FAQ Netkit

© 2011-2018 tv <tvaira@free.fr> - v.1.1 - produit le 8 janvier 2018

Table des matières

Préambule

Mise en situation

1. **Solution n°1** : Vous devez disposer d'un PC possédant une distribution Linux (sur une partition spécifique, sur une clé USB bootable, sur un Live CD ou encore à l'aide d'un logiciel de virtualisation du type *VMware* ou *VirtualBox*). Le logiciel de virtualisation **Netkit** doit être installé sur la machine Linux ainsi que le programme `uml_dump` qui permet d'utiliser `wireshark` plus facilement. Évidemment, le logiciel `wireshark` doit être installé sur votre système.

   - **Site de NetKit :** www.netkit.org
   - **Site pour `uml_dump` :** kartoch.msi.unilim.fr

2. **Solution n°2** : utilisez un **Live CD/DVD/USB Netkit** "prêt à l'emploi". Vous pouvez aussi utiliser l'image ISO de ces versions Live à l'aide d'un logiciel de virtualisation du type *VMware* ou *VirtualBox*.

On utilisera la **solution n°2** : récupérer l'ISO `LiveCD-Raizo-tv-v3.iso` sur le serveur de la section. On vous rappelle qu'il y a un quota de taille limite pour vos sessions. Il est donc conseillé de copier l'ISO sur un espace local (`/var/iso`) de votre machine.

```
$ sudo mkdir /var/iso
```

Créer ensuite une machine virtuelle dans *VirtualBox* avec cette ISO. Passer ensuite l'interface réseau de la machine virtuelle en mode "**Accès par pont**" (dans l'onglet "Réseau"). Une fois votre machine virtuelle démarrée, vous pouvez utilisez `ssh` pour vous y connecter et travailler. Cela vous permettra de faire des copier/coller plus facilement pour rédiger vos compte-rendus.

```
$ ssh -X user@192.168.52.xx
```

Installation des TP

Certains TP sont fournis avec une archive du type `tpX.tgz` dans un répertoire `sujets-tp`. Par exemple, il vous faudra faire :

```
host> cd ~
host> tar zxvf sujets-tp/tpX.tgz
host> cd tpX
```

Rappels

- le prompt `host>` indique que la commande doit être tapée dans le terminal de votre machine réelle (par opposition aux terminaux ouverts par les machines virtuelles).
- le prompt `name:~#` représente le terminal de la machine virtuelle name.

Démarrage des machines virtuelles

Démarrer le tp en lançant la commande `lstart` dans le répertoire du lab :

```
host> cd ~/tp/tpX
# lancement en mode séquentiel :
host> lstart -s
# lancement en mode parallèle :
host> lstart -p
```

Remarque : la commande `lstart` fonctionne si vous possédez un fichier `lab.conf` dans le répertoire du lab.

FAQ

Remarque : il est conseillé de visiter le site de Netkit (wiki.netkit.org) et de lire cette introduction (keepin.org/pdf/keepin-netkit.pdf).

Qu'est ce que *User-Mode Linux* (ou UML) ?

C'est une modification du noyau linux afin qu'il tourne au niveau applicatif, comme une simple application. Ainsi il est possible de démarrer un deuxième (voir plusieurs) linux sur son installation Linux classique.

Qu'est ce que Netkit ?

C'est un ensemble de scripts permettant de démarrer et configurer un ensemble de machines UML et de créer un réseau virtuel entre ces machines. On a donc un simulateur réseau léger, ne nécessitant pas de droits root et capable de supporter l'ensemble des protocoles réseaux gérés par Linux.

En résumé :

- Émule des réseaux d'ordinateurs
- Outil de maquettage et de simulation
- Pour comprendre le fonctionnement des protocoles
- Sans avoir à investir dans des équipements
- Logiciels open source (licence GPL)
- Utilise des logiciels libres

Où puis-je télécharger Netkit ?

