

Réinstallation de postes clients avec PXE et Partimage

Guilhem Borghesi

Université Louis Pasteur

UFR de Mathématique et d'Informatique

7 Rue René Descartes, 67084 Strasbourg

borghesi@dpt-info.u-strasbg.fr

Christophe Boccheciampe

Université Louis Pasteur

UFR de Mathématique et d'Informatique

7 Rue René Descartes, 67084 Strasbourg

cb@dpt-info.u-strasbg.fr

Résumé

Le métier d'administrateur système demande de gérer un maximum de machines avec un minimum d'efforts. Pour ce faire, nous avons cherché une solution de réinstallation automatique d'ordinateurs personnels dual-boot (Windows, Linux) afin d'alléger le travail d'administration et de proposer une autonomie aux usagers.

La solution retenue s'appuie sur la technologie PXE (Preboot eXecution Environment) et le logiciel libre Partimage. PXE permet à une machine de télécharger et d'exécuter une mini-image dès son démarrage, cette image contenant le système d'exploitation minimal utilisé par la suite. Une fois ce chargement effectué, l'application Partimage est alors utilisée pour remplacer les différentes partitions du disque dur par des images préparées à l'avance.

Le but de cette présentation est de montrer les avantages de cette solution mais aussi d'aborder les problèmes rencontrés ainsi que les améliorations possibles.

Mots clefs

PXE, Partimage, TFTP, NFS, DHCP, administration système, poste client

1 Introduction

1.1 Le contexte

L'UFR de Mathématique et d'Informatique de l'Université Louis Pasteur de Strasbourg héberge environ 160 postes informatiques. Ils sont destinés à un usage pédagogique pour les enseignements en informatique dans les filières de l'UFR. Ce parc comprend en majorité des postes de type « clients légers » ainsi que 60 PC. Ces PC *dual-boot* (Windows XP et Linux Gentoo) sont utilisés en travaux pratiques dans des enseignements tels que « Réseaux », « Ingénierie Logicielle », « Calcul Scientifique », « Développement Graphique », « Administration Système », ... cette solution devait donc être capable de restaurer les systèmes Windows et Linux en même temps.

Ces machines offrent un contexte logiciel complexe à mettre en place et à maintenir. Elles sont amenées à être reconfigurées, réinstallées ou même reformatées durant ces séances de travaux pratiques. Sur certaines d'entre elles, les

usagers sont également amenés à utiliser le compte « super-utilisateur » pour réaliser leurs travaux.

Nous avons donc cherché une solution de réinstallation semi-automatique de ces postes afin d'alléger le travail d'administration et de proposer une possibilité de réinstallation par les usagers. De plus, ces réinstallations peuvent avoir lieu plusieurs fois par semaine, voire par jour.

Il existe de nombreuses solutions logicielles, payantes ou libres, pour réaliser ce type de réinstallation automatique, nous avons donc commencé par déterminer nos contraintes.

1.2 Les contraintes

Tout d'abord, en accord avec nos convictions en terme de choix techniques ainsi que pour des raisons pratiques, une solution issue du monde du logiciel libre nous a paru toute naturelle. C'était sans conteste, celle qui nous offrirait une adaptabilité optimale et saurait répondre aux différentes exigences de nos salles de TP. Par ailleurs, après avoir précédemment utilisé une solution à base de réinstallation par CD-ROM, nous voulions nous débarrasser de cette contrainte matérielle, fragile et coûteuse, en privilégiant une réinstallation par le réseau. Il fallait également que le temps de réinitialisation d'une salle complète n'excède pas un quart d'heure, afin de ne pas imposer un délai trop important aux usagers.

D'un point de vue strictement matériel, nous disposons de salles de PC homogènes. Dans ces salles, chaque poste possède une carte *FastEthernet* compatible **PXE** et l'infrastructure réseau est constituée d'un commutateur (*switch*) équipé de 24 ports *FastEthernet* ainsi que de deux *uplinks GigaEthernet*.

2 Présentation générale

2.1 Principe de fonctionnement

La procédure de réinstallation est semi-automatique : son déclenchement est manuel, mais la restauration du ou des systèmes d'exploitation est ensuite automatique, jusqu'au redémarrage complet du PC.

En théorie, les images à restaurer peuvent se trouver sur les supports suivants :

- un CD/DVD-ROM : « *bootable* » + images (taille réduite dans le cas d'un CD-ROM)
- un disque dur local dédié : idem
- une clé USB (si le BIOS permet le démarrage sur un tel support) : « *bootable* » + images, si la capacité de la clé le permet
- sur une machine tierce accessible via le réseau (répertoire partagé via NFS, Samba, ...)

