

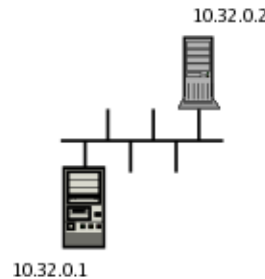


par Gerrit Renker
<gerrit.renker(at)gmx.de>

L'auteur:
L'auteur a obtenu un diplôme en Informatique en 2001.

Traduit en Français par:
Florent Morel <fleuh-(at)-free.fr>

LF Tip : Cloner complètement des PC par le réseau



Résumé:

Il arrive souvent que vous ayez à dupliquer les données d'un ordinateur sur un autre. Une méthode sûre, facile à mettre en oeuvre et efficace est d'utiliser le clonage à travers un réseau tel que décrit dans ce tutoriel.

Scénario

Cloner des animaux ("Dolly la brebis") ou même des embryons humains est âprement discuté et d'un avenir incertain. Cependant, il ne fait aucun doute que cloner des machines est non seulement moins dangereux (à condition de le faire correctement) mais aussi utile pour l'évolution de scripts de configuration et de réglages écrits avec soin. Du fait de la loi de Moore et des rapides progrès de l'industrie des composants informatiques, l'utilisateur d'une distribution est quasiment certain d'avoir à réaliser un scénario de clonage au moins une fois. Par exemple lorsque le PC de la maison est remplacé par un ordinateur portable, ou simplement qu'un processeur plus rapide est disponible et que l'achat d'un nouvel ordinateur complet est trop onéreux. La tâche à réaliser est de *recupérer le système de fichiers en entier* depuis une machine A et de le rendre opérationnel sur une machine B.

Il y a deux manières de le faire ; celle qui n'est pas décrite dans ce tutoriel est d'ouvrir l'ordinateur, transférer physiquement le disque dur vers la machine cible pour effectuer la copie localement. Cela n'est pas souvent possible – ouvrir le boîtier entraîne souvent l'annulation de la garantie – et présente des risques, car une personne inexpérimentée peut endommager le matériel, physiquement ou électriquement. Il m'est même arrivé de perdre le contenu d'un disque à cause d'un logiciel de bas niveau bogué. L'autre approche, qui nécessite que les deux PC aient des cartes réseaux (ce qui est le cas très fréquemment), est bien plus sûre et décrite ici.

L'idée directrice de toutes les méthodes décrites ci-dessous est d'établir une connexion réseau entre la machine "source" (celle qui doit être clonée) et la "cible" (le clone). Cette étape est des plus simples si les deux machines sont branchées sur un hub, sinon vous pouvez connecter les cartes réseaux via un *câble croisé* (dans ce cas, les câbles normaux – droits – ne peuvent être utilisés). Pour le PC cible, un Live-CD (tel que Knoppix ou LNX-BBC) ou une installation minimale est requise afin que la carte réseau soit opérationnelle et

que `ssh` et/ou `netcat` puisse être lancé. Il y a même des disquettes qui permettent cela (bien que dans mon cas `tomsrtb` plantait lors de l'initialisation de la carte réseau). Si vous voulez installer une autre (nouvelle) distribution quoi qu'il arrive, c'est une alternative simple. Les deux machines doivent être configurées avec des adresses IP sur le même réseau pour qu'elles puissent communiquer entre elles, tel que décrit sur l'image ci-dessus.

Méthodes possibles

Considérant une configuration de base, il existe plusieurs façons de réaliser un clonage :

- copie binaire via `dd`
- pipes `tar` / `cpio`
- `rsync`
- `dump` et `restore`

La première est compliquée sinon impossible si vos disques durs n'ont pas exactement les mêmes types et géométries. C'est une façon excellente pour copier des images iso (`dd if=/dev/cdrom of=image.iso`) ou des disquettes – [ici \(shell script de copie de disque – diskcopy\)](#) est un exemple de script `diskcopy` utilisant `dd`. L'autre inconvénient de l'approche basée sur `dd` est que l'espace inoccupé est aussi copié, prenant donc du temps inutilement. Les pipes `tar` et `cpio` prennent beaucoup de temps (jusqu'à plusieurs heures) et présentent quelques problèmes. Par exemple, il existe des restrictions sur les noms de fichiers et l'utilisation de liens symboliques, les fichiers dans `/dev`, etc. Je ne recommanderais généralement pas cette approche pour le clonage.