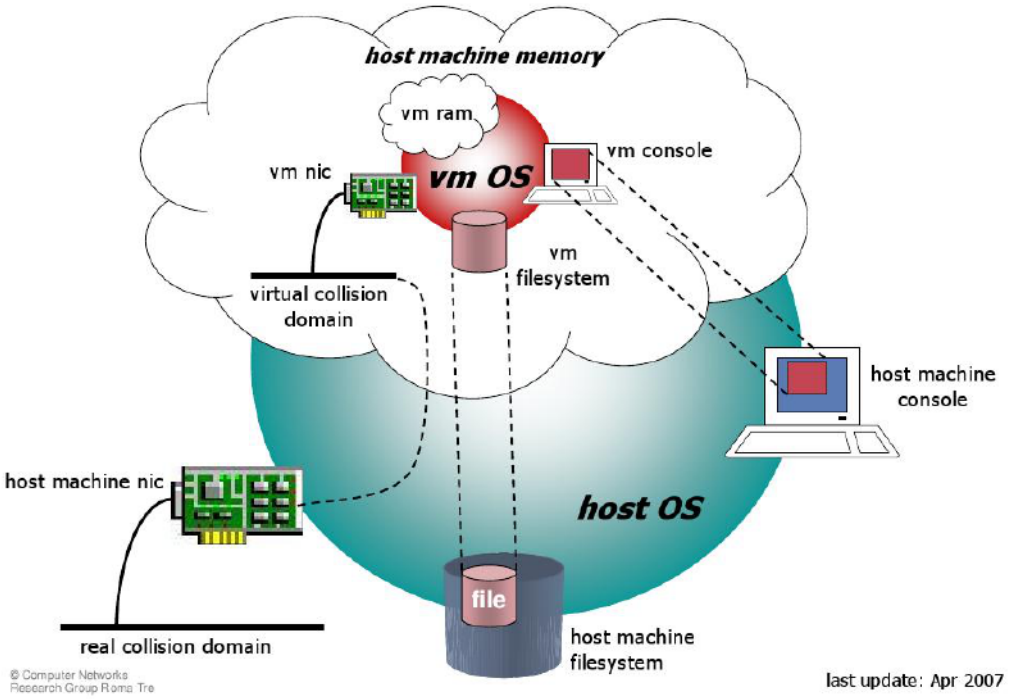
Site de téléchargement : wiki.netkit.org

Les archives de Netkit à récupérer :

- `netkit-X.Y.tar.bz2`
- `netkit-filesystem-FX.Y.tar.bz2`
- `netkit-kernel-KX.Y.tar.bz2`

Que peut-on faire avec une machine virtuelle ?

Les machines virtuelles sont reliées à des domaines de collisions virtuels (un *hub*) et elles peuvent donc communiquer entre-elles. Chaque machine virtuelle peut jouer le rôle de PC, routeur. ou switch.



Qu'est ce qu'un domaine de collision virtuel ?

Les nœuds sont raccordés sur des domaines de collision. Un domaine de collision virtuel peut :

- être connecté à plusieurs interfaces

- chaque interface peut être connectée à un ou plusieurs domaines de collision

Qu'est ce qu'un lab ?

C'est un ensemble de fichier permettant de créer puis de configurer un scénario réseau netkit automatiquement. En général, un scénario est contenu dans un répertoire.

L'organisation basique d'un lab est la suivante :