Bien qu'éprouvées et performantes, les trois premières solutions sont contraignantes : pour une restauration simultanée, il faut en effet posséder autant de supports que de postes cibles. Or, dupliquer un système sur plusieurs CD/DVD ou clés USB devient rapidement fastidieux, et installer un disque dur dédié sur chaque machine cible peut s'avérer relativement onéreux. De plus, chaque mise à jour majeure des images sources implique une nouvelle duplication sur ces supports.

La solution retenue est la dernière : afin d'accéder aux images sources stockées sur un serveur, la machine cible récupère dans un premier temps un noyau Linux et une arborescence (système de fichiers) via le réseau, grâce aux protocoles PXE, DHCP et TFTP. Ces éléments sont alors placés en mémoire puis le noyau est exécuté. Les options passées à ce dernier (via PXE) font en sorte que l'arborescence récupérée soit utilisée comme racine (point de montage « / »). Un simple script « *bash* » se charge alors d'exécuter Partimage, application utilisée pour restaurer les partitions locales du disque dur à partir des images sources.

2.2 Technologies utilisées

Notre solution repose sur les technologies suivantes :

- PXE (Pre eXecution Environment) : environnement de démarrage de système dont les spécifications ont été publiées par Intel et SystemSoft en 1999.¹
- Partimage : Utilitaire libre, sous Linux, permettant de sauvegarder et de restaurer les partitions d'un disque dur dans un fichier image. Il est développé depuis juin 2000 par François Dupoux et Franck Ladurelle, sous licence GPL.² Nous utilisons actuellement la version 0.6.4.
- TFTP (Trivial File Transfert Protocol) : protocole simplifié de transfert de fichiers. Ce protocole est défini dans la RFC 1350 de juillet 1992.³
- NFS (Network File System) : protocole développé par Sun Microsystems qui permet à un ordinateur d'accéder à des fichiers via un réseau. La version 3 de ce protocole que nous avons employé est défini par la RFC 1813 de juin 1995.⁴
- DHCP (Dynamic Host Configuration Protocol) : protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres TCP/IP d'un poste de travail. Ce protocole a été présenté pour la

première fois en octobre 1993 et est défini par le RFC 1531.⁵

2.3 Matériel requis

Tout d'abord, il est inutile de se lancer dans cette solution si le parc de machines clientes à restaurer n'est pas homogène, cela rendrait le travail trop fastidieux car il serait alors indispensable de posséder autant d'images que de machines différentes.

En ce qui concerne les composants, les postes clients se doivent de posséder une carte réseau ou une carte mère intégrant les fonctionnalités du protocole PXE.

Le serveur hébergeant les images source (sur lequel le démon NFS va être exécuté) doit quant à lui posséder une carte réseau *GigabitEthernet* (une simple carte *FastEthernet* limite les performances de manière trop importante pour réinstaller simultanément plusieurs postes). L'infrastructure réseau doit bien entendu répondre à ces exigences : des commutateurs *GigabitEthernet* sont donc fortement conseillés.

3 Configuration logicielle

Nous allons maintenant parler de la configuration des différents services à mettre en place pour que les images soient restaurées.

3.1 DHCP

Les cartes réseaux PXE peuvent démarrer sur n'importe quel serveur DHCP. Cependant, pour que PXE puisse fonctionner correctement il faut rajouter quelques lignes au début du fichier de configuration du serveur (souvent */etc/dhcp/dhcpd.conf*) :

```
allow booting ;
allow bootp;
```

Ensuite, suivent très traditionnellement les paramètres de sous-réseaux, l'adresse de *broadcast* et l'adresse de la passerelle par défaut :

```
subnet 130.79.6.0 netmask 255.255.254.0 {
option broadcastaddress
130.79.7.255;
option routers 130.79.7.254;
```

A ces options sont ajoutées les informations propres au démarrage PXE, dans une sous-section nommée « *group* ». Ces informations sont l'adresse du serveur TFTP (*next-server*), le répertoire racine de ce dernier (*root-path*) et le nom de l'exécutable PXE à lancer (*filename*):

```
group {
next-server                130.79.6.254;
option    root-path        "/tftpboot/";
option    netbios-name-servers 130.79.7.2;
filename                "pxelinux.0";
}
```

¹<http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf>

²<http://www.partimage.org/Index.fr.html>

³<http://www.ietf.org/rfc/rfc1350.txt>

⁴<http://tools.ietf.org/html/rfc1813>

⁵<http://tools.ietf.org/html/rfc1531>

Enfin, on ajoute à ce groupe les informations propres à chacune des machines, avec une distribution d'adresse statique par adresse MAC :

```
host res-a {
    hardware ethernet AA:BB:CC:DD:EE:FF;
    fixed-address 192.168.1.1;
}
```

Une fois le serveur DHCP relancé, on peut se pencher sur le serveur TFTP.