Si vous avez des systèmes de fichiers différents sur la machine source et cible, **`rsync` (1)** est probablement le meilleur choix. Il nécessite seulement `ssh` et transmet efficacement les fichiers grâce à un protocole des plus performants. Il a même une option `-D` pour les fichiers device, ainsi que plusieurs options pour répondre à la plupart des scénarii pratiques. C'est un outil très utile pour les backups quotidiens, le mirroring et autres. La page de manuel contient de nombreux exemples et est une très bonne source d'informations. Un exemple de clonage via `rsync` est présenté dans [1].

Ici nous utilisons la méthode via `dump` et `restore` dans le but de re-créeer le système de fichiers entier. C'est rapide, efficace et délivre le résultat attendu avec un minimum d'efforts, donc idéal pour le clonage.

J'ai dû réaliser la procédure entière de clonage deux fois, car le PC cible a été rappelé et remplacé. Dans les deux cas, aucun problème n'apparut et le clonage mena à un clone fonctionnel, bootable en à peine une heure de copie des gigaoctets. Cette approche requiert que la machine source et la machine cible possèdent le même système de fichiers. Nous supposons que ce soit `ext2` ou `ext3`, puisque c'est actuellement le plus fréquemment utilisé (voir notes [ci-dessous](#)).

Configurer ssh

Une fois que vous avez un système minimal ou un Live-CD tournant sur la machine cible, la prochaine étape est de configurer `ssh` (si vous n'utilisez pas `netcat` pour le transfert comme détaillé plus loin). Ceci requiert que le PC source ait `sshd` (secure shell daemon) configuré et lancé. Vérifiez `/etc/init.d/` si vous avez un doute. Sur le PC cible entrez (en tant que root) :

```
ssh-keygen -t rsa
```

Pour faire simple, n'entrez pas de passphrase. La clé publique est maintenant dans le fichier `/root/.ssh/id_rsa.pub`. Copiez ce fichier sur votre machine source via

```
scp /root/.ssh/id_rsa SourcePC:/tmp
```

où `SourcePC` est l'adresse IP de votre PC source. Lorsqu'on vous demande si vous êtes sûr, entrez « `yes` » (« `y` » tout seul ne fonctionne que rarement). On vous demande alors le mot de passe root du PC source. Ajoutez à présent le PC cible en tant qu'hôte de confiance sur le réseau en entrant

```
cat /tmp/id_rsa.pub >> /root/.ssh/authorized_keys
```

sur le PC source. Pour vérifier si tout va bien, répétez la commande de copie sur le PC cible. On ne devrait plus vous demander le mot de passe.

Créer un système de fichiers sur le PC cible

La première étape est toujours de partitionner le disque dur sur le système cible et de créer le système de fichiers `ext2/ext3`. Le second est une variante journalisée du premier et est disponible simplement en ajoutant l'option `-j` (journalisation) dans `mke2fs` (nécessite un noyau supportant `ext3`). Vous pouvez même convertir un système `ext2` en `ext3`, voir **tune2fs (8)**. Supposons que sur le PC source nous ayons la configuration suivante :

Filesystem	Size	Used	Use%	Mounted on
/dev/hda3	2.7G	552M	22%	/
/dev/hda5	7.8G	1.6G	22%	/usr
/dev/hda7	6.3G	1.7G	28%	/usr/share
/dev/hda8	3.4G	601M	19%	/home
/dev/hda12	5.3G	1.9G	37%	/opt
/dev/hda1	587M	70M	13%	/var/backup

Je recommande de toujours effectuer un partitionnement. Sinon, un simple problème dans l'utilisation du système ou un problème de disque dur concernant un nombre même limité de secteurs vous fera perdre *l'ensemble* de vos données. Si on se reporte à la loi de Murphy, soyez certains que cela arrivera si vous ne prenez pas la précaution d'utiliser plusieurs partitions au lieu d'une seule monolithique. J'ai rencontré cela avec un noyau étrange et si je n'avais pas partitionné le disque, j'aurais perdu toutes mes données en même temps que le système racine. Le tableau ci-dessus montre que `/usr` devenait trop volumineux, `/usr/share` a donc été ajouté. Il est temps d'acquérir un disque plus grand.

