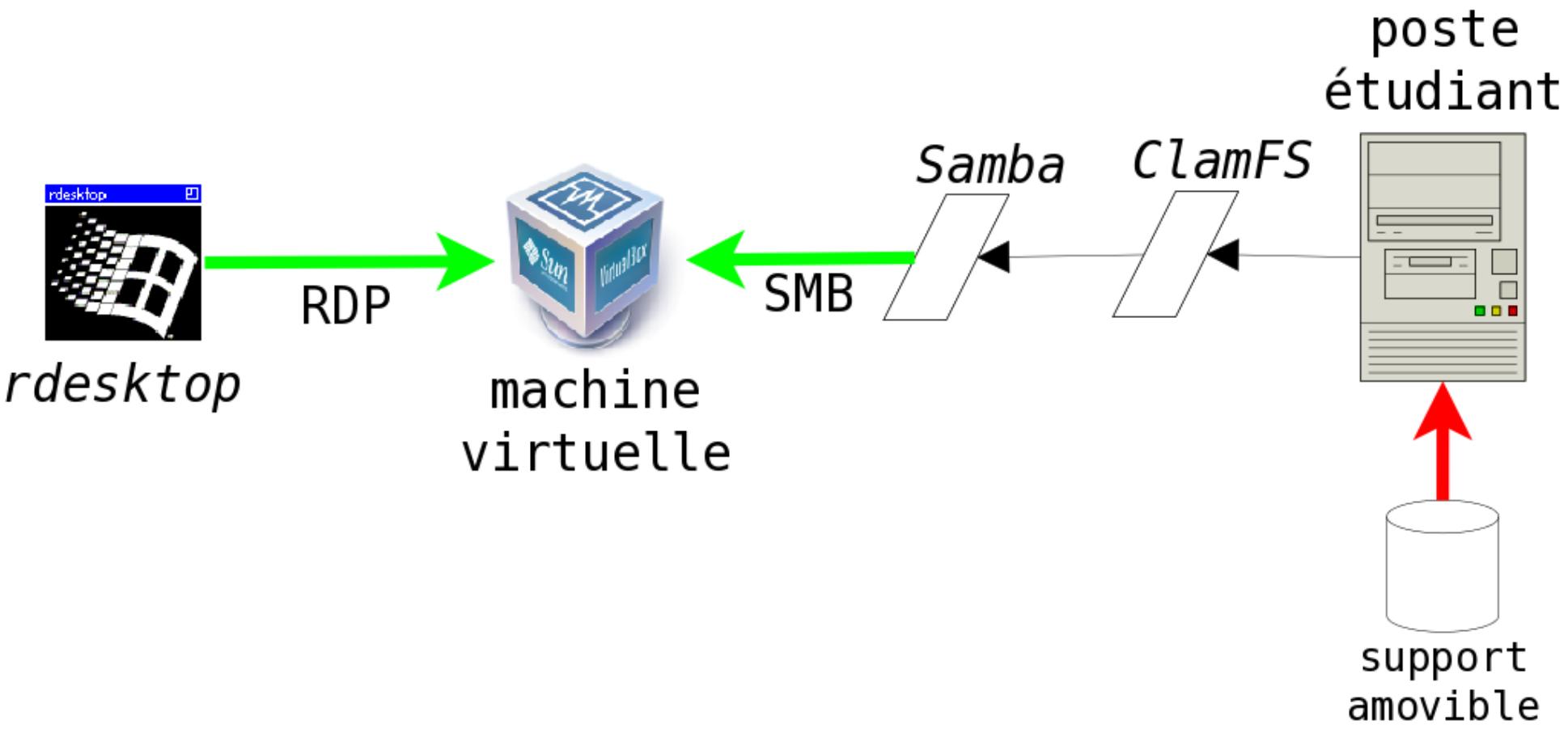
Besoin d'accéder à un support amovible ?

Constats : notre anti-virus sous *Windows*

- consomme des ressources (CPU, accès disques)
- est peu performant et peu efficace
- a une console de gestion et remonte les alertes

Première option : déporter l'anti-virus sur *GNU/Linux*.

Virus et anti-virus (2/4)



Virus et anti-virus (3/4)

Nécessite un filtrage réseau efficace (*Squid* bloque les fichiers `.exe`, `.dll`, `.scr`, `.bat`, ...)

Variante : monter très régulièrement (sur le système hôte *GNU/Linux*) les disques virtuels et les analyser avec *ClamAV*...

Virus et anti-virus (4/4)

Seconde option : mettre un anti-virus moins lourd en CPU.

Microsoft Security Essentials semble répondre à nos besoins. Reste à savoir s'il est efficace…

Troisième option : pas d'anti-virus, les clefs USB restent sous *GNU/Linux* (répertoire partagé hôte / VM) avec les disques des VMs en mode immuable.

Ces options ne sont pas incompatibles entre elles.

Solution *admin-friendly*

- facilement automatisable par des scripts
- scripts *shell* pour la plupart et lancés avec `dsh` (*ClusterIt* et *OpenSSH*)
- quelques `.bat` copiés avec `smbclient` et lancés avec `cmdat` (*Samba-TNG*)
- déploiement rapide, éventuellement pendant un cours
- choix du système hôte : *GNU/Linux*, (*Open*)*Solaris* et depuis peu *FreeBSD*

Solution *user-friendly*

- plus de double-*boot*
- des performances nettement améliorées à l'ouverture de session (suppression de *Windows Steady State*)
- des performances globalement meilleures (anti-virus)
- choix de la machine virtuelle simple

Questions ?

Merci !

Pascal Cabaud,
http://www.eila.univ-paris-diderot.fr/~pc

Laurent Joly,
http://www.eila.univ-paris-diderot.fr/~laurent