3.2 Tftpd avec inetd

Il est conseillé d'utiliser TFTP-HPA ou ATFTP pour faire cette manipulation car il faut un serveur TFTP qui supporte l'option « tsize » (RFC 1784)⁶. Il suffit d'indiquer dans le fichier de configuration de TFTP l'emplacement de la racine du serveur (*/tftpboot* dans notre cas). Ensuite, dans le fichier de configuration d'*inetd* (*/etc/inetd.conf*) on ajoute le démon TFTP :

```
tftp dgram udp wait root /usr/sbin/in.tftpd /tftpboot
```

Un redémarrage du service *inetd* donne alors accès au répertoire */tftpboot*, via TFTP sur le port 69 du serveur.

3.3 PXE

Dans le répertoire */tftpboot* doit se trouver l'exécutable *pxelinux.0*, lequel va chercher un fichier de configuration qui devra se trouver obligatoirement dans le sous-répertoire */tftpboot/pxelinux.cfg/*. Le nom de ces fichiers est normalisé : chacun va correspondre à la machine pour laquelle on cherche la configuration de démarrage.

Ce nom peut prendre deux formes : soit l'adresse MAC (*01-88-99-aa-bb-cc-dd*), soit l'adresse IP donnée par le serveur DHCP, codée en hexadécimal : *824F0601* pour la machine d'adresse 130.79.6.1, par exemple. Si toutes les machines possèdent la même configuration de démarrage, on peut se contenter de ne mettre qu'un fichier *default*.

Vous pouvez trouver le contenu de ce type de fichier de configuration en annexe.

3.4 NFS

La restauration des images sur le poste client peut se faire via un serveur *partimaged* qui serait sur le serveur d'images, mais il s'avère bien plus lent qu'un simple export NFS. Le développeur de Partimage lui-même conseille plutôt l'utilisation de NFS. Pour procéder à la configuration du serveur d'images, il faut modifier le fichier */etc/exports* en y ajoutant une simple ligne de ce type :

```
/tftpboot 192.168.1.1/26(rw)
```

Un redémarrage du démon NFS est nécessaire pour prendre les modifications en compte et partager ses images sur le réseau.

Sur les postes clients (dans le mini-système de restauration), l'ajout d'une ligne dans le fichier */etc/fstab* suffira à « monter » le répertoire partagé :

```
serveur:/tftpboot /mnt/images nfs defaults 0 0
```

Dans cet exemple, le répertoire */tftpboot* du serveur d'images sera monté en local dans le répertoire */mnt/images*, via le système de fichiers NFS.

Lorsque tous ces services sont opérationnels sur le serveur d'images, la prochaine étape consiste à préparer le mini-noyau qui fera démarrer le mini-système sur le poste client, lors de la phase de restauration.

3.5 Préparation du mini-noyau

Pour préparer le noyau minimal nécessaire, nous avons téléchargé les sources de ce dernier, configuré en fonction de nos besoins puis compilé de façon classique. Les besoins en question sont les pilotes et options nécessaires au bon fonctionnement de la restauration sur le poste client :

- pilote spécifique à la carte réseau,
- pilote pour la gestion des disques durs (IDE, SATA ou SCSI),
- gestion des systèmes de fichiers réseaux (client NFS) et locaux (en fonction des images à restaurer : EXT3, ReiserFS, FAT32, NTFS, ...),
- support des disques mémoire (RAM disk) et de la fonctionnalité « Initial RAM filesystem and RAM disk (initramfs/initrd) », nécessaire pour la gestion de l'arborescence racine qui sera chargée en mémoire,
- toute option pouvant d'une façon ou d'une autre réduire le temps nécessaire à la restauration des images (microprocesseur spécifique, par exemple).

Afin d'éviter la gestion des modules par le noyau, tous ces pilotes sont directement intégrés à ce dernier. Ainsi, un seul fichier sera nécessaire au système pour que le matériel présent sur la machine cible soit automatiquement détecté et initialisé.

Si plusieurs ensembles de machines sont à restaurer (ajout d'une nouvelle salle composée de machines plus récentes, par exemple), il est, bien entendu, possible d'intégrer à ce noyau les pilotes nécessaires à chaque ensemble. Il n'est donc pas nécessaire de disposer d'un noyau spécifique à chacun, à moins que l'espace occupé par ce noyau en mémoire ne soit une contrainte (une augmentation du nombre de pilotes augmente la taille du noyau).

Le noyau ainsi généré est à placer sur le serveur d'images, le nom à lui donner étant celui qui figure dans le fichier de configuration de PXE.