Sur le PC cible, lancez `parted` (recommandé) ou votre outil de partitionnement préféré (`Qt parted` est une variante graphique bien réalisée, dite clone de Partition Magic). Créez les partitions au moins aussi grandes que celles du PC source. N'oubliez pas le swap. Après avoir sauvé la table de partition, désignez le système de fichiers sur toutes les partitions nouvellement créées en utilisant

```
mke2fs -j -L <label> /dev/xxx
```

où `xxx` est le nom de la partition et `<label>` la chaîne de label. J'utilise généralement des labels tels que « `/usr` » (vous verrez cela au moment du boot). Vous pouvez définir plusieurs points via **tune2fs (8)**, comme l'intervalle de vérification du système de fichiers.

Transfert du système

Premièrement vous devez monter toutes les partitions nouvellement créées. Commençons par la racine (« / ») et montons les répertoires restants au cours de la procédure. Il est parfaitement possible de condenser deux partitions du PC source en une seule sur le PC cible, en fait c'est ce que nous allons faire avec `/usr` et `/usr/share` dans l'exemple ci-dessus. Montez donc votre futur système en entrant

```
mount /dev/xxx /mnt
```

Lors du clonage, il est nécessaire de faire un `chdir` dans le répertoire cible

```
cd /mnt
```

Maintenant la partie réseau, sur le PC cible entrez

```
ssh sourcePC 'dump -0 -f - /' | restore -r -f -
```

où `sourcePC` est l'adresse IP du PC source. L'option « `-0` » désigne backup complet, « `-f -` » indique d'utiliser `stdin/stdout` comme descripteurs de fichier et « `-r` » indique à `restore` de re-crée le système copié à travers le réseau sur le PC cible. Pour plus d'options voir **dump (8)** et **restore (8)**. Ci-dessous est présentée la sortie du transfert du système racine.

```
$ ssh 10.42.3.42 'dump -0 -f - /' | restore -r -f -
DUMP: Date of this level 0 dump: Tue Feb 22 15:50:12 2005
DUMP: Dumping /dev/hda3 (/) to standard output
DUMP: Label: debian
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 547312 blocks.
DUMP: Volume 1 started with block 1 at: Tue Feb 22 15:50:14 2005
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Volume 1 completed at: Tue Feb 22 15:51:43 2005
DUMP: Volume 1 546590 blocks (533.78MB)
DUMP: Volume 1 took 0:01:29
DUMP: Volume 1 transfer rate: 6141 kB/s
DUMP: 546590 blocks (533.78MB)
DUMP: finished in 89 seconds, throughput 6141 kBytes/sec
DUMP: Date of this level 0 dump: Tue Feb 22 15:50:12 2005
DUMP: Date this dump completed: Tue Feb 22 15:51:43 2005
DUMP: Average transfer rate: 6141 kB/s
DUMP: DUMP IS DONE
```

`Restore` crée toujours un fichier `restoresymtable` qui peut être supprimé une fois que vous êtes sûr qu'aucune erreur n'est intervenue lors de la restauration du système. Une fois le système racine copié, procédons maintenant à chaque sous-système monté, en commençant par `/usr` (supposant que votre répertoire courant est la racine du futur système de fichiers).

```
mount /dev/xxx ./usr
```

```
cd ./usr
```

```
ssh targetPC 'dump -0 -f - /usr' | restore -r -f -
```

L'opération `mount-cd-dump/restore` est à répéter pour tous les répertoires que vous pouvez avoir. Pour ce qui est de `/usr/share` (qui avait sa propre partition sur le PC source), vous pouvez, après l'étape précédente,

faire un chdir vers `./usr/share` (notez le « . ») et ensuite répéter

```
ssh targetPC 'dump -0 -f - /usr/share' | restore -r -f -
```

Restore se plaint uniquement si des fichiers existent déjà dans le système à restaurer, donc si vous mettez deux partitions différentes (sur le PC source) en une seule (sur le PC cible) il n'y a pas de problème. Cloner un PC entier prend à peine une heure avec ssh et des cartes réseaux 100MB (un gain de performance est à noter avec un câble croisé).

Note: Pour réaliser le dump d'un système, il n'est pas nécessaire qu'il soit monté. Vous pouvez aussi passer *le nom de la partition*, tel que `/dev/hda6`, en lieu et place du nom de répertoire de la partition montée.