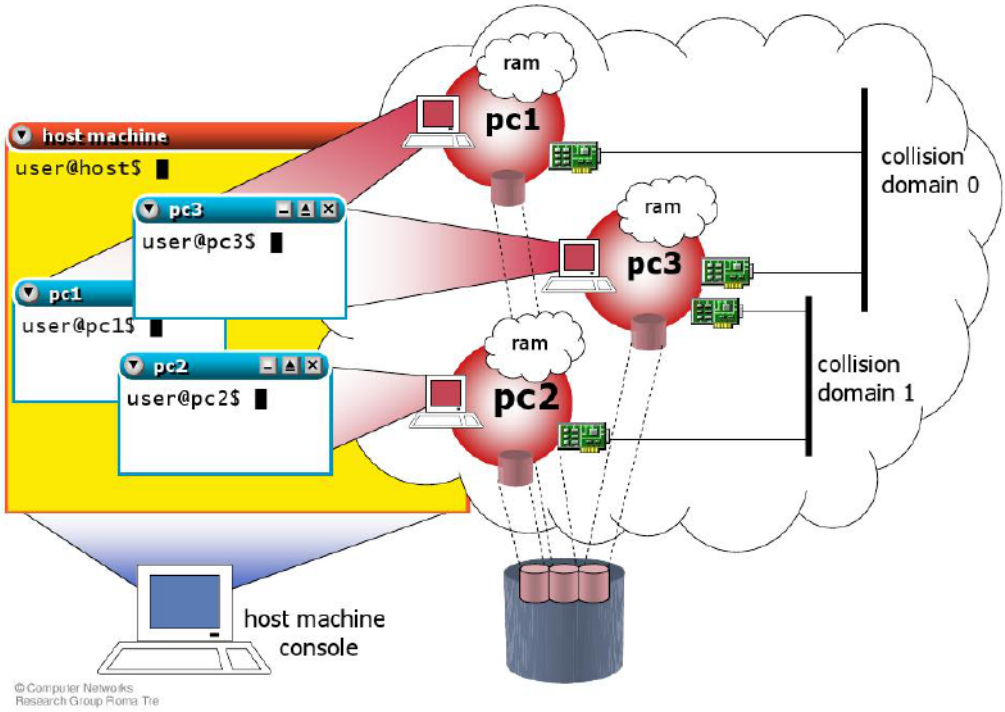
- un fichier `lab.conf` contenant le nom des machines (pc1 et pc2) et la configuration réseau (une domaine de collision A partagé entre l'interface 0 (ou eth0) de la machine pc1 et l'interface 0 (ou eth0) de la machine pc2.

- un fichier `pc1.startup` et un fichier `pc2.startup` contenant les commandes à exécuter respectivement sur les machines pc1 et pc2 après le démarrage de chacune.

- un répertoire `pc1` et un répertoire `pc2` contenant les fichiers à copier sur machine virtuelle après le démarrage de chacune.

Exemple de fichier `lab.conf` :

```
host> cat lab.conf
pc1[0]="A"  # eth0 de pc1 est connecté au domaine A (0 sur la figure)
pc2[0]="B"
pc3[0]="A"
pc3[1]="B"  # eth1 de pc3 est connecté au domaine B (1 sur la figure)
```



Comment démarrer les machines ou le lab ?

Il faut taper :

- `vstart pc1` dans le repertoire de votre lab, ...

- soit taper `lstart -s` (séquentiel) ou `lstart -p` (parallèle) pour démarrer l'ensemble des machines de votre lab.

On peut démarrer manuellement une machine : `vstart pc1 --eth0=A`.

Netkit fournit deux groupes de commandes :

- les `vcommandes`, préfixées par '**v**'

- les `lcommandes`, préfixées par '**l**'

De manière générale :

- les `vcommandes` servent pour manipuler une seule VM

- les `lcommandes` servent à manipuler des ensembles complexes de machines virtuelles en réseau (lab)

Comment ajouter à la volée une interface (ethX) à une machine virtuelle ?

Il faut taper dans la machine réelle :

```
host> vconfig pc1 --eth2=F
```

Comment arrêter les machines ou le lab ?

Pour arrêter les machines :

- utiliser la commande `halt` dans le terminal de chaque machine virtuelle à arrêter.
- soit taper `vhalt pc1` dans le repertoire de votre lab, ...
- soit taper `lhalt -q` dans le repertoire de votre lab.

Est ce qu'une machine virtuelle peut être auto-configurée ?

Oui. Lorsqu'on démarre un lab avec `lstart`, la machine exécutera un script nommé `nom.startup`.

Exemple de commande dans un script pour la machine pc1 :

```
host> cat pc1.startup
ifconfig eth0 192.168.1.1/24
```

Comment copier manuellement des fichiers de la machine réelle vers une machine virtuelle ?

Vous pouvez lire et écrire des fichiers entre la machine virtuelle et la machine hôte par l'intermédiaire de deux repertoires spéciaux disponibles dans chaque machine virtuelle :

- `/hosthome` sur votre machine virtuelle représente votre home directory sur votre machine réelle.
- `/hostlab` sur votre machine virtuelle représente le répertoire de votre lab sur votre machine réelle.

Comment copier manuellement les fichiers de configuration destinés à une machine virtuelle à partir de la machine réelle ?

```
pc1:~# cp -r /hostlab/pc1/etc /
```

Remarque : il est possible que les machines virtuelles n'aient pas eu accès à leurs fichiers de configuration à partir d'un lancement avec lstart -s. Dans ce cas, il vous faudra faire ces opérations manuellement.

Faut-il redémarrer un service sur une machine virtuelle si on a modifié manuellement ses fichiers de configuration ?

Oui

Comment connaître l'état d'un service sur une machine virtuelle ?

```
httpd:~# /etc/init.d/apache2 status
```

Comment arrêter un service sur une machine virtuelle ?

```
httpd:~# /etc/init.d/apache2 stop
```

Comment démarrer un service sur une machine virtuelle ?

```
httpd:~# /etc/init.d/apache2 start
```

Comment redémarrer un service sur une machine virtuelle ?

```
httpd:~# /etc/init.d/apache2 restart
```

Comment observer les communication avec uml_dump (vdump) et wireshark sur le réseau virtuel reliant les machines UML ?

Pour connecter `wireshark` (qui permet de sniffer un réseau local sur un *hub*) avec le domaine de collision **A** d'un *lab*, il faut taper :

```
host> vdump A | wireshark -i - -k
```

Pour ne perdre l'accès à cette console, il est préférable de lancer le processus en arrière plan :

```
host> vdump A | wireshark -i - -k &
```

Pour se déconnecter, il suffit de fermer l'application `wireshark`.

Attention : si vous avez plusieurs domaines de collision (A, B, C, ...), il vous faudra exécuter plusieurs `wireshark` pour *sniffer* les différents trafics.

Comment observer les communication sans uml_dump (vdump) et avec wireshark sur le réseau virtuel reliant les machines UML ?

C'est un peu plus compliqué !

Tout d'abord, vous pouvez "sniffer" les échanges réseaux en utilisant l'outil `tcpdump` à partir des machines virtuelles.

Pour utiliser `wireshark`, il faut passer par un fichier de capture entre les deux machines. Ce fichier sera généré par `tcpdump` sur la machine virtuelle et lu en "temps réel" par `wireshark` sur la machine hôte.

Pour connecter `wireshark` (qui permet de "sniffer" un réseau local sur un *hub*) avec le domaine de collision **A** d'un *lab*, il faudra donc taper :

```
host> tail -f ./outA.cap | wireshark -l -S -H -i - -k
```

Maintenant, il vous faut lancer l'outil `tcpdump` sur la machine virtuelle en indiquant l'interface ethX à "sniffer" :

```
pc1:~# tcpdump -U -n -i ethX -w /hostlab/outA.cap
```

Pour ne perdre l'accès à cette console, il est préférable de lancer ces processus en arrière plan en utilisant & à la fin des commandes saisies au clavier.

Attention : si vous fermez `wireshark` ou `tcpdump`, il vous faudra relancer proprement les deux commandes. Un script `shark.sh` est disponible pour vous faciliter ces manipulations.

Comment sauver l'ensemble des commandes ou leurs résultats ?

Vous pouvez sauvegarder l'ensemble de vos commandes tapées :

```
pc:~# history » /hostlab/save.txt
```

Cette commande va sauvegarder vos commandes en les ajoutant à la fin du fichier `save.txt` situé à la racine de votre lab.

On peut aussi utiliser ce principe pour sauvegarder le résultat des commandes exécutées sur une machine virtuelle :

```
pc:~# ifconfig eth0 » /hostlab/compte-rendu.txt
```