Les commandes utilisées pour configurer et compiler le noyau figurent en annexe de ce document.

3.6 Préparation de la mini-arborescence

L'arborescence racine (« initrd »), mise en mémoire au démarrage et montée par le noyau, doit contenir les

⁶<http://syslinux.zytor.com/pxe.php>

exécutables et les bibliothèques nécessaires au démarrage du système.

Pour préparer cette arborescence, il faut tout d'abord se placer sur une machine tierce munie d'un système d'exploitation de type Linux (de préférence) et copier les fichiers qui composeront cette arborescence dans un répertoire dédié. Il s'agit notamment des répertoires et fichiers de base (*/etc*, */dev*, */bin*, */usr/sbin*, ...) sans lesquels le mini-système ne pourrait démarrer correctement. Ceci inclut bien entendu les fichiers de configuration et autres scripts qui seront utilisés pendant et après le démarrage du mini-système sur la machine cible.

La liste des répertoires et fichiers copiés se trouve en annexe de ce document.

Une fois ces fichiers copiés, l'étape suivante consiste à créer un fichier vide d'une taille égale à 32 Méga-octets (taille normalement suffisante pour contenir les fichiers mentionnés précédemment) :

```
dd if=/dev/zero of=initrd bs=1M count=32
```

avec :

- *if* : fichier source (*/dev/zero* génère des octets nuls),
- *of* : fichier cible (notre fichier image « *initrd* »),
- *bs* = taille d'un bloc de données, ici en méga-octets,
- *count* = nombre de blocs dans le fichier cible.

Le fichier ainsi créé va contenir notre arborescence, il doit donc posséder un système de fichiers :

```
mke2fs -i 1024 -m0 -F initrd
```

avec :

- *-i* : nombre d'octets par inode,
- *-m0* : pourcentage de blocs réservés au super-utilisateur,
- *-F* : force l'exécution de *mke2fs* sur le fichier en paramètre,
- *initrd* : nom du fichier dans lequel créer le système de fichiers.

Après avoir créé le système de fichiers dans notre *initrd*, il ne reste plus qu'à le « monter » dans un répertoire, de la même manière qu'un disque classique, puis de transférer le répertoire préparé précédemment vers le point de montage :

```
mount -o loop initrd /mnt/notre_initrd/
```

```
sync -a notre_repertoire/* /mnt/notre_initrd
```

L'option « *-o loop* » est utilisée pour faire correspondre un périphérique virtuel (*/dev/loop**) à notre fichier. Ce dernier sera alors vu comme un périphérique « montable ».

La commande *rsync* permet quant à elle de synchroniser deux dossiers (s'ils existent déjà, les fichiers cibles sont mis à jour). L'option « *-a* » permet de conserver un certain nombre de paramètres entre la source et la destination (copie récursive avec conservation des liens symboliques, des permissions, des propriétaires des fichiers, des groupes, des descripteurs de périphériques).

La dernière étape consiste à « démonter » le fichier *initrd* puis à le compresser à l'aide de « *gzip* » pour lui donner sa forme finale :

```
umount /mnt/notre_initrd
```

```
gzip -f -S .img initrd
```

avec :

- *-f* : force la compression,
- *-S .img* : force le suffixe « *.img* » à la place du suffixe par défaut « *.gz* ».

Notre mini-arborescence « *initrd.img* » est maintenant prête : il ne reste plus qu'à la transférer sur le serveur TFTP, dans le répertoire prévu à cet effet.

3.7 Master Boot Record (MBR)

Le MBR d'un disque dur (secteur de « boot » du disque, situé au début de ce dernier) contient des données essentielles sur la structure des partitions qui le composent (table des partitions primaires) ainsi que les informations nécessaires au démarrage du système d'exploitation présent sur ce disque.

Afin de restaurer correctement les différentes partitions de la machine cible, le disque dur de cette dernière doit au moins posséder une table de partitions équivalente (d'où l'importance de l'homogénéité des machines à restaurer). Pour ce faire, nous recopions purement et simplement le MBR du disque qui a servi à créer les images sur le disque de la machine cible.

Pour effectuer une sauvegarde du MBR de la machine source, nous exécutons la commande suivante (depuis cette machine, bien entendu) :

```
dd if=/dev/hda of=sauvegarde_mbr bs=512 count=1
```

avec :

- *if* : source : descripteur de périphérique, i.e le disque dur dont on veut sauvegarder le MBR (*/dev/sdb*, */dev/sda*, ...),
- *of* : fichier dans lequel sauvegarder les données,
- *bs* : taille du bloc de données à sauvegarder, ici en octets,
- *count* : nombre de blocs à prendre en compte.

Le programme *dd* va donc lire les 512 premiers octets du disque dur */dev/hda* et les écrire dans le fichier « *sauvegarde_mbr* ».

Lors de la restauration, il suffira d'intervertir les paramètres « *if* » et « *of* » de la commande *dd* afin de remplacer le MBR du disque cible.

Remarque : les machines cibles ne comportant pas de partitions étendues (uniquement 3 partitions primaires), il n'est pas nécessaire de sauvegarder et restaurer les informations de ces dernières.

4 Création des images sources

Nous entrons maintenant dans le vif du sujet. Tout ce qui a été mis en place jusqu'à maintenant va permettre de reproduire les systèmes que nous allons mettre en place. Plusieurs facteurs sont maintenant à prendre en considération : le partitionnement ainsi que le choix de l'offre logicielle.

En effet, la procédure de préparation des images à restaurer étant suffisamment longue, il faudra prendre tout le temps nécessaire pour installer l'offre logicielle complète. Naturellement, cette phase du projet demande de prendre contact avec les utilisateurs pour connaître leurs besoins de façon précise. Par ailleurs, lors de la création de la machine « Source » nous avons pris un soin tout particulier à n'installer que le strict minimum pour éviter de rallonger inutilement la procédure de restauration. Il faut donc trouver le juste milieu entre ces deux contraintes antagonistes.

L'arrêt des services installés par défaut, sur les systèmes Unix et Windows permet, enfin, d'améliorer la sécurité d'un parc de machines. Voyons cela plus en détail.

4.1 Partitionnement

Avant de commencer à faire l'installation des systèmes d'exploitation, il faut effectuer un partitionnement correct de l'espace disque à disposition. Les enseignements qui ont lieu dans les salles de l'UFR ont besoin d'être faits à la fois sur les plate-formes Linux et Windows. Il nous faut donc opter pour un système *dual-boot* avec un partitionnement adapté.

Partimage ne restaurant pas les parties vides du disque, cela permet de prendre des partitions de taille bien supérieure au besoin réel sans impacter pour autant le temps de réinstallation. Les disques durs ont une capacité de 60 Go. Nous avons opté pour un partitionnement de ce type :

Linux (ext3) : 29 Go

Linux Swap : 2 Go

Windows (FAT32) : 29 Go

Voyons maintenant la construction de l'image Windows.

4.2 Image Windows

L'installation de la machine sous Windows se fait classiquement. Nous avons testé cette méthode avec Windows 2000 et Windows XP Professionnel.

L'installation des logiciels par la suite se fait également très normalement. Pour les besoins des enseignements, nous avons mis en place notamment Firefox, OpenOffice, IZArc et Adobe Reader. Il est important de bien penser à effacer les fichiers exécutables servant à l'installation de ces logiciels qui prendront de la place inutilement dans l'image. Il ne faut pas non plus oublier d'effacer le cache de navigation de Firefox et tous les fichiers temporaires de Windows (Corbeille, cache d'Internet Explorer, cookies, fichiers TMP, ...).

Les images étant réinstallées sur plusieurs machines d'un même sous-réseau, il faut procéder à la banalisation de la machine source, afin que les machines cibles ne soient pas toutes identiques lors de la réinstallation massive.

Typiquement, après le clonage, il faut éviter que toutes les machines restaurées aient le même nom ou le même identifiant de sécurité (SID, sous Windows NT). Pour ce faire, on utilise le logiciel Sysprep fourni avec Windows depuis la version Windows NT 4.0. Pour fonctionner, il doit être installé dans le répertoire [C:\Sysprep](#) de la machine. Dans ce répertoire se trouve l'exécutable

sysprep.exe ainsi que le fichier de configuration sysprep.inf. Celui-ci contient toutes les données spécifiques à la machine qui va être restaurée : Nom de la machine (ComputerName), mot de passe administrateur (AdminPassword), nom de domaine ou de groupe de travail (JoinWorkGroup), clé d'activation Windows (ProductID), le fuseau horaire (TimeZone), la configuration des cartes réseaux (InstallDefaultComponents), ...

Lors du premier redémarrage de la machine après le clonage, Windows passera par une phase de mini-installation où toutes les informations contenues dans le fichier sysprep.inf seront passées au système. On peut d'ailleurs automatiser cette mini-installation en rajoutant des valeurs dans sysprep.inf pour faire en sorte que le programme ne demande aucune intervention humaine et redémarre de lui-même le système finalisé : avoir une installation complètement automatisée (UnattendMode = FullUnattended), passer le contrat de licence (OEMSkipEula), passer la page de bienvenue (OEMSkipWelcome), passer la demande de fuseau horaire (OEMSkipRegional). Vous trouverez notre fichier sysprep.inf générique en annexe.

Évidemment, il ne faut pas que ce fichier sysprep.inf soit identique pour toutes les machines, cela n'aurait aucun intérêt. La méthode employée pour personnaliser chaque fichier sysprep.inf est la suivante : le script de réinstallation choisit le bon fichier de configuration Sysprep, en fonction de l'adresse MAC de la machine. Il le copie alors dans le répertoire [C:\Sysprep](#), juste avant que le premier redémarrage ait lieu et le tour est joué.

Lorsque l'image est considérée comme définitivement prête, vous pouvez lancer Sysprep. On le lancera de préférence en utilisant « Démarrer » puis « Exécuter », afin d'éviter l'ouverture d'une fenêtre « Poste de travail » sur tous les postes après clonage. L'interface nous propose plusieurs choix de restauration. Nous avons choisi « Utiliser la mini-installation » puis « Resceller ». La machine s'éteint après le processus d'initialisation et votre image est prête.

Passons maintenant à la création de l'image Linux.

4.3 Image linux

L'installation de la partition Linux s'effectue également tout à fait normalement, hormis peut-être le choix du « Boot Loader » (gestionnaire de démarrage). Pour un souci de maîtrise de l'outil, nous avons choisi LILO plutôt que GRUB, mais la manipulation reste, dans le fond, identique. Une fois les installations de logiciels et le nettoyage de toutes les sources et du répertoire temporaire */tmp effectués*, on peut se lancer dans la fabrication des images.

On prendra garde toutefois avant d'éteindre la machine de bien arrêter et faire en sorte que ne redémarrent pas des services inutiles sur ces postes. On portera d'ailleurs une attention particulière aux services réseaux qui sont des points d'attaques externes d'un système (smtp, imap, http).

4.4 Sauvegarde des images

Pour pouvoir créer les images avec le client Partimage, les partitions ne doivent pas être montées. Cela signifie qu'il

faut pouvoir accéder à ces partitions à partir d'un autre système d'exploitation. Les développeurs de Partimage conseillent d'utiliser le CDROM bootable « SystemRescueCD » qui permet de monter un disque distant avec NFS et de lancer le client Partimage à partir d'un système chargé en RAM.

La méthode employée pour faire ces images est donc simple : Démarrage sur SystemRescueCD, montage de la partition NFS du serveur d'images, lancement du client Partimage pour la sauvegarde des images directement sur le serveur.

5 Déploiement sur poste client

Maintenant nous allons voir la réalisation concrète de ce travail sur un poste client.

5.1 Boot PXE

La première étape est le démarrage en PXE des postes clients. Cela s'effectue sur nos postes (mais cela est très variable d'une machine à l'autre) grâce à la touche F12. La machine cherche alors en *broadcast* un serveur DHCP qui lui réponde. On a vu que la configuration du serveur DHCP lui donnait l'emplacement de l'exécutable *pxelinux.0*. Celui-ci cherche le répertoire *pxelinux.cfg* qui contient les fichiers de configuration pour le lancement des images. L'exécutable se lance avec le fichier de configuration qui correspond à l'adresse IP de la machine (fichier en annexe). Ce fichier de configuration indique à PXE de charger le noyau se trouvant à la racine du serveur TFTP.

5.2 TFTP

A partir de cet instant, le mini-noyau, chargé via TFTP puis placé en mémoire vive (RAM), monte le fichier *initrd* qui lui est indiqué dans le fichier de configuration. La machine démarre alors avec le système complet chargé en RAM (cf section 3.6).

Le strict nécessaire est exécuté afin de lancer un client DHCP, un client NFS et enfin le client Partimage qui va lancer la restauration des images se trouvant sur le serveur d'images.

5.3 Script de réinstallation

Le script de réinstallation est automatiquement exécuté grâce aux mécanismes offerts par Linux (création d'un lien symbolique dans le répertoire */etc/rc2.d/*, pointant vers notre script).

Les opérations suivantes sont effectuées dans cet ordre :

- montage du répertoire partagé par le serveur, via NFS,
- remplacement du MBR du disque de la machine à restaurer et relecture de la table de partitions (afin que le système prenne en compte le nouveau MBR),
- création du système de fichiers sur les partitions à restaurer (Linux, Swap Linux et Windows),
- exécution de Partimage pour restaurer la ou les partitions souhaitées,
- personnalisation de chaque poste cible en fonction des fichiers de configuration préparés à l'avance (*sysprep.inf* pour Windows, fichier *hostname* pour

Linux). Les fichiers à copier sont déterminés grâce au fichier de correspondance « MAC_HOST » placé sur le serveur,

- exécution de LILO pour mettre à jour le menu de démarrage,
- redémarrage de la machine.

Une version simplifiée de ce script se trouve en annexe.

6 Bilan

Pour faire un bilan des 4 années d'exploitation de cette solution, nous pouvons dire qu'elle nous a apporté beaucoup de facilité pour la gestion des postes clients, bien qu'il nous ait fallu surmonter quelques problèmes avant d'arriver à une situation stable. Voyons les plus en détail.

Il nous a fallu, à chaque nouvelle image créée, faire en sorte que les partitions soient les plus petites possibles et par conséquent, faire la chasse à l'espace gaspillé par le système. Typiquement, dans une distribution Debian, les archives des mises à jour sont téléchargées et stockées dans le répertoire */var/cache/apt/archives*. Ce répertoire est souvent très volumineux et les fichiers peuvent être effacés sans risque.

Un autre angle d'attaque est celui des performances réseaux. La recherche d'optimisations dans ce domaine a très nettement montré des gains notables sur le temps de réinstallation des postes clients. On peut notamment s'orienter vers l'utilisation systématique de cartes réseaux *GigabitEthernet* (commutateurs inclus), tout en mettant en place des agrégations de liens entre les différents équipements actifs du réseau, afin d'éviter les goulots d'étranglement.

Dans cette quête de performances, la qualité du serveur est évidemment primordiale, notamment de par la bande passante du réseau ainsi que le débit de données offert par ses disques. Des disques SCSI ou SATA (10000 tr/mn) seraient sans doute souhaitables pour bénéficier de performances correctes.

En conclusion, malgré le temps passé à chercher dans une documentation parfois absente, la complexité technique de certaines parties de cette solution et les installations nécessaires de nouveaux serveurs, nous pouvons dire que nous sommes pleinement satisfaits.

Actuellement, 5 salles de notre UFR sont installées de cette façon (environ 80 PC), et nous sommes de moins en moins sollicités pour l'installation d'un logiciel sur toute une salle machine.

Annexe

Préparation et compilation d'un (mini-)noyau

Les sources du noyau Linux sont téléchargeables à cette adresse : <ftp://ftp.kernel.org/pub/linux/kernel/v2.6>

- décompression des sources (répertoire `/usr/src`) :

```
tar jxf linux-2.6.xx.tar.bz2
```

- dans le répertoire créé, lancement de l'interface de configuration :

```
cd linux-2.6.xx && make menuconfig
```

L'interface de configuration permet de sélectionner tous les éléments dont vous avez besoin pour faire fonctionner votre matériel. Une documentation simple est disponible directement via cette interface, et des fichiers détaillés se trouvent dans le sous répertoire « Documentation » des sources du noyau. La configuration d'un noyau étant relativement compliquée pour un novice, il est recommandé, en cas de doute, de chercher les informations nécessaires sur Internet via n'importe quel moteur de recherche.

Une fois la configuration terminée, il suffit de sortir de l'interface (la sauvegarde des paramètres est automatiquement proposée, il faut bien entendu l'accepter).

La commande suivante lance la compilation du noyau :

```
make bzImage
```

Le paramètre « `bzImage` » permet de créer une version compressée du noyau (gain d'espace). Dès que le système rend la main et si aucune erreur n'est survenue, le noyau pourra être copié depuis l'emplacement suivant :

```
/usr/src/linux-2.6.xx/arch/i386/boot/bzImage
```

Le répertoire « `i386` » peut être différent en fonction de l'architecture pour laquelle vous avez compilé votre noyau (par exemple : « `sparc` », ou « `ia64` »).

Un autre fichier, « `/usr/src/linux-2.6.xx/System.map` » devra être copié en même temps que le noyau : il s'agit de la table des « symboles » (variables, fonctions, ...) utilisés par votre noyau.

Fichier de configuration de PXE

```
localboot pxelinux
default lanlinux
prompt 1
timeout 5
```

```
label lanlinux
kernel vmlinuz-2.6.18.2
append load initrd=tftpboot.img load_ramdisk=1
ramdisk_size=32768 ro root=/dev/ram0 irqpoll
```

Fichier `sysprep.inf`

```
[Unattended]
UnattendMode=FullUnattended
OemSkipEula=Yes
```

```
[GuiUnattended]
```

```
OemSkipWelcome = 1
OemSkipRegional=1
AdminPassword="xxxxxx"
TimeZone=23
```

```
[UserData]
FullName="Etudiants"
OrgName="UFR Math Info"
Computername=MACHINETP1
ProductID=XXXX-XXXX-XXXX-XXX...
```

```
[Networking]
InstallDefaultComponents=Yes
```

```
[Identification]
JoinWorkGroup=TPINFO
```

Composition de l'« `initrd` » :

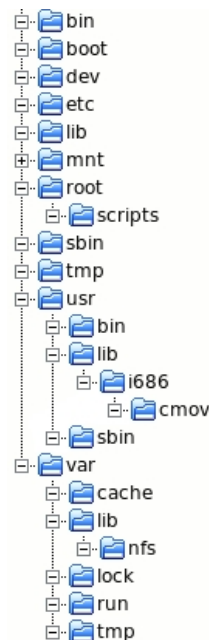


Figure 1 : Composition de l'initrd

Fichiers et exécutables composant l'« `initrd` » :

```
/sbin
/lib
/usr/bin : expr, awk, find, host, cut (indispensables)
/usr/sbin/
/usr/lib
(*)/etc : group, passwd,shadow,fstab,hostname,lilo.conf
(*)/etc/network : interfaces
(*)/root : .bashrc, .profile, bootscript.sh
(*)/etc/init.d/rc2.d/S99bootscript (lien symbolique
vers /root/bootscript.sh)
```


Les exécutable et bibliothèques des répertoires */bin*, */sbin*, */usr/bin*, */usr/sbin*, */lib* et */usr/lib* doivent être sélectionnés avec attention, car une copie brute de ces derniers pourrait augmenter inutilement la taille du fichier « *initrd* » (si ce dernier est trop imposant, le système refusera de le charger en mémoire). Nous conseillons vivement la création de cet « *initrd* » à partir d'une distribution la plus légère possible et dépourvue de tous les paquetages superflus.

(*) : ces fichiers doivent être modifiés pour refléter la configuration de la machine cible ; ce ne sont pas les fichiers présents sur la machine à partir de laquelle l'image « *initrd* » est générée.

Script de réinstallation

Remarque : ce script a été volontairement simplifié afin de le rendre plus lisible. En particulier, de nombreux tests ont été masqués (vérification de présence des fichiers, vérification du résultat des commandes, etc).

```
#!/bin/sh
# Déclaration de variables :
MNT=/mnt
BOOT=boot
WIN=image_XP.img.gz
LINUX=image_linux.img.gz
# Montage NFS : répertoire contenant les images et le MBR
mount -a

# Remplacement du MBR sur le disque cible (/dev/sda)
dd if=$MNT/$BOOT/mbr of=/dev/sda bs=512 count=1

# Relecture forcée du MBR du disque cible
sfdisk -R /dev/sda

# Création des systèmes de fichiers sur le disque cible
# Système VFAT sur la première partition
mkfs.vfat /dev/sda1

# Linux-Swap sur la seconde
mkswap /dev/sda2

# EXT3 sur la dernière
mkfs.ext3 /dev/sda3

# Restauration de l'image Windows
partimage -b -f 3 restore /dev/sda1 $MNT/$BOOT/$WIN

# Restauration de l'image Linux
```

```
partimage -b -f 3 restore /dev/sda3 $MNT/$BOOT/$LINUX
```

```
# Personnalisation des systèmes d'exploitation :
```

```
# Montage des deux partitions restaurées
```

```
mount /dev/sda1 $MNT/win
```

```
mount /dev/sda3 $MNT/linux
```

```
# Extraction de l'adresse MAC de la machine cible
```

```
MAC=`ifconfig | grep eth | grep HWaddr | awk -F " " \
{'print $5}`
```

```
# Recherche du nom de la machine dans "MAC_HOST"
```

```
HOST=`grep $MAC $MNT/$BOOT/sysprep/MAC_HOST \
\ awk -F " " {'print $2}`
```

```
# Linux : écriture de ce nom dans le fichier "hostname"
```

```
echo "HOSTNAME=${HOST}" \
```

```
> $MNT/linux/etc/conf.d/hostname
```

```
# Windows : suppression de l'archive sysprep résiduelle,
```

```
rm -f $MNT/win/*sysprep*.tgz
```

```
# Répertoire dans lequel placer notre fichier sysprep.inf
```

```
SYSPREP=`ls $MNT/win | grep prep | awk {'print $NF'}`
```

```
# Si aucun répertoire trouvé : création
```

```
if [ -z $SYSPREP ]
```

```
then
```

```
    SYSPREP="Sysprep"
```

```
    mkdir -p $MNT/win/$SYSPREP
```

```
fi
```

```
# Copie du fichier sysprep.inf sur la partition Windows
```

```
cp $MNT/$BOOT/sysprep/$HOST.inf \
```

```
$MNT/win/$SYSPREP/sysprep.inf
```

```
# Exécution de LILO pour que la machine restaurée démarre
```

```
# correctement
```

```
lilo -r $MNT/linux -C /etc/lilo.conf
```

```
# Redémarrage final
```

```
reboot
```