Alternative : netcat

Une alternative à ssh est d'utiliser `netcat` (1), dont l'abréviation est `nc`. Netcat est un couteau suisse facile à mettre en oeuvre, fonctionnant en mode client-serveur sur TCP/IP qui permet de créer un tuyau à travers le réseau. Les exemples précédents sont modifiés comme suit. Disons que la partition montée sur `/var/backup` doit être transférée via `dump/restore` depuis le PC source vers le PC cible.

Sur la machine cible, créez une instance de `netcat` qui écoute via `-l` et qui redirige sa sortie vers `restore`.

```
nc -l -p 2000 -q 1 | restore -r -f -
```

Sur la machine source, créez une autre instance de `netcat` qui récupère l'entrée depuis le pipe créé précédemment où `target-IP` représente l'adresse IP du PC cible.

```
dump -0 -f - /var/backup | nc <target-ip> 2000
```

L'option `-q` est censée arrêter `netcat` après avoir reçu l'identifiant de fin de fichier, mais lors de mon essai, il m'a fallu le stopper manuellement. Je recommande d'utiliser `ssh` malgré tout.

Faire le ménage

Félicitations, si vous êtes arrivés jusqu'ici, vous avez cloné votre système. Maintenant il est temps de rendre cette machine utilisable. La première chose à faire est de mettre à jour `/etc/fstab` avec les nouveaux réglages, sinon vous ne pourrez pas utiliser les partitions clonées. Si l'adresse IP change, celle-ci doit aussi être mise à jour (`/etc/hosts`, `/etc/network/interfaces` sous debian). La prochaine étape importante concerne la configuration du démarrage qu'il sera pratiquement toujours impératif de mettre à jour. Avec `lilo`, éditez `/etc/lilo.conf` (en particulier l'option `root=...`) puis exécutez `lilo -v`. Sous `grub`, éditez `/boot/grub/menu.lst` (ou `/boot/grub/grub.conf`, tout dépend sur lequel pointe le lien symbolique) puis entrez `grub`,

```
grub> root (hd0,xxx)
... filesystem is ...
grub> setup (hd0)
... lots of output here
```

```
grub> quit
```

ou lancez `grub-install /dev/xxx` où `xxx` est votre disque dur. Ici, vérifiez les réglages pour `root` (`hdn, xx`) et `root=/dev/xxx`.

Dans le cas probable où vous avez un matériel plus récent dans votre PC cloné, vous pouvez mettre à jour les *réglages pour votre noyau personnalisé*. Si vous utilisez des systèmes qui proposent de nombreux modules pré-configurés (tels que RedHat, SuSE, Mandrake, Fedora...) ne vous inquiétez pas, il est fort probable qu'il existe un module correspondant. Sinon, `lspci -vv` et compilation du noyau habituels tels que décrits ailleurs vous permettront d'arriver à vos fins. Si votre carte graphique est différente, mettez à jour `/etc/X11/XF86Config-4` (ou `xorg.conf` sous RH/Fedora) pour le prendre en compte, sinon vous n'obtiendrez aucun signal. Si possible, utilisez des outils graphiques pour configurer X en démarrant en niveau 3 si vous disposez de tels outils. Sous Debian, quelques investigations sont nécessaires. Je fus chanceux de découvrir que le driver avait changé de `r128` vers `radeon` et que cela fonctionnait.

Autres systèmes

Ce tutoriel explique la procédure de clonage pour les systèmes de fichiers `ext2/ext3`. Des commandes similaires conviennent pour de nombreux autres systèmes Unix-like. Par exemple, plusieurs Unix tels que FreeBSD, HP-UX, IRIX disposent aussi des commandes `dump/restore`; sous Solaris ce sont `ufsdump/ufsrestore`. Il existe des systèmes de fichiers qui n'offrent pas la fonctionnalité « dump », c'est le cas de ReiserFS. Dans ce cas, il est conseillé d'utiliser `rsync`. Voir [1] pour un rapport concernant l'utilisation de `rsync` lors d'une procédure de clonage d'un système Linux.

Références

[1] "*Replicating a Linux System – Yet Another Method.*" Ben Okopnik, Linux Gazette Issue 83, October 2002.

<p>Site Web maintenu par l'équipe d'édition LinuxFocus © Gerrit Renker "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Gerrit Renker <gerrit.renker(at)gmx.de> en --> fr: Florent Morel <fleuh-(at)-free.fr></p>
--	---

2005-09-15, generated by lfparsr_pdf version 